Tell Me Why It's Fake: Developing an Explainable User Interface for a Fake News Detection System

Erasmo Purificato[1,2,*,†], Saijal Shahania[1,3,†] and Ernesto William De Luca[1,2]

[1]Otto von Guericke University Magdeburg, Germany
[2]Leibniz Institute for Educational Media | Georg Eckert Institute, Brunswick, Germany
[3]German Centre for Higher Education Research and Science Studies, Hanover, Germany

Abstract

In this paper, we present the design and development of an *explainable user interface* for a *fake news detection system*. The problem of distinguishing real from fake articles gained a lot of popularity in the last few years, mainly due to the soaring diffusion of social networks and internet bots as means for propaganda and disinformation sharing. By leveraging various explainability methods, i.e. *feature importance*, *partial dependence plots* and *SHAP* values, we aim to show how the combination of different techniques embedded in an interactive user interface can lead to enhance trust in a detection system for a non-expert user, such as a fact-checker or a content manager. Through several examples, we describe all the explainability components along with the benefits and limitations they can provide to end users.

Keywords

Explainability, User Interface, Fake News Detection

1. Introduction

The concept of *fake news* has become extraordinarily familiar in the last few years, mainly due to the common idea of using social networks as an influential source to retrieve news. Ignoring the recent pandemic period, this expression reached the peak of its popularity during the historic 2016 US elections, where according to several studies (e.g. [1]), the elected president Donald Trump gained an advantage by disseminating unverified misinformation and propaganda through social media. As a consequence of that unusual scenario, the term was appointed as "word of the year" in 2017 by the American Dialect Society, which defines *fake news* as "*disinformation or falsehoods presented as real news or actual news that is claimed to be untrue*"[1].

Moreover, the advent and rapid diffusion of *internet bots* (simply known as *bots*) in social media [2] have added new perspectives to the already long-standing disinformation problem. Bots are nowadays massively present on the web and often difficult to recognise. In 2019, a

XAI.it 2022 - Italian Workshop on Explainable Artificial Intelligence
*Corresponding author.
†These authors contributed equally.
erasmo.purificato@ovgu.de (E. Purificato); saijal.shahania@ovgu.de (S. Shahania); ernesto.deluca@ovgu.de (E. W. De Luca)
https://erasmopurif.com/ (E. Purificato); https://ernestodeluca.eu/ (E. W. De Luca)
0000-0002-5506-3020 (E. Purificato); 0000-0003-3621-4118 (E. W. De Luca)

[1]https://www.americandialect.org/fake-news-is-2017-american-dialect-society-word-of-the-year.

study conducted by Liu [3] reported that ca. 17% of Twitter accounts are suspected to be bots, contributing to ca. 30% of the number of tweets posted. In addition to that, the use of bots can help spread fake news at a faster rate than any human user can. Considering that every single day ca. 500 million tweets are published[2], it is evident how reliable tools for detecting potential fake news are needed.

Currently, fact-checkers and content managers can make use of existing automated systems trained on a standard manually-tagged corpus, but this is not sufficient for several reasons. First, in order to keep up with continuously emerging fake news, a huge corpus of facts would be needed for cross-checking. Secondly, not all fake news is equal: some news entries are entirely fabricated, while others are built upon facts and injected with deceitful fake content or distorted and decontextualised opinions. Given this context, it is of paramount importance to design and develop automated systems able to detect the specific factors that distinguish the fake parts of an article from the rest. Since such factors have to be checked by content managers to be deemed false, the resulting decisions must be transparent and interpretable.

In such a scenario, the design and development of explainable user interfaces (UIs) play a fundamental role in providing the proper explanations to the end-users, even more than implementing the system itself in many cases [4]. Much research is being done in several fields, such as recommender systems [5] and responsible artificial intelligence [6], and also a new line of research is investigating the topic of adaptive and personalised explainable UIs [7].

In this paper, we present the development of an explainable UI for a fake news detection system, which can help users understand what parts of the analysed article are likely to be fake and for what reasons. Through the implementation of feature importance and post-hoc methods, we show how different explainability approaches can be combined and embedded in a UI to increase non-expert users' trust in such a system.

2. Related Work

With the described spread of fake news in our daily life due to the constant use of social media, a lot of research has been produced to find the specific characteristics of disinformation articles to apply strategies for detecting them and stopping their distribution. Initially, the proposed tools dealt with manual fact-checking methods, and one of the most popular is *Truth-o-Meter*[3]. Focusing especially on political news, it classifies news into six groups, considering different degrees of truth. Another famous resource in this direction is *Snopes*[4], which adds the analysis of images and videos to any trending news, not only politics.

Currently, more and more automatic detection systems are emerging, and Conroy et al. [8] presented an in-depth comparison between manual and automatic approaches. Several different types of techniques are exploited for building an automated fake news detection system. Shu et al. [9] show how to leverage different linguistic features extracted from an article's content, while other contributions make use of semantics and discourse features (e.g. [10, 11]). More recent works tried to analyse writing style aspects to unravel bias, such as [12], or to focus on

[2] https://www.internetlivestats.com/.
[3] https://www.politifact.com/truth-o-meter/.
[4] https://www.snopes.com/fact-check/.

knowledge-based features to enhance the credibility of the system, for example by utilising an external database for fact extraction and verification [13].

As in most research fields related to artificial intelligence (AI), the state of the art of fake detection is constituted by systems leveraging deep learning (DL) architectures. Zhang et al. [14] introduce the concept of *dual emotion features* in their neural network model to improve the performance by taking into account the relationship between the emotions depicted in the news and users' emotions expressed in the comments. A graph neural network-based approach to analyse the propagation behaviour of fake news has been proposed by Kipf and Welling [15].

Despite the proliferation of methodologies, in the illustrated works, the elements for transparent and explainable systems and related UIs are almost missing or at least latent. According to Ha et al. [16] and Zhou et al. [17], due to the coexistence of fake and real news, it is necessary to incorporate the vision of the audience, and this can be achieved via *explainability*. Shu et al. [18] propose *dEFEND*, a system that uses a deep hierarchical co-attention network to represent the textual features. The network is also used to discover relevant sentences in the articles and the subsequent users' comments responsible for detecting them as fake news. We use this sentence-level explainability feature in the presented work. Yang et al. [19] exploit attributes, semantics and linguistics for the proposed detection system, called *Xfake*. In this case, the explainability aspect derives from self-explainable features aided by visualisations, which also serve as the basis of our detector's explainability module.

3. Components of the Fake News Detection System

We illustrate the design of the fake news detection system, whose logic architecture is shown in Fig. 1, by briefly describing its components with a specific focus on the explainability component, which constitutes the basis for the core part of this paper, i.e. the explainable UI, presented in detail in the next section.

The system is designed with a cascading architecture composed of three main phases:

1. *Feature extraction*: low-level and high-level features are extracted from the fake news corpus and provided to the connected classifier. The datasets exploited as sources to extract the features are: the *ISOT Fake News Dataset*[5] provided by the University of Victoria, the *Fake News Dataset*[6] by Kaggle, the *Fake News Corpus*[7], the *Multi-Perspective Question Answering Dataset* (MPQA)[8] and the *Myers-Briggs Personality Type Dataset* (MBTI)[9].
2. *Classification*: leveraging the extracted features, a classifier is applied to the documents to produce the probability of how likely the analysed news is fake or real.
3. *Filtering*: based on the resulting probability and the related confidence level of the classifier that receives low-level descriptors in input, each news is filtered and marked as *fake*, *real* or *uncertain*. For the latter group, a deeper classification making use of high-level descriptors is applied.

[5]https://www.uvic.ca/ecs/ece/isot/datasets/fake-news/index.php.
[6]https://www.kaggle.com/datasets/jruvika/fake-news-detection.
[7]https://github.com/several27/FakeNewsCorpus.
[8]https://mpqa.cs.pitt.edu/corpora/mpqa_corpus/.
[9]https://www.kaggle.com/datasets/datasnaek/mbti-type.

The overall goal is to develop a system able to reach a high confidence score for each tested article. For this reason, as mentioned before, two different classifiers are used, each leveraging two different types of features:

- *Low-level descriptors*: basic linguistic features from the article texts and headlines, such as size, number of grammatical errors, parts of speech and term frequencies;
- *High-level descriptors*: more complex features detected from the news content with additional algorithms, like sentiment, entailment, attribution, syntactical structure, tones and latent topics.

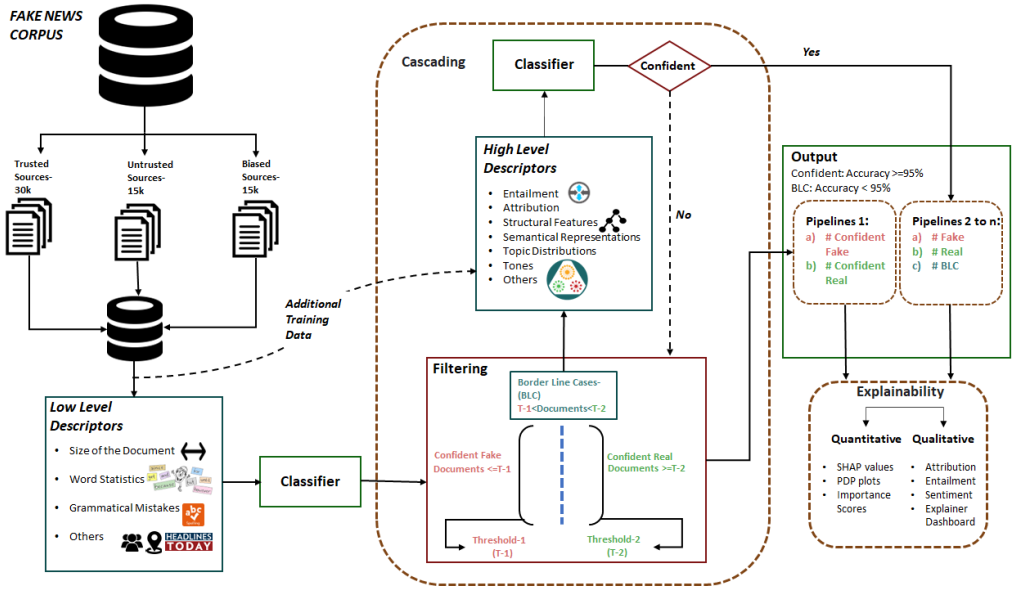


Figure 1: Logic architecture of the fake news detection system.

3.1. Explainability Component

The *explainability* component of the presented detection system incorporates both *intrinsic* methods, directly derived from the features extracted from the fake news datasets, and *extrinsic* methods provided through *quantitative* and *qualitative* approaches.

Intrinsic methods are related to the self-explainable low-level (and some of the high-level) linguistic features, which do not need any further processing procedure to be presented to the end users. For the extrinsic methods, we discuss the two approaches separately.

3.1.1. Quantitative explainability methods

The *quantitative* explainability methods included in our system are the following:

- *Feature importance*: it is a widely used method for finding the attributes that contribute the most towards the classifier's predictions. In particular, in our work, we adopted the

permutation-based feature importance approach proposed by Altmann et al. [20]. This method improves the interpretability of the results by enabling the users to understand what are the most influential features for a specific outcome. Furthermore, it can help build trust in non-experts by allowing them to validate the facts. For instance, if an article is predicted as fake because of its attribution to a fake source, one can cross-check this information by directly analysing the pointed source.

- *Partial Dependence Plots* (PDP): this method was proposed by Goldstein et al. [21] and aims to create a link between the target label (in our case, fake or real) and the attributes utilised by the classifiers (i.e. low-level and high-level descriptors). PDP is a model-agnostic and global method, meaning that it can handle any machine learning algorithm and all the derived observations are considered while presenting the final output. PDP plots display how a prediction changes by altering the value of a single feature. Since not all features affect the final outcome, this method is applied only to the descriptors with the maximum feature importance score.
- *SHapley Additive exPlanations* (SHAP): presented by Lundberg and Lee [22], this is one of the state-of-the-art techniques for explainability, and it is mainly used to figure out the effect of each attribute of a classifier's prediction. It is based on the concept of Shapley values [23], derived from games theory, and considers the reproduction of the prediction as a game and the input attributes as the players. We make use of the contribution plots obtained by the SHAP values, wherein we display the contribution of all the attributes towards its prediction for each observation.

3.1.2. Qualitative explainability method

The implemented *qualitative* explainability method involves merging the classification predictions with the extracted features and the results of the quantitative explainability techniques. To clearly illustrate this concept, we provide some examples. Fig. 2 show an excerpt from a news article by The Indian Express entitled "*Monkeypox virus could become entrenched as new STD in the US*"[10]. The figure demonstrates how *attribution* features can be used to aid explainability. A sentence is highlighted and coloured according to the predicted category. In this case, the red colour means that the sentence "Experts don't agree on the likely [...]" has been attributed to the reported sentence from a fake article published in a known disinformation source called *AmericaBlog.com*. Furthermore, the similarity score feature is exploited to weigh the sentence. The higher the similarity between the targeted sentence with the attributed source, the darker the highlighting colour.

In Fig. 3, we show the explanations made by the *entailment* and *sentiment* features with SHAP plots. Through the analysis of the sentence "*What a great movie!...if you have no taste.*", in the provided example, we can see which part of the sentence is responsible for the statement being contradictory (Fig. 3a) or having a positive sentiment (Fig. 3b). However, the models could not capture the sarcasm in the sentence, which would have made the sentence more likely to have negative emotions. This example shows how we can enable system transparency, giving the end user a chance to validate the predictions, and it can also be used by a developer.

[10] https://indianexpress.com/article/world/monkeypox-virus-could-become-entrenched-as-new-std-us-8046596/.

Although the sentence is labelled as having a positive emotion, it is clear to the reader that it is a sarcastic sentence that is not captured. This can point us toward what other techniques can be experimented with to overcome the limitations.

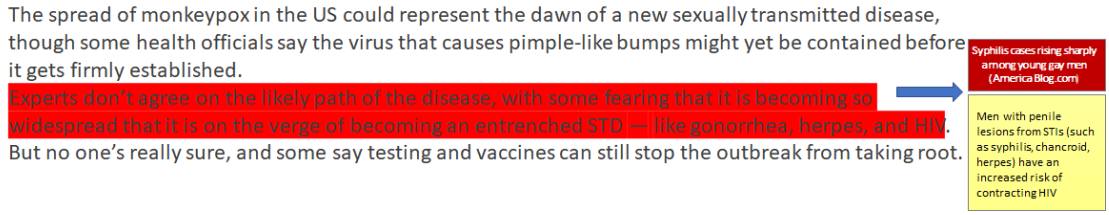
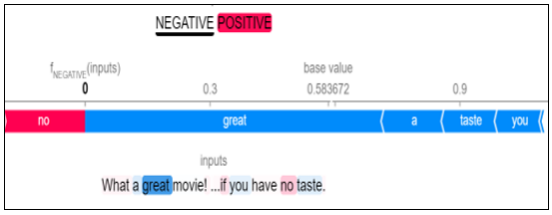


Figure 2: Example of the use of attribution features for the qualitative explainability method.

(a) Entailment score resulting as a contradiction.



(b) Sentiment score resulting as a positive emotion.

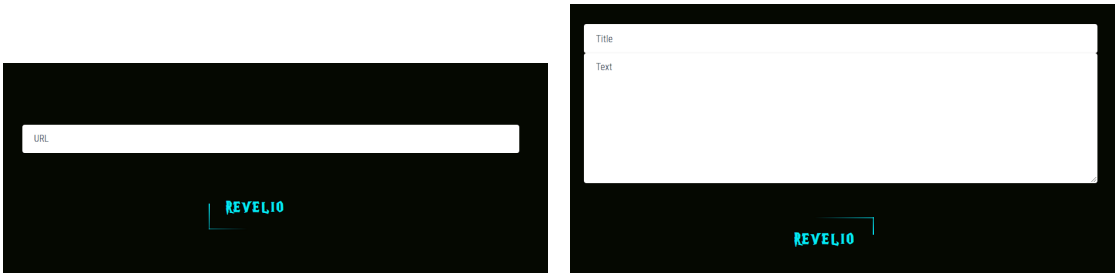
Figure 3: Example of the use of SHAP values for the qualitative explainability method.

4. Explainable User Interface

In this section, we describe the elements composing the *explainable user interface*, designed to provide end users with the possibility to upload a document and get all the information about the prediction of the fake news detection system, as well as the motivation behind it. As visible in every image displayed below, we developed the first version of our UI with a *Harry Potter* style, starting from the idea of a wizard revealing the truth or the lies in the analysed article.

When opening the application homepage, a user can choose between two input options for querying the detection system:

1. Insert the URL of a publicly available article (Fig. 4a);
2. Type manually the text to analyse (Fig. 4b). The second option can be useful to evaluate only a piece of news instead of the entire body, but we discourage this practice since the implemented fake news detection system could reveal important information about its truthfulness from the whole content.



(a) URL input (b) Custom text input

Figure 4: Input modes for the explainable user interface.

We illustrate the UI components by considering, as an example, an article from *bbc.com* about the recent Ukraine war[11]. **Note**: *Due to the delicate situation regarding the topic at hand, it is very important to specify that the intent of this paper is not to assess the correctness of the prediction of the fake news detection system (as it is still under development) but only to show how the explanations can be assembled effectively in a user interface.*

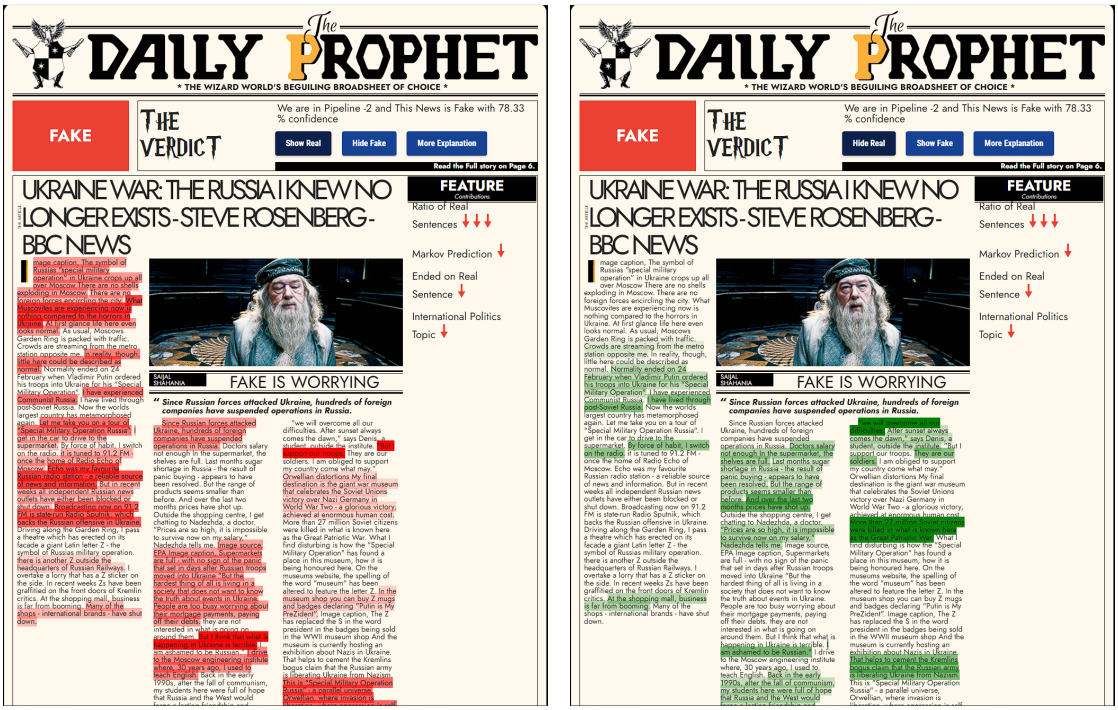
After clicking on the *Revelio* button, the pipelines described in Section 3 start. At the end of the computation, the user lands on the result page shown in Fig. 5.

Figure 5: Result page with the main information after the fake news detection system computation.

[11] https://www.bbc.com/news/world-europe-61188783.

The page presents at first the prediction of the detection system in the top left-hand corner, coloured according to the resulting label (in this case red, because the article has been predicted as fake). To the right of the result, we can read the confidence of the prediction for the system (i.e. 78.33%) and the landing pipeline for the classification. *Pipeline-2* means that the fake news detection system, after the filtering step described in Section 3, needed to exploit also high-level descriptors in order to reach a consistent confidence score.

By clicking on *Show Real* and/or *Show Fake*, the sentences contributing to the real and/or fake predictions are highlighted in the displayed text, as shown in Fig. 6.



(a) Contributions to *fake* prediction

(b) Contributions to *real* prediction

Figure 6: Result page with *fake* and *real* contributions highlighted.

As already seen in Fig. 2, the colour gradient relates to the similarity score between the specific sentence and the attributed source, which becomes visible by hovering over the text corresponding with a highlighted sentence. Examining the fake contributions in Fig. 6a, we can see in Fig. 7a that the statement "*But I think that what is happening in Ukraine is terrible*" is labelled as strongly fake because of the attribution to a very similar phrase (i.e. "*But what is happening in Ukraine is horrible*") from a known fake news source.

On the right-hand side of the result page, users can check the list of the most influential feature contributions towards the overall prediction. The arrows next to each feature name indicate how strong the contribution of that feature is through their number (from one to three), as well as their direction (i.e. green up arrows for real, red down arrow for fake). When hovering over a feature name, a popup appears showing the exact value and a short description of the selected feature, as shown in Fig. 7b.

(a) Attribution result (b) Feature description

Figure 7: Examples of explainable elements.

The last functionality proposed in the presented explainable UI is an in-depth analysis of a single instance. By clicking on the *More Explanation* button (close to the *Show/Hide Fake* one), we land on a new page showing the importance score, i.e. SHAP values, of all the features used by the classifier for that particular input. This visualisation element is developed by making use of the *ExplainerDashboard* tool[12], and it also contains two additional tabs. One is for the visualisation of the partial dependence plots, while the other, called "*What If...*", allows the user to play with the features and adjust their importance score to see if the prediction changes, in a counterfactual scenario. Since our aim is to address this tool to fact-checkers and content managers, through this functionality, they can even remove the features considered useless from the displayed list and check how the result and the related explanations change accordingly.

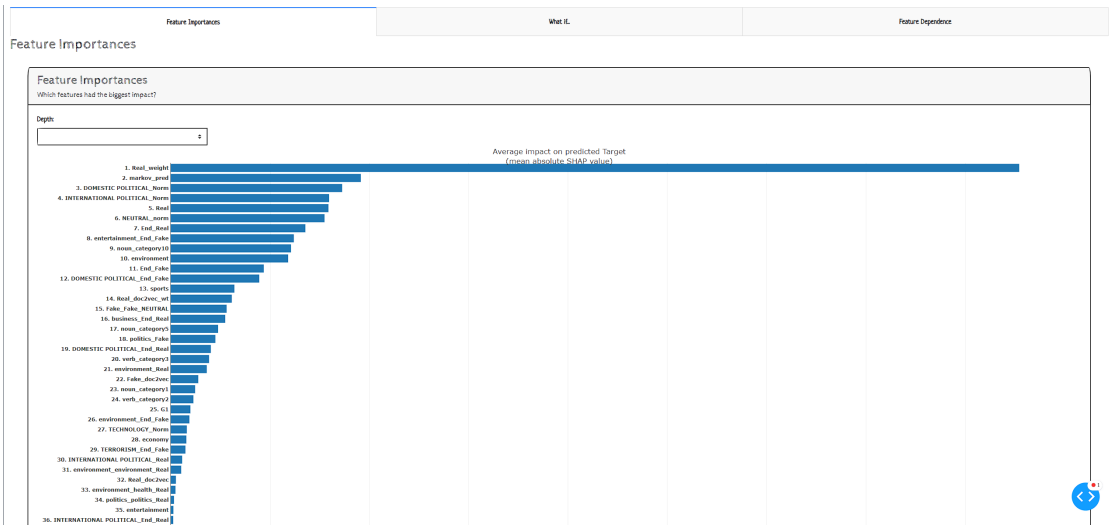


Figure 8: Visualisation of SHAP values through the *ExplainerDashboard* for a particular example.

[12]https://explainerdashboard.readthedocs.io/en/latest/.

5. Discussion

As presented in the previous section, we designed and implemented an *explainable UI* to provide an overview of how the developed *fake news detection system* behaves across the whole dataset as well as individual instances. In particular, we have given some explaining information, such as the sentences attributed to being fake or real, the four most important features used for classification, and a three-level scale of importance. With the latter, we mostly want to provide an end user, who could be a non-expert, the possibility to check the news for their reliability by giving them hints about parts of the news to look out for and which might be trustworthy.

One of the most interesting areas a fact-checker or a content manager can investigate in the dashboard is the overall model behaviour of the model in terms of the features used for classification. In Fig. 9, we restricted the figure to only displaying the ten most important features, which are calculated using the average of all SHAP values across the whole test set.

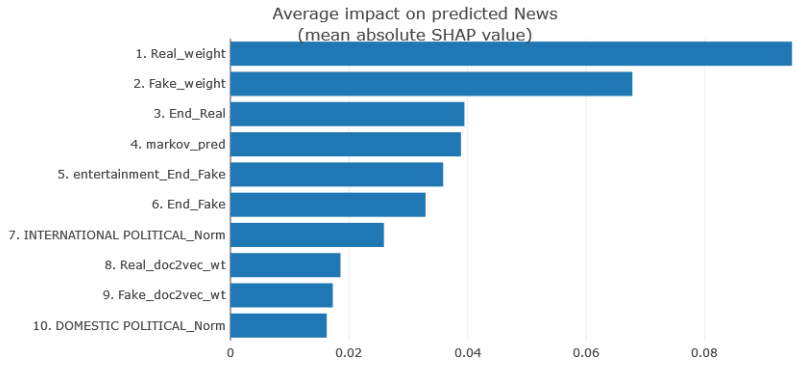


Figure 9: *ExplainerDashboard* extract showing the most influential features ranked by SHAP values.

The described visualisation element does not provide any insight into how exactly a document is classified. For that, we can make an individual feature importance analysis per article. We discuss below an example of a wrongly classified entry by analysing possible reasons to manually check the system result.

The article taken into account is entitled "*Israeli forces shoot dead Palestinian man in northern West Bank*"[13] and is an evident fake news document. However, our system detected it as real with a 78.3% confidence score. Fig. 10 shows that the most important characteristic is the text structure (denoted as *markov_pred*, i.e. whether it is more similar to fake or real articles) resembling a real article. Additionally, the article is detected as being about the environment and health, whereas the sentence preceding the health topic must be real. Of course, the article is neither about the environment nor about health. However, it is probably detected as the latter due to the text talking about critically injured, wounds, and so on. Very few sentences are only attributed to being fake, and most text is written neutrally.

[13] https://web.archive.org/web/20160829121057/http://www.presstv.com/Detail/2016/08/26/481771/Israeli-forces-shooting-Palestinian-man-death-Silwad-West-Bank.

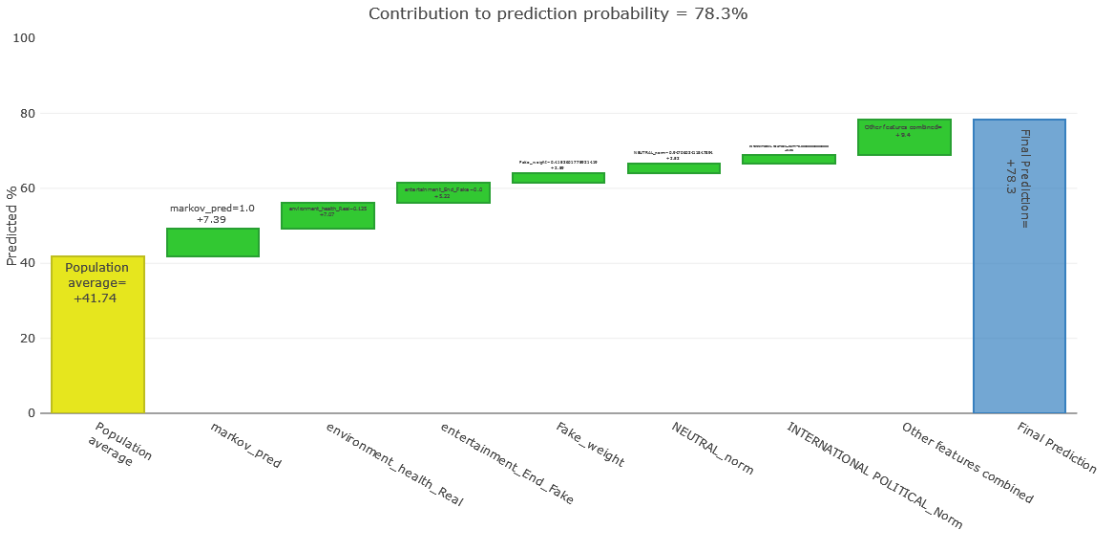


Figure 10: Feature importance for an individual example tested article.

6. Conclusion and Future Work

The presented paper describes the implementation of an explainable user interface for a fake news detection system designed with a cascading architecture and exploiting both low-level (basic linguistic features extracted from the content) and high-level (complex features computed by additional algorithms, like sentiment and attribution to external sources) descriptors. The UI is composed of several explainability elements related to feature importance, partial dependence plots and SHAP values. On the result page, a user can check the entire text with information related to the system prediction, the most influential features ranked by importance, as well as the sentences that have been attributed to either fake or real sources.

As the next steps, we aim to finalise the development of the detection system and execute a robust evaluation of the two classification pipelines, as well as the explainable user interface. We will also insert more explainability components in the UI, such as text plot for SHAP on entailment and sentiment features.

References

[1] H. Allcott, M. Gentzkow, Social media and fake news in the 2016 election, Journal of Economic Perspectives 31 (2017) 211–36.

[2] E. Ferrara, O. Varol, C. Davis, F. Menczer, A. Flammini, The Rise of Social Bots, Communications of the ACM 59 (2016) 96–104.

[3] X. Liu, A big data approach to examining social bots on Twitter, Journal of Services Marketing 33 (2019) 369–379.

[4] T. Miller, Explanation in artificial intelligence: Insights from the social sciences, Artificial intelligence 267 (2019) 1–38.
[5] E. Purificato, B. A. Manikandan, P. V. Karanam, M. V. Pattadkal, E. W. De Luca, Evaluating explainable interfaces for a knowledge graph-based recommender system, in: Proceedings of the 8th Joint Workshop on Interfaces and Human Decision Making for Recommender Systems co-located with 15th ACM Conference on Recommender Systems (RecSys 2021)„ CEUR, 2021, pp. 73–88.
[6] E. Purificato, F. Lorenzo, F. Fallucchi, E. W. De Luca, The use of responsible artificial intelligence techniques in the context of loan approval processes, International Journal of Human–Computer Interaction (2022) 1–20.
[7] E. Purificato, C. Musto, P. Lops, E. W. De Luca, First workshop on adaptive and personalized explainable user interfaces (apex-ui 2022), in: 27th International Conference on Intelligent User Interfaces, 2022, pp. 1–3.
[8] N. K. Conroy, V. L. Rubin, Y. Chen, Automatic deception detection: Methods for finding fake news, Proceedings of the association for information science and technology 52 (2015) 1–4.
[9] K. Shu, A. Sliva, S. Wang, J. Tang, H. Liu, Fake news detection on social media: A data mining perspective, ACM SIGKDD explorations newsletter 19 (2017) 22–36.
[10] J. Li, M. Ott, C. Cardie, E. Hovy, Towards a general rule for identifying deceptive opinion spam, in: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2014, pp. 1566–1576.
[11] V. Pérez-Rosas, R. Mihalcea, Cross-cultural deception detection, in: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), 2014, pp. 440–445.
[12] M. Potthast, J. Kiesel, K. Reinartz, J. Bevendorff, B. Stein, A stylometric inquiry into hyperpartisan and fake news, arXiv preprint arXiv:1702.05638 (2017).
[13] J. Thorne, A. Vlachos, C. Christodoulopoulos, A. Mittal, FEVER: a large-scale dataset for fact extraction and VERification, in: NAACL-HLT, 2018.
[14] X. Zhang, J. Cao, X. Li, Q. Sheng, L. Zhong, K. Shu, Mining dual emotion for fake news detection, in: Proceedings of the Web Conference 2021, 2021, pp. 3465–3476.
[15] T. N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks. 2017, ArXiv abs/1609.02907 (2017).
[16] L. Ha, L. Andreu Perez, R. Ray, Mapping recent development in scholarship on fake news and misinformation, 2008 to 2017: Disciplinary contribution, topics, and impact, American behavioral scientist 65 (2021) 290–315.
[17] X. Zhou, R. Zafarani, K. Shu, H. Liu, Fake news: Fundamental theories, detection strategies and challenges, in: Proceedings of the twelfth ACM international conference on web search and data mining, 2019, pp. 836–837.
[18] K. Shu, L. Cui, S. Wang, D. Lee, H. Liu, defend: Explainable fake news detection, in: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining, 2019, pp. 395–405.
[19] F. Yang, S. K. Pentyala, S. Mohseni, M. Du, H. Yuan, R. Linder, E. D. Ragan, S. Ji, X. Hu, Xfake: Explainable fake news detector with visualizations, in: The World Wide Web Conference, 2019, pp. 3600–3604.

[20] A. Altmann, L. Toloşi, O. Sander, T. Lengauer, Permutation importance: a corrected feature importance measure, Bioinformatics 26 (2010) 1340–1347.
[21] A. Goldstein, A. Kapelner, J. Bleich, E. Pitkin, Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation, Journal of Computational and Graphical Statistics 24 (2015) 44–65.
[22] S. M. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions, Advances in neural information processing systems 30 (2017).
[23] L. S. Shapley, A value for n-person games, Contributions to the Theory of Games 2 (1953) 307–317.