Machine Learning
https://doi.org/10.1007/s10994-024-06527-w

Exposing and explaining fake news on-the-fly

Francisco de Arriba-Pérez[1] · Silvia García-Méndez[1] · Fátima Leal[2] · Benedita Malheiro[3,4] · Juan Carlos Burguillo[1]

Received: 29 November 2022 / Revised: 7 July 2023 / Accepted: 14 February 2024

Abstract
Social media platforms enable the rapid dissemination and consumption of information. However, users instantly consume such content regardless of the reliability of the shared data. Consequently, the latter crowdsourcing model is exposed to manipulation. This work contributes with an explainable and online classification method to recognize fake news in real-time. The proposed method combines both unsupervised and supervised Machine Learning approaches with online created lexica. The profiling is built using creator-, content- and context-based features using Natural Language Processing techniques. The explainable classification mechanism displays in a dashboard the features selected for classification and the prediction confidence. The performance of the proposed solution has been validated with real data sets from Twitter and the results attain 80% accuracy and macro *F*-measure. This proposal is the first to jointly provide data stream processing, profiling, classification and explainability. Ultimately, the proposed early detection, isolation and explanation of fake news contribute to increase the quality and trustworthiness of social media contents.

Keywords Artificial intelligence · Data stream architecture · Machine learning · Natural language processing · Reliability and transparency · Social networking

1 Introduction

In social media, information is shared collaboratively through platforms like Facebook,[1] Twitter,[2] or Wikinews.[3] Such platforms enable the rapid dissemination of information regardless of its trustworthiness, leading to instant consumption of non-curated news. The negative consequence of this openness of social media platforms is the spread of false information disguised as truth, i.e., fake news. Fake news can be defined as deceptive posts

[1] Available at https://www.facebook.com, June 2023.
[2] Available at https://twitter.com, June 2023.
[3] Available at https://www.wikinews.org, June 2023.

Editor: Michelangelo Ceci, Joãso Gama, Jose Lozano, André de Carvalho, Paula Brito.

Extended author information available on the last page of the article

with an intention to mislead consumers in their purchase or approaching the context of misinformation and disinformation (Xiao et al., 2020). Specifically, while misinformation is an inadvertent action, disinformation is a deliberate creation/sharing of false information. The authenticity and intention can be distinguished as: (i) non-factual and mislead, i.e., deceptive news and disinformation; (ii) factual and mislead (cherry-picking); (iii) undefined and mislead (click-bait); and (iv) non-factual and undefined, i.e., misinformation.

Misinformation and fake news are characterized by their big volume, uncertainty, and short-lived nature. Furthermore, they disseminate faster and further on social media sites causing serious impact on politics and economics (Tandoc, 2019). Accordingly, the report on digital transformation of media and the rise of disinformation/fake news of the European Union (EU) (Martens et al., 2018) reinforces the need to strengthen trust in digital media.

This work contributes with a real-time explainable classification method to recognize fake news, promoting trust in digital media as suggested by the SocialTruth project.[4] In fact, the early discarding of fake news has a positive impact on both information quality and reliability. The proposed method employs stream processing, updating the profiling and classification models on each incoming event. The profiling is built using side-based (related to the creator user and propagation context) and content-based features (extracted from the news text through Natural Language Processing (NLP) techniques), together with unsupervised methods, to create clusters of representative features. The classification relies on stream Machine Learning (ML) algorithms to classify in real-time the nature of each cluster. Finally, the proposed method includes an explanation mechanism to detail why an event has been classified as fake or non-fake. The explanations are presented visually and in natural language on the user dashboard.

The rest of this paper is organized as follows. Section 2 overviews the relevant work on fake news concerning the profiling, classification and detection tasks. Section 3 introduces the proposed method, detailing the data processing and stream-based classification procedures along with the online explainability. Section 4 describes the experimental set-up and the empirical evaluation results considering the online classification and explanation. Finally, Sect. 5 concludes and highlights the achievements and future work.

2 Related work

Social media plays a crucial role in news consumption due to its low cost, easy access, variety, and rapid dissemination (Hu et al., 2014). Indeed, social media is becoming an increasing source of breaking news. However, the fake news problem indicates that social platforms suffer from lack of transparency, reliability, and real-time modeling. In this context, fake news (misinformation/disinformation, such as rumor, deception, hoaxes, spam opinion, click-bait and cherry-picking) are false information created with the dishonest intention to mislead consumers (Choraś et al., 2021; Xiao et al., 2020). To characterize the nature of fake news and understand whether they result from inadvertent or deliberate action, it is necessary to establish their authenticity and the intention of the creator (Shu et al., 2017). In addition, social media streams are subject to feature variation over time (Bondielli and Marcelloni, 2019; Choraś et al., 2021). Thus, the accurate detection of fake

[4] Available at http://www.socialtruth.eu/index.php/documentation, June 2023.

news in real time requires proper profiling and classification techniques. However, according to Shu (2022), the current detection techniques are based on opaque models, leaving users clueless about classification outcomes. Consequently, the current work addresses transparency through explanations, reliability through fake news detection, and real-time modeling through incremental content profiling.

The following discussion compares existing works in terms of: (i) stream-based profile modeling for fake detection; (ii) stream-based classification mechanisms; and (iii) transparency and credibility in detection tasks.

2.1 Profiling

Profiling methods model the stakeholders according to their contributions and interactions. Due to information sparsity, it is frequent to represent profiles using side and content information. In addition, in stream-based modeling, profiles are continuously updated and refined. To model fake news stakeholders, the literature contemplates multiple types of profiling methods: (i) creator-based; (ii) content-based; and (iii) context-based.

Creator-based profiling focuses on both demographic and behavioral characteristics of the creator. Specifically, the literature contemplates account name, anomaly score,[5] credibility score, geolocation information, ratio between friends and followers, total number of tweets/posts, etc. (Castillo et al., 2011; Goindani and Neville, 2019; Jang et al., 2021; Jain et al., 2022; Li et al., 2021; Liu and Wu, 2020; Mosallanezhad et al., 2022; Silva et al., 2021a; Vicario et al., 2019; Zubiaga et al., 2017).

Content-based profiling explores textual features extracted from the post aiming to identify the meaning of the content. It can be obtained using linguistic and semantic knowledge, or style analysis via NLP approaches together with fact-checking resources,[6]. Most of the revised works exploit this type of features. Therefore, content-based profiling encompasses:

- *Lexical and syntactical features* are properties related to the syntax, e.g., sentence-level features, such as bag-of-words approaches, *n*-grams, and part-of-speech. These features are exploited by Dong et al. (2020); Zhou et al. (2020). In addition, Vicario et al. (2019); Jang et al. (2021) compute the overall sentiment score of sentences.
- *Stylistic features* provide emphasis and clarity to the text. Tweet-writing styles can be determined through: (i) physical style analysis (e.g., number of adjectives, nouns, hashtags and mentions as well as emotion words and casual words); and (ii) non-physical style analysis (e.g., complexity and readability of the text). The work by Jang et al. (2021) is a representative example of the physical style analysis.
- *Visual features* describe the properties of images or videos used to ascertain the credibility of multimedia content. Visual features can: (i) be purely statistic (e.g., number of images/videos); (ii) represent distribution patterns; or (iii) describe user accounts (e.g.,

[5] It is computed by the number of the user's interaction in a time window divided by the user's monthly average.

[6] E.g. Classify.news, FackCheck.org, Factmata.com, Hoaxy.iuni.iu.edu, Hoax-Slayer.com, PolitiFact.com, Snopes.com, TruthOrFiction.com.

background images). While Jang et al. (2021) compute statistic visual features, Liu and Wu (2020) consider information from the user account. Li et al. (2021) verify if the image has been tampered, integrating this information as visual content, and Ying et al. (2021) combine textual with visual content to generate multi-level semantic features.

Context-based profiling analyses both the surrounding environment and the creator engagements around the piece of information posted (Castillo et al., 2011; Goindani and Neville, 2019; Jain et al., 2022; Jang et al., 2021; Li et al., 2021; Liu and Wu, 2020; Puraivan et al., 2021; Shu et al., 2019b; Silva et al., 2021a; Song et al., 2021; Zhao et al., 2020). Specifically, it applies user-network analysis and distribution pattern analysis to obtain:

- *Network-based features* which aggregate similar online users in terms of location, education background, and habits (Liu and Wu, 2020; Shu et al., 2019b; Silva et al., 2021a).
- *Propagation-based features* that describe the dissemination of fake news based on the propagation graph as in the work by Mosallanezhad et al. (2022). These may include, for an online account, the root degree, sub-trees number, the maximum/average degree and depth tree depth (Castillo et al., 2011; Jang et al., 2021) or the number of retweets/re-posts for the original tweet/post, the fraction of tweets/posts retweeted (Li et al., 2021; Zhao et al., 2020).
- *Temporal-based features* which detail how two posts/tweets relate in time. They may comprise the posting frequency, the day of the week of the post (Jang et al., 2021; Silva et al., 2021a), the interval between two posts or even a complete temporal graph (Song et al., 2021).

2.2 Classification

Fake news detection is a classification task. The main news classification techniques in the literature encompass supervised, semi-supervised, unsupervised, deep learning, and reinforcement learning approaches. Deep learning, depending on the problem, can fall into the supervised or unsupervised classification scope (Mathew et al., 2021). Moreover, its high computational cost requires more computational resources than the corresponding traditional approaches, motivating a separate discussion.

Supervised classification is a widely used technique to map objects to classes based on numeric features or inputs (see Table 1). The most frequently used supervised fake news detectors are Bayes, Probabilistic, Neighbor-based, Decision Trees, and Ensemble classifiers.

Semi-supervised classification algorithms learn from both labeled and unlabeled samples. They are employed when it is difficult to annotate manually or automatically the samples. The works by Dong et al. (2020) and Shu et al. (2019b) use supervised learning for fake news detection.

Unsupervised classification techniques group statistically similar unlabeled data based on underlying hidden features, using clustering algorithms or neural network approaches. The most commonly used cluster algorithms include k-means, Iterative

Self-Organizing Data Analysis Technique, and Agglomerative Hierarchical. Li et al. (2021) and Puraivan et al. (2021) are representative examples of this approach.

Deep Learning classification relies essentially on neural networks with three or more layers. In terms of fake news, deep learning has been employed mainly for text classification using Convolutional Neural Networks (CNN), Long Short Term Memory (LSTM), and Recurrent Neural Networks (RNN) as in the works by Akinyemi et al. (2020) and Nasir et al. (2021).

Reinforcement Learning classification works with unlabeled data (Sutton and Barto, 2018), but tends to be slow when applied to real-world classification problems (Dulac-Arnold et al., 2021). While Goindani and Neville (2019), Mosallanezhad et al. (2022), and Wang et al. (2020) perform fake news detection through reinforcement learning, the most used technique is the Multivariate Hawkes Process (MHP) by Goindani and Neville (2019).

Classification can be performed offline or online. Offline or batch processes build static models from pre-existing data sets, whereas online or stream-based processes compute incremental models from live data streams in real-time.

Offline classification divides the data set into training—used to create the model—and testing—to assess the quality of the model—partitions. The model remains static throughout the testing stage. This is the most popular fake news detection approach found in the literature.

Online classification mines data streams in real-time. Fake news, being dynamic sequences of data originated from multiple sources, i.e., the crowd, demand real-time processing. Typically, whenever new data arrive, the models are incrementally updated, enabling the generation of up-to-date classifications. To the best of the authors' knowledge, only Ksieniewicz et al. (2020) perform online fake news detection, processing samples as a data stream and considering concept drifts, i.e., that sample classification may naturally change over time.

Classification models can be interpretable and opaque. While opaque models behave as black boxes (e.g., standalone deep neural networks), interpretable models are self-explainable (e.g., trees- or neighbor-based algorithms). Interpretable classifiers explain classification outcomes (Škrlj et al., 2021), clarifying why a given content is false or misleading. More in detail, the explainable fake news detection framework by Shu et al. (2019a) integrates a news content encoder, a user comment encoder, and a sentence-comment co-attention network. The latter captures the correlation between news contents and comments and chooses the top-k sentences and comments to explain the classification outcome. Zhou et al. (2020) explore lexicon-, syntax-, semantic-, and discourse-level features to enhance the interpretablity of the models. Mahajan et al. (2021) and Kozik et al. (2022) adopt model agnostic interpretability techniques, such as Local Interpretable Model-agnostic Explanations (LIME) (Ribeiro et al., 2016) and the Shapley Additive Explanations (SHAP) (Lundberg and Lee, 2017), respectively. Finally, Silva et al. (2021a) provide explanations based on feature weights assigned to tweet/retweet nodes in the propagation patterns.

Table 1 provides an overview of the above works considering profiling (creator-, content-, and context-based), classification (supervised, semi-supervised, unsupervised, and reinforcement learning), processing (offline and online) and explainability. Summing up, this literature review shows that existing explainable fake news detectors explore creator-,

Table 1 Comparison of fake news detection approaches considering: (i) profiling (creator, content, context), (ii) classification (supervised, semi-supervised, unsupervised, reinforcement learning), (iii) execution (offline, online), and (iv) explainability (Ex.)

Proposal	Profiling	Classification	Execution	Ex.
Castillo et al. (2011)	Creator Content Context	Supervised	Offline	No
Liu and Wu (2020)				
Jang et al. (2021)				
Jain et al. (2022)				
Zubiaga et al. (2017)	Creator	Supervised	Offline	No
Vicario et al. (2019)	Content			
Song et al. (2021)	Content Context	Supervised	Offline	No
Akinyemi et al. (2020)	Content	Supervised	Offline	No
Silva et al. (2020)				
Nasir et al. (2021)				
Ying et al. (2021)				
Galli et al. (2022)				
Zhao et al. (2020)	Context	Supervised	Offline	No
Dong et al. (2020)	Content	Semi-supervised	Offline	No
Shu et al. (2019b)	Context	Semi-supervised	Offline	No
Puraivan et al. (2021)	Content Context	Unsupervised and supervised	Offline	No
Li et al. (2021)	Creator Content Context	Unsupervised	Offline	No
Mosallanezhad et al. (2022)	Creator Content	Reinforcement Learning	Offline	No
Goindani and Neville (2019)	Creator Context	Reinforcement Learning	Offline	No
Wang et al. (2020)	Content	Reinforcement Learning	Offline	No
Silva et al. (2021a)	Creator Content Context	Supervised	Offline	Yes
Shu et al. (2019a)	Content	Supervised	Offline	Yes
Zhou et al. (2020)				
Mahajan et al. (2021)				
Kozik et al. (2022)				
Ksieniewicz et al. (2020)	Content	Supervised	Online	No
Current	Creator Content Context	Unsupervised and Supervised	Online	Yes

content-, and context-based profiles, essentially adopt supervised classification and mostly implement offline processing.

The most closely related works from the literature, considering the PHEME experimental data used for design and evaluation, are the fake news classification solutions proposed by Akinyemi et al. (2020), Jain et al. (2022), Ying et al. (2021), and Zubiaga et al. (2017). Firstly, Zubiaga et al. (2017) experimented with sequential (Conditional Random Fields, Maximum Entropy and Enquiry-based) and non-sequential (Naive Bayes, Support Vector Machines (SVM) and Random Forests (RF)) classifiers. Secondly, Akinyemi et al. (2020) applied a RF model as the meta classifier trained with a stack-ensemble of SVM, RF, and RNN models as base learners. Thirdly, Ying et al. (2021) presented a Multi-level Multi-modal Cross-attention Network for batch fake detection. Furthermore, Jain et al. (2022) employed a Hierarchical Attention Network (HAN) and a Multi-Layer Perceptron (MLP) trained with creator-, content-, and context-based features. The final prediction (fake or non-fake) combines both classifier outputs through a logical OR. Nonetheless, all these solutions work offline without explaining the outcomes. In contrast, our work exploits a wide variety of profiling features (creator, content, and context), operates online and is able to explain the classification outcomes.

Similarly to our research, Puraivan et al. (2021) combineed both unsupervised and supervised techniques, for feature extraction (Principal Component Analysis and t-Distributed Stochastic Neighbor Embedding) and classification (optimized distributed gradient boosting), respectively. However, this offline work disregards the textual content of the news and lacks transparency.

Finally, the sole online system found explores fake news detection with Gaussian Naive Bayes, MLP, and Hoeffding Tree base learners independently and in ensembles (Ksieniewicz et al., 2020). Unfortunately, this work uses another data set collected by the authors and automatically labeled by BS Detector Chrome Extension. Profiles are exclusively based on content features and the outcomes are not explained.

2.3 Research contribution

As previously stated, this work contributes with an explainable classification method to recognize in real-time fake news and, thus, promote trust in digital media. Particularly, the method implements online processing, updating profiles and classification models on each incoming event. First, user profiles are built using creator-, content- and context-based features engineered through NLP. Then, unsupervised methods are exploited to create clusters of representative features. Finally, interpretable stream-based ML classifiers establish the trustworthiness of tweets in real-time. As a result, the proposed method provides the user with a dashboard, combining visual data and natural language knowledge, to make tweet classification transparent.

3 Proposed method

The proposed online and explainable fake news detection system is described in Fig. 1. It is composed of three main modules: (i) the stream-based data processing module (Sect. 3.1) which comprises feature engineering (Sect. 3.1.1), and analysis and selection tasks (Sect. 3.1.2); (ii) the stream-based classification module (Sect. 3.2) composed of lexicon-based (Sect. 3.2.1), unsupervised and supervised (Sect. 3.2.2) classifiers; and (iii) the

stream-based explainability module (Sect. 3.3). The explored data comprises two collections of tweets related to breaking news events released in 2016 (PHEME) and augmented in 2018 (PHEME-R).

3.1 Stream-based data processing

This module exploits NLP techniques to take full advantage of the ML models. Firstly, the feature engineering process generates new knowledge from the experimental data. Then, it analyses the resulting feature set to finally select the most relevant features for the classification.

3.1.1 Feature engineering

The proposed system computes features from a wide spectrum: (i) creator-, (ii) content- (lexical and syntactical features, stylistic features, and visual features), and (iii) context-based (network, distribution and temporal) features.

The creator-based features specify whether the user has an account description, a profile image and if the account has been protected and/or verified, the timezone, the number of followers and friends, the ratio between friends and followers, as well as the number of favourite tags received by the user. In the end, the time span in days between user registration and tweet post is calculated along with the weekly post frequency of the user.[7]

The linguistic and syntactic content-based features include the word *n*-grams from the processed tweet and whether the content is duplicated in the experimental data set. The physical style features comprise the adjective, auxiliary, bad word, determiner, difficult word, hashtag, link (also repeated), noun, pronoun, punctuation, uppercase word and word counters. The sentiment-related features comprise emotion (anger, fear, happiness, sadness and surprise) and polarity (negative, neutral and positive). The non-physical style-based features are based on the Flesch reading ease metric (see Table 2), the McAlpine EFLAW readability score for English foreign speakers[8] and the reading time in seconds. Concerning visual-based features, the system verifies if the tweet contains links to images and videos.

The generated context-based features consider whether the tweet has been retweeted and/or favourited, the depth of the retweet distribution network and the number of first-level retweets. Finally, the distribution pattern is analysed through the retweet and favourite counters.

The specific techniques applied to compute the aforementioned features will be described in Sect. 4.2.1 along with the data processing details.

3.1.2 Feature analysis and selection

Prior to feature selection, the system computes the variance of the features to establish their relative importance and, finally, discard those with low variance. Thus, the feature space dimension is reduced to minimize the computational load and time needed by ML models to classify tweets.

[7] These last two features may be considered as context-based temporal.

[8] It is recommended to be equal or lower than 25 points.

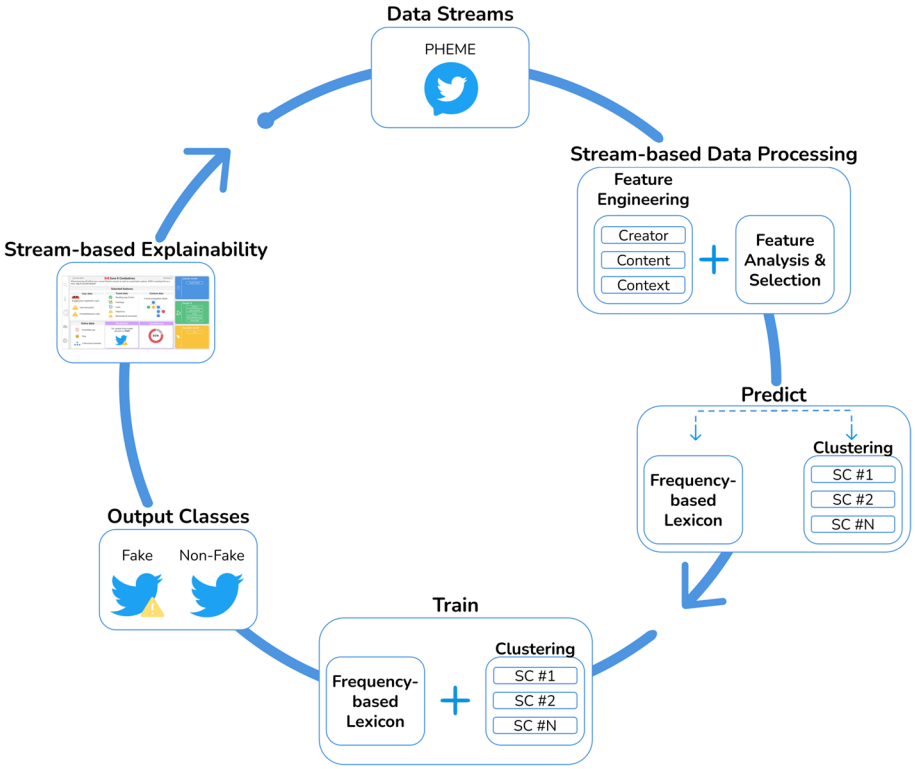

Fig. 1 System diagram composed of: (i) stream-based data processing, (ii) online classification, and (iii) stream-based explainability

3.2 Stream-based classification

The proposed method involves lexicon-based (Sect. 3.2.1), unsupervised and supervised classification 3.2.2 in both the predict and train steps of each incoming tweet.

3.2.1 Frequency-based lexicon

The adopted frequency-based lexicon is applied to the content of each incoming tweet. Algorithm 1 provides the corresponding pseudo-code. The lexica allow swift prediction followed by updating (training) based on the tweet content. The training stage considers the target class the n-grams represent and their frequency. More in detail, it defines three thresholds: (i) the n-gram range to extract the words; (ii) the number of elements to be included in the resulting lexica; and (iii) the frequency used as insert condition.

Table 2 Flesch reading ease score and difficulty

Score	Difficulty
90–100	Very easy
80–89	Easy
70–79	Fairly easy
60–69	Standard
50–59	Fairly difficult
30–49	Difficult
0–29	Very confusing

Algorithm 1 Frequency-based lexicon generation

Require: Content-related data of the incoming tweet: `tweet_processed, ngram_range, num_elements, threshold, frequency_lexicon.`
Ensure: The algorithm returns the `fake_lexicon` and `non-fake_lexicon` disjoint sets.
1: tweet_ngrams=$generate\_ngrams$(tweet_processed, ngram_range); // `Holds the` n-`gram representation (within the given` n-`gram_range of the tweet content) and frequency.`
2: frequency_lexicon=$update\_frequency\_lexicon$(frequency_lexicon, tweet_ngrams);
3: fake_lexicon=$frequency\_lexicon[class$ = fake, $frequency >$ threshold$][0{:}num_elements]$;
4: non-fake_lexicon=$frequency\_lexicon[class$ = non-fake, $frequency >$ threshold$][0{:}num_elements]$;
return $frequency\_lexicon$, $fake\_lexicon$, $non\text{-}fake\_lexicon$. // `Returns the updated frequency_lexicon plus the fake and non-fake lexica.`

3.2.2 Unsupervised and supervised classification

First, the unsupervised classification creates clusters of comparable spatial extent, by splitting the input data based on their proximity. It applies *k*-means clustering (Sinaga and Yang, 2020; Vouros et al., 2021) to minimise within-cluster variances, also known as squared Euclidean distances. Then, for each discovered cluster, one supervised classifier is trained.

The method involves several well-known stream-based ML models, selected according to their good performance in similar classification problems (Aphiwongsophon and Chongstitvatana, 2018; Silva et al., 2021b; Xiao et al., 2020).

- *Adaptive Random Forest Classifier* (ARFC) (Gomes et al., 2017). It induces diversity using re-sampling, random feature subsets for node splits and drift detectors per base tree.
- *Hoeffding Adaptive Tree Classifier* (HATC) (Bifet and Gavaldà, 2009). It uses a drift detector to monitor branch performance. Moreover, it presents a more efficient and effective bootstrap sampling strategy compared to the original Hoeffding Tree classifier.
- *Hoeffding Tree Classifier* (HTC) (Pham et al., 2017). It is an incremental decision tree algorithm which quantifies the number of samples needed to estimate the statistics while guarantying the prescribed performance.

- *Gaussian Naive Bayes* (GNB) (Xue et al., 2021). It enhances the original Naive Bayes algorithm by exploiting a Gaussian distribution per feature and class.

Algorithmic performance is determined with the help of classification accuracy, *F*-measure (macro and micro-averaging) and run-time metrics, following the prequential evaluation protocol (Gama et al., 2013).

3.3 Stream-based explainability module

Transparency is essential to make results both understandable and trustworthy for the end users. This means that outcomes need to be accompanied by explanatory descriptions. The designed fake news classification solution relies on interpretable models to obtain and present the relevant data in an explainability dashboard. The explanation of each prediction includes:

- Relevant user, content and context features selected by the supervised ML models.
- Predicted class (fake and non-fake) together with confidence.
- *K* disjoint elements ordered by their appearance frequency extracted from the fake and non-fake lexica.
- *K* features that surround the centroid of the cluster to which the entry belongs.

The latter is completed with natural language descriptions of the corresponding tree decision path.

4 Experimental results

All experiments were performed using a server with the following hardware specifications:

- Operating System: Ubuntu 18.04.2 LTS 64 bits
- Processor: Intel@Core i9-10900K 2.80 GHz
- RAM: 96 GB DDR4
- Disk: 480 GB NVME + 500 GB SSD

4.1 Experimental data sets

The experiments were performed with temporally ordered data streams created from the PHEME and PHEME-R data sets[9] and, for additional testing, from the Nikiforos et al. (2020) data set.[10] The PHEME collections comprise 6424 tweets created by 2893 users between August 2014 and March 2015. All tweets were manually labeled as fake and non-fake. The data set from Nikiforos et al. (2020) contains 2366 tweets posted by 51 users between April

[9] Available at https://figshare.com/articles/dataset/PHEME_dataset_for_Rumour_Detection_and_Veracity_Classification/6392078 and https://figshare.com/articles/dataset/PHEME_dataset_of_rumours_and_non-rumours/4010619, June 2023.

[10] Available at https://hilab.di.ionio.gr/wp-content/uploads/2020/02/HILab-Fake_News_Detection_For_Hong_Kong_Tweets.xlsx, June 2023.

2013 and December 2019. This data set was exclusively used to confirm the performance of the proposed method (see Sect. 4.3.3). Table 3 details the number of users and tweets per class in each experimental data set.

4.2 Stream-based data processing

As previously mentioned, data processing applies NLP techniques to ensure the competing performance of the ML models. The procedures used for online feature engineering, analysis and selection are presented below.

4.2.1 Feature engineering

Firstly, tweet content is purged from URL, redundant blank spaces, special characters (non-alphanumerical items, like accents and punctuation marks) and stop-words from the list provided by the Natural Language Toolkit (NLTK).[11] The remaining content is lemmatised with the English `en_core_web_md` model[12] of the spaCy library[13] and content polarity is established with `TextBlob`,[14] a sentiment analysis component for spaCy. The tweet emotion is calculated using `Text2emotion` Python library.[15]

The creation of non-physical style features relies on the `TextDescriptives`[16] spaCy module (features 13, 14, 17, 26, 28 and 29 in Table 4) and on the `Textstat`[17] Python library (features 18, 20, 25 and 30 in Table 4). The bad word count (feature 15 in Table 4) depends on the list provided by Wikimedia Meta-wiki.[18]

Given the importance of hashtags within tweets, hashtags are decomposed into their elementary constituents, i.e., words. This is applied to the cases where the hashtag is not represented in title format.[19] This splitter uses a freely available English corpus, the `Alpha` lexicon,[20] along with the English corpus by García-Méndez et al. (2019). It employs a recursive and reentrant algorithm to minimise the number of splits needed to decompose the hashtag into correct English words. As an example, the proposed text decomposition solution splits *hatecannotdriveouthate* as *hate cannot drive out hate*.

The word *n*-grams are extracted from the accumulated tweet textual data using `CountVectorizer`[21] Python library. Listing 1 shows the ranges and best values for the `CountVectorizer` configuration parameters based on iterative experimental tests with `GridSearch`[22] meta transformer wrapper for the HATC classifier.

[11] Available at https://gist.github.com/sebleier/554280, June 2023.
[12] Available at https://spacy.io/models/en, June 2023.
[13] Available at https://spacy.io, June 2023.
[14] Available at https://pypi.org/project/spacytextblob, June 2023.
[15] Available at https://pypi.org/project/text2emotion, June 2023.
[16] Available at https://spacy.io/universe/project/textdescriptives, June 2023.
[17] Available at https://pypi.org/project/textstat, June 2023.
[18] Available at https://meta.wikimedia.org/wiki/Research:Revision_scoring_as_a_service/Word_lists/en, June 2023.
[19] The first letter of each of the words which compose the hashtag capitalised.
[20] Available at https://github.com/dwyl/english-words, June 2023.
[21] Available at https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html, June 2023.
[22] Available at https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html, June 2023.

Table 3 Classes, number of users and tweets of the experimental data sets

Data set	Class	Users	Tweets
PHEME	Fake	1023	2402
	Non-fake	2204	4022
	Total	2893	6424
Nikiforos et al. (2020)	Fake	42	272
	Non-fake	9	2094
	Total	51	2366

```
maxdf = [0.7, 0.5, 0.3]
mindf = [0.1, 0.01, 0.001]
ngramrange = [(1, 2), (1, 3), (1,4)]
```

Listing 1 Parameter ranges for the generation of n-grams (best values in bold)

Table 4 shows the creator-, content- and context-based features selected for the detection of fake news. An additional pair of features is created for each user and numerical feature in Table 4 (features 6–9, 13–18, 21–24, 26, 28, 29, 31–33, 39 and 40): the user incremental feature average and latest feature trend, a Boolean feature that compares the last user feature value with the current user feature average.[23]

4.2.2 Feature analysis and selection

The method analyses the variance of features in Table 4 to compute their relative importance. Those features with low variance are discarded. Particularly, feature selection is performed at each incoming event using the `VarianceThreshold`[24] algorithm from River[25] library to improve the fake class recall metric.

4.3 Stream-based classification

Online classification involves prediction and training for each incoming sample. This section presents the results obtained by the lexicon-based, unsupervised and supervised classification procedures.

4.3.1 Frequency-based lexicon

The building of dynamic frequency-based lexicon starts after accumulating 5% of the samples. More in detail, the system extracts 700 from 2- to 4-word-length unique elements for each target class (fake and non-fake). Listing 2 provides the configuration parameter ranges. Best values were obtained once again from iterative experimental tests and using the HATC classifier.

[23] True if the feature value is equal or higher than the user feature average; otherwise is false.
[24] Available at https://riverml.xyz/0.11.1/api/feature-selection/VarianceThreshold, June 2023.
[25] Available at https://riverml.xyz/0.11.1, June 2023.

Table 4 Features considered for the classification by profile (creator, content, context) and data type (Boolean, categorical, numerical, textual)

Profiling	Data type	Number	Name
Creator-based	Boolean	1	Has profile description
		2	Has profile image
		3	Protected
		4	Verified
	Categorical	5	Timezone
	Numerical	6	Follower count
		7	Friend count
		8	Friends-followers ratio
		9	User favourite count
		10	Tweet-registration time spam (in days)
		11	Weekly tweet frequency
	Boolean	12	Text duplicated
Content-based	Numerical	13	Adjective count
		14	Auxiliary count
		15	Bad word count
		16	Char count
		17	Determiner count
		18	Difficult word count
		19	Emotion (anger, fear, happiness, sadness, surprise)
		20	Flesch reading ease
		21	Hashtag count
		22	Image count
		23	Link count
		24	Link repeated count
		25	McAlpine EFLAW readability
		26	Noun count
		27	Polarity
		28	Pronoun count
		29	Punctuation count
		30	Reading time
		31	Uppercase word count
		32	Video count
		33	Word count
	Textual	34	Word *n*-grams
Context-based	Boolean	35	Retweeted
		36	Tweet favourited
	Numerical	37	Distribution depth
		38	First level retweet
		39	Retweet count
		40	Tweet favourite count

```
ngrams = [(1,4),(2,4),(3,4)]
numberwords = [800,700,600,500, 400, 300, 200]
minfreqvalue = [1, 2, 5, 8, 10, 15, 20, 30]
```

Listing 2 Parameter ranges for the generation of the frequency-based lexicon(best values in bold)

4.3.2 Unsupervised and supervised classification results

As described in Sect. 3.2, the first step applies unsupervised clustering. The latter uses the widely known *k*-means model.[26] Then, for each of the discovered clusters, one supervised classifier is trained using the following implementations:

- ARFC[27]
- HATC[28]
- HTC[29]
- GNB[30]

Hyperparameter optimisation is performed for the aforementioned ML algorithms. Listings 3, 4, 5 and 6 show the configuration ranges and best values (in bold) for each algorithm.

```
clusters = [10, 20, 30]
models = [50, 100, 200]
features = [50, 100, 200]
lambda = [50, 100, 200]
```

Listing 3 Hyperparameter ranges for the arfc model (best values in bold)

```
clusters = [10, 20, 30]
depth = [50, 100, 200]
tiethreshold = [0.5, 0.05, 0.005]
maxsize = [50, 100, 200]
```

Listing 4 Hyperparameter ranges for the hatc model (best values in bold)

```
clusters = [10, 20, 30]
depth = [50, 100, 200]
tiethreshold = [0.5, 0.05, 0.005]
maxsize = [50, 100, 200]
```

Listing 5 Hyperparameter ranges for the htc model (best values in bold)

[26] Available at https://riverml.xyz/dev/api/cluster/KMeans, June 2023.
[27] Available at https://riverml.xyz/0.11.1/api/ensemble/AdaptiveRandomForestClassifier, June 2023.
[28] Available at https://riverml.xyz/0.11.1/api/tree/HoeffdingAdaptiveTreeClassifier, June 2023.
[29] Available at https://riverml.xyz/0.11.1/api/tree/HoeffdingTreeClassifier, June 2023.
[30] Available at https://riverml.xyz/0.11.1/api/naive-bayes/GaussianNB, June 2023.

```
clusters = [10, 20, 30]
```

Listing 6 Hyperparameter ranges for the gnb model (best value in bold)

Table 5 shows the performance of the ML models. Set A of features includes those in Table 4 except for word *n*-grams, whereas, set B includes set A plus the latter textual features. Finally, set C is composed of set B plus the frequency-based lexicon. The proposed solution exhibits a processing time of 0.42 s/sample in the worst scenario (ARFC model and the set of features A), which can be considered real time.

In light of the results, ARFC exhibits the best performance with all feature sets and for all evaluation metrics. The use of word *n*-grams results in significant improvement across all algorithms. The highest boost occurs for the GNB model (+ 12% percent points in accuracy and micro *F*-measure for the fake class). Despite the promising results, micro *F*-measure values for the target fake class remain under the 70% threshold with feature sets A and B. Finally, the solution reaches accuracy and macro *F*-measure about 80% with all engineered features (set C).

4.3.3 Discussion

Since the majority of the competing works implement batch rather than stream processing and use different data sets, result comparison may not be straightforward. Batch and stream results are only directly comparable if obtained with the same data samples. This means that, ideally, the comparison should be made with a chronologically ordered data set, and the evaluation should consider only the test partition samples. In the case of stream processing, this is achieved by setting the dimension of the sliding window to the number of samples of the test partition and then processing the data set as a stream.

The batch classification works by Zubiaga et al. (2017), Akinyemi et al. (2020) and Ying et al. (2021) explore the same PHEME data set with cross-folded validation, using 80% of the samples for training and 20% for testing. The related online fake news classification system of Ksieniewicz et al. (2020) employs another data set, preventing direct comparison.

Table 6 provides the theoretical comparison results of the most related works together with those of the proposed solution with a sliding window holding 20% of the data (for offline comparison) and a sliding window comprising all data (for online comparison)[31]. The proposed solution with a sliding window of 20% of the data achieves an improvement in macro *F*-measure of 20.12 and 17.42 percent points with respect to the work of Zubiaga et al. (2017) and Jain et al. (2022), respectively. Moreover, it attains + 4.62 percent points in fake *F*-measure compared to Akinyemi et al. (2020). When compared with the batch and online deep learning approaches of Ying et al. (2021) and Ksieniewicz et al. (2020), the proposed solution exhibits slightly lower performance but grants algorithmic transparency with lesser memory and computation time. Finally, for a fair comparison with the most related work by Ksieniewicz et al. (2020), due to the fact the authors provided the implementation of the solution, we were able to run the experiments with the PHEME data set and the accuracy obtained in this regard is 74.10% (−6.16 percent points than our proposal).

Originally, Nikiforos et al. (2020) achieved an accuracy of 99.79% and 99.37% with Naive Bayes and RF offline classifiers, respectively. Both models were trained with a

[31] NA is used to indicate when the competing works did not provide results for specific metrics.

Table 5 Online fake detection results in terms of accuracy, macro and micro *F*-measure (best values in bold) and run-time for the ARFC, HATC, HTC and GNB models by feature set

Set	Classifier	Accuracy	*F*-measure			Time
			Macro	#non-fake	#fake	(s)
A	ARFC	**73.09**	**70.62**	**79.14**	**62.10**	2677.82
	HATC	64.95	63.29	71.10	55.49	8.91
	HTC	64.76	62.79	71.36	54.21	7.45
	GNB	52.95	49.29	62.91	35.68	6.36
B	ARFC	**75.43**	**73.17**	**80.96**	**65.38**	1644.25
	HATC	70.11	66.28	77.65	54.90	29.88
	HTC	69.46	64.59	77.72	51.47	23.25
	GNB	64.09	60.21	72.64	47.79	20.44
C	ARFC	**80.26**	**78.97**	**84.18**	**73.77**	1910.07
	HATC	78.20	76.42	82.91	69.92	299.35
	HTC	77.94	76.11	82.72	69.51	293.74
	GNB	74.66	73.45	79.13	67.76	286.24

synthetic minority over-sampled set generated from 80% of the original data (to overcome the class imbalance of the original data) and tested with the 20% of the original data. To compare with these results, the experiment was repeated with a sliding window comprising 20% of the total number of samples and the best ARFC model. In this case, the current solution attained 99.14% accuracy, macro *F*-measure of 97.54%, and micro *F*-measure of 99.52% and 95.56% for non-fake and fake classes, respectively. This means that the proposed online method achieves, without oversampling and in real time, a comparable accuracy.

4.4 Stream-based explainability module

Figure 2 shows the user explainability dashboard, which aims to make the model outcome comprehensible. In the upper part, it displays the classification of the tweet sample. The user name is *Zone 6 Combatives* and the timezone Canadian. The top center displays the tweet content and the center presents the creator-, content- and context-related features selected by the ML classifier. Feature warnings are shown when a feature deviates from the user average as is the case of *reading ease and time* feature. Otherwise, the features include an OK symbol as in the case of the *5-years post-registration span* feature. The classifier singled out the word *pilot* as relevant. The tweet was classified as fake with an 81% of confidence, according to the `Predict_Proba_One`[32] from River ML library. In the end, the most representative features for both the frequency-based lexicon and the clustering procedure are provided.[33]

The bottom part of the dashboard displays the decision tree path (obtained using `debug one` and `draw`[34] libraries) and the corresponding natural language description. Particularly, the first decision is based on the `surprise` feature (see feature 19 in Table 4). If its

[32] Available at https://riverml.xyz/0.11.1/api/base/Classifier, June 2023.

[33] The sample belongs to cluster 5.

[34] Available at https://riverml.xyz/0.11.1/api/tree/HoeffdingAdaptiveTreeRegressor, June 2023.

Table 6 Fake detection theoretical comparison in terms of accuracy, macro and micro F-measure between related works and the proposed solution

Authorship	Processing	Accuracy	F-measure		
			Macro	#non-fake	#fake
Zubiaga et al. (2017)	Offline	NA[1]	60.70	NA	NA
Akinyemi et al. (2020)	Offline	81.90	78.00	87.00	70.00
Ying et al. (2021)	Offline	87.20	NA	90.40	80.70
Jain et al. (2022)	Offline	NA	63.40	NA	NA
Ksieniewicz et al. (2020)	Online	81.90	NA	NA	NA
Proposed solution	Online[2]	82.82	80.82	87.02	74.62
	Online[3]	80.26	78.97	84.18	73.77
	Offline[4]	99.14	97.54	99.52	95.56

[1] Not available

[2] Sliding window holds 20% of data

[3] Sliding window holds the full data

[4] Sliding window holds 20% of data for the data set provided by Nikiforos et al. (2020)

value is lower or equal to 0.55, the reasoning continues through the left branch. Otherwise it goes to the right branch.

5 Conclusion

Social media is becoming an increasing source of breaking news. In these platforms, information is shared regardless of the context and reliability of the content and creator of the posted information. This instant news dissemination and consumption model easily propagates fake news, constituting a challenge in terms of transparency, reliability, and real-time processing. Accordingly, the proposed solution addresses transparency through explanations, reliability through fake news detection, and real-time processing through incremental profiling and learning. The motivation for the current work relies on the early detection, isolation and explanation of misinformation, all of them crucial procedures to increase the quality and trust in digital media social platforms.

More in detail, this work contributes with an explainable classification method to recognise fake news in real-time. The proposed method combines both unsupervised and supervised approaches with online created lexica. Specifically, it comprises (i) stream-based data processing (through feature engineering, analysis and selection), (ii) stream-based classification (lexicon-based, unsupervised and supervised classification), and (iii) stream-based explainability (prediction confidence and interpretable classification). Furthermore, the profiles are built using creator-, content- and context-based features with the help of NLP techniques. The experimental classification results of 80% accuracy and macro F-measure, obtained with a real data set manually annotated, endorse the promising performance of the designed explainable real-time fake news detection method.

Analyzing the related work, this proposal is the first to jointly provide stream-based data processing, profiling, classification and explainability. Future work will attempt to mitigate further the impact of fake news within social media by automatically identifying and

Fig. 2 Explainability dashboard comprising: (i) selected features from the content, context, and creator, (ii) the prediction, (iii) representative entries of the frequency-based lexicon and the clustering procedure, and (iv) the decision path and its natural language transcription

isolating potential malicious accounts as well as extend the research to related tasks like stance detection, by exploiting new creator-, content- and context-based features.

Author Contributions Francisco de Arriba-Pérez: Conceptualization, Methodology, Software, Resources, Data Curation, Writing—Original Draft, Writing—Review and Editing. Silvia García-Méndez: Conceptualization, Methodology, Resources, Data Curation, Writing— Original Draft, Writing—Review and Editing. Fátima Leal Methodology, Data Curation, Writing—Original Draft, Writing— Review and Editing. Benedita Malheiro: Conceptualization, Methodology, Validation, Writing—Review and Editing, Supervision. Juan Carlos Burguillo-Rial: Conceptualization, Methodology, Validation, Writing—Review and Editing, Supervision.

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature. This work was partially supported by: (i) Xunta de Galicia grants ED481B-2021-118 and ED481B-2022-093, Spain; (ii) Portuguese national funds through FCT – Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) – as part of project UIDB/50014/2020; and (iii) University of Vigo/CISUG for open access charge.

Availability of data and material The used data is openly available.

Code Availability The code will become available at Github upon acceptance of the manuscript.

Declarations

Conflict of interest The authors have no competing interests to declare that are relevant to the content of this article.

References

Akinyemi, B., Adewusi, O., & Oyebade, A. (2020). An improved classification model for fake news detection in social media. *International Journal of Information Technology and Computer Science, 12*(1), 34–43. https://doi.org/10.5815/ijitcs.2020.01.05

Aphiwongsophon, S., & Chongstitvatana, P. (2018). Detecting fake news with machine learning method. In *Proceedings of the international conference on electrical engineering/electronics, computer, telecommunications and information technology* (pp. 528–531). IEEE. https://doi.org/10.1109/ECTICon.2018.8620051

Bifet, A., & Gavaldà, R. (2009). Adaptive learning from evolving data streams. In *Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics)* (vol. 5772, LCNS, pp. 249–260). Springer. https://doi.org/10.1007/978-3-642-03915-7_22

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

Castillo, C., Mendoza, M., & Poblete, B. (2011). Information credibility on twitter. In *Proceedings of the international conference on world wide web* (pp. 675–684). Association for Computing Machinery. https://doi.org/10.1145/1963405.1963500

Choraś, M., Demestichas, K., Gielczyk, A., Herrero, Á., Ksieniewicz, P., Remoundou, K., Urda, D., & Woźniak, M. (2021). Advanced machine learning techniques for fake news (online disinformation) detection: A systematic mapping study. *Applied Soft Computing, 101*, 107050–107064. https://doi.org/10.1016/j.asoc.2020.107050

Dong, X., Victor, U., & Qian, L. (2020). Two-path deep semisupervised learning for timely fake news detection. *IEEE Transactions on Computational Social Systems, 7*(6), 1386–1398. https://doi.org/10.1109/TCSS.2020.3027639

Dulac-Arnold, G., Levine, N., Mankowitz, D. J., Li, J., Paduraru, C., Gowal, S., & Hester, T. (2021). Challenges of real-world reinforcement learning: Definitions, benchmarks and analysis. *Machine Learning, 110*(9), 2419–2468. https://doi.org/10.1007/s10994-021-05961-4

Galli, A., Masciari, E., Moscato, V., & Sperlí, G. (2022). A comprehensive Benchmark for fake news detection. *Journal of Intelligent Information Systems, 59*(1), 237–261. https://doi.org/10.1007/s10844-021-00646-9

Gama, J., Sebastião, R., & Rodrigues, P. P. (2013). On evaluating stream learning algorithms. *Machine Learning, 90*(3), 317–346. https://doi.org/10.1007/s10994-012-5320-9

García-Méndez, S., Fernández-Gavilanes, M., Costa-Montenegro, E., Juncal-Martínez, J., González-Castaño, F. J., & Reiter, E. (2019). A system for automatic english text expansion. *IEEE Access, 7*, 123320–123333. https://doi.org/10.1109/ACCESS.2019.2937505

Goindani, M., Neville, J. (2019). Social reinforcement learning to combat fake news spread. In *Proceedings of the conference on uncertainty in artificial intelligence* (pp. 1006–1016). Association for Uncertainty in Artificial Intelligence.

Gomes, H. M., Bifet, A., Read, J., Barddal, J. P., Enembreck, F., Pfharinger, B., Holmes, G., & Abdessalem, T. (2017). Adaptive random forests for evolving data stream classification. *Machine Learning, 106*(9–10), 1469–1495. https://doi.org/10.1007/s10994-017-5642-8

Hu, C., Xu, Z., Liu, Y., Mei, L., Chen, L., & Luo, X. (2014). Semantic link network-based model for organizing multimedia big data. *IEEE Transactions on Emerging Topics in Computing, 2*(3), 376–387. https://doi.org/10.1109/TETC.2014.2316525

Jain, D. K., Kumar, A., & Shrivastava, A. (2022). CanarDeep: A hybrid deep neural model with mixed fusion for rumour detection in social data streams. *Neural Computing and Applications, 34*, 15129–15140. https://doi.org/10.1007/s00521-021-06743-8

Jang, Y., Park, C. H., Lee, D. G., & Seo, Y. S. (2021). Fake news detection on social media a temporal-based approach. *Computers, Materials & Continua, 69*(3), 3563–3579. https://doi.org/10.32604/cmc.2021.018901

Kozik, R., Kula, S., Choraś, M., & Woźniak, M. (2022). Technical solution to counter potential crime: Text analysis to detect fake news and disinformation. *Journal of Computational Science, 60*, 101576–101582. https://doi.org/10.1016/j.jocs.2022.101576

Ksieniewicz, P., Zyblewski, P., Choraś, M., Kozik, R., Giełczyk, A., Woźniak, M. (2020). Fake news detection from data streams. In *Proceedings of the international joint conference on neural networks* (pp. 1–8). IEEE. https://doi.org/10.1109/IJCNN48605.2020.9207498

Li, D., Guo, H., Wang, Z., & Zheng, Z. (2021). Unsupervised fake news detection based on autoencoder. *IEEE Access, 9*, 29356–29365. https://doi.org/10.1109/ACCESS.2021.3058809

Liu, Y., & Wu, Y. F. B. (2020). FNED: A deep network for fake news early detection on social media. *ACM Transactions on Information Systems, 38*(3), 1–33. https://doi.org/10.1145/3386253

Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. In *Proceedings of the international conference on neural information processing systems* (pp. 4768–4777). Curran Associates Inc. https://doi.org/10.5555/3295222.3295230

Mahajan, A., Shah, D., Jafar, G. (2021). Explainable AI approach towards toxic comment classification. In *Proceedings of the emerging technologies in data mining and information security conference* (pp. 849–858). Springer. https://doi.org/10.1007/978-981-33-4367-2_81

Martens, B., Aguiar, L., Gomez, E., & Mueller-Langer, F. (2018). The digital transformation of news media and the rise of disinformation and fake news. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.3164170

Mathew, A., Amudha, P., & Sivakumari, S. (2021). *Deep learning techniques: An overview* (Vol. 1141). Springer. https://doi.org/10.1007/978-981-15-3383-9_54

Mosallanezhad, A., Karami, M., Shu, K., Mancenido, M. V., & Liu, H. (2022). Domain adaptive fake news detection via reinforcement learning. In *Proceedings of the ACM web conference* (pp. 3632–3640). Association for Computing Machinery. https://doi.org/10.1145/3485447.3512258

Nasir, J. A., Khan, O. S., & Varlamis, I. (2021). Fake news detection: A hybrid CNN-RNN based deep learning approach. *International Journal of Information Management Data Insights, 1*(1), 100007–100019. https://doi.org/10.1016/j.jjimei.2020.100007

Nikiforos, M. N., Vergis, S., Stylidou, A., Augoustis, N., Kermanidis, K. L., & Maragoudakis, M. (2020). *Fake news detection regarding the Hong Kong events from tweets, IFIP* (Vol. 585). Springer. https://doi.org/10.1007/978-3-030-49190-1_16

Pham, X. C., Dang, M. T., Dinh, S. V., Hoang, S., Nguyen, T. T., & Liew, A. W. C. (2017). Learning from data stream based on random projection and Hoeffding tree classifier. In *Proceedings of the international conference on digital image computing: Techniques and applications* (Vol. 2017-Decem, pp. 1–8). IEEE. https://doi.org/10.1109/DICTA.2017.8227456

Puraivan, E., Godoy, E., Riquelme, F., & Salas, R. (2021). Fake news detection on Twitter using a data mining framework based on explainable machine learning techniques. In: *Proceedings of the international conference of pattern recognition systems* (pp. 157–162). Institution of Engineering and Technology. https://doi.org/10.1049/icp.2021.1450

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). Why should I trust you explaining the predictions of any classifier. In *Proceedings of the ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 1135–1144). Association for Computing Machinery. https://doi.org/10.1145/2939672.2939778

Shu, K. (2022). Combating disinformation on social media: A computational perspective. *BenchCouncil Transactions on Benchmarks, Standards and Evaluations, 2*(1), 100035–100040. https://doi.org/10.1016/j.tbench.2022.100035

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

Shu, K., Cui, L., Wang, S., Lee, D., & Liu, H. (2019a). dEFEND: Explainable fake news detection. In *Proceedings of the ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 395–405). Association for Computational Linguistics. https://doi.org/10.1145/3292500.3330935

Shu, K., Wang, S., Liu, H. (2019b). Beyond news contents the role of social context for fake news detection. In *Proceedings of the ACM international conference on web search and data mining* (pp. 312–320). Association for Computing Machinery. https://doi.org/10.1145/3289600.3290994

Silva, A., Han, Y., Luo, L., Karunasekera, S., & Leckie, C. (2021a). Fake news detection on social media a data mining perspective. *Information Processing & Management, 58*(5), 102618–102634. https://doi.org/10.1016/j.ipm.2021.102618

Silva, C. V. M., Fontes, R. S., & Júnior, M. C. (2021b). Intelligent fake news detection: A systematic mapping. *Journal of Applied Security Research, 16*(2), 168–189. https://doi.org/10.1080/19361610.2020.1761224

Silva, R. M., Santos, R. L., Almeida, T. A., & Pardo, T. A. S. (2020). Towards automatically filtering fake news in Portuguese. *Expert Systems with Applications, 146*, 113199–113212. https://doi.org/10.1016/j.eswa.2020.113199

Sinaga, K. P., & Yang, M. S. (2020). Unsupervised K-means clustering algorithm. *IEEE Access, 8*, 80716–80727. https://doi.org/10.1109/ACCESS.2020.2988796

Škrlj, B., Martinc, M., Lavrač, N., & Pollak, S. (2021). autoBOT: Evolving neuro-symbolic representations for explainable low resource text classification. *Machine Learning, 110*(5), 989–1028. https://doi.org/10.1007/s10994-021-05968-x

Song, C., Shu, K., & Wu, B. (2021). Temporally evolving graph neural network for fake news detection. *Information Processing & Management, 58*(6), 102712–102729. https://doi.org/10.1016/j.ipm.2021.102712

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

Tandoc, E. C. (2019). The facts of fake news: A research review. *Sociology Compass, 13*(9), 12724–12732. https://doi.org/10.1111/soc4.12724

Vicario, M. D., Quattrociocchi, W., Scala, A., & Zollo, F. (2019). Polarization and fake news early warning of potential misinformation targets. *ACM Transactions on the Web, 13*(2), 1–22. https://doi.org/10.1145/3316809

Vouros, A., Langdell, S., Croucher, M., & Vasilaki, E. (2021). An empirical comparison between stochastic and deterministic centroid initialisation for K-means variations. *Machine Learning, 110*(8), 1975–2003. https://doi.org/10.1007/s10994-021-06021-7

Wang, Y., Yang, W., Ma, F., et al. (2020). Weak supervision for fake news detection via reinforcement learning. *Proceedings of the AAAI Conference on Artificial Intelligence, 34*, 516–523. https://doi.org/10.1609/aaai.v34i01.5389

Xiao, Y., Li, W., Qiang, S., Li, Q., Xiao, H., & Liu, Y. (2020). A rumor & anti-rumor propagation model based on data enhancement and evolutionary game. *IEEE Transactions on Emerging Topics in Computing, 10*(2), 690–703. https://doi.org/10.1109/TETC.2020.3034188

Xue, Q., Zhu, Y., & Wang, J. (2021). Joint distribution estimation and Naïve Bayes classification under local differential privacy. *IEEE Transactions on Emerging Topics in Computing, 9*(4), 2053–2063. https://doi.org/10.1109/TETC.2019.2959581

Ying, L., Yu, H., Wang, J., Ji, Y., & Qian, S. (2021). Multi-level multi-modal cross-attention network for fake news detection. *IEEE Access, 9*, 132363–132373. https://doi.org/10.1109/ACCESS.2021.3114093

Zhao, Z., Zhao, J., Sano, Y., Levy, O., Takayasu, H., Takayasu, M., Li, D., Wu, J., & Havlin, S. (2020). Fake news propagates differently from real news even at early stages of spreading. *EPJ Data Science, 9*(1), 7–20. https://doi.org/10.1140/epjds/s13688-020-00224-z

Zhou, X., Jain, A., Phoha, V. V., & Zafarani, R. (2020). Fake news early detection a theory-driven model. *Digital Threats Research and Practice, 1*(2), 1–25. https://doi.org/10.1145/3377478

Zubiaga, A., Liakata, M., & Procter, R. (2017). Exploiting context for rumour detection in social media. In *Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics)* (vol. 10539, LNCS, pp. 109–123). Springer. https://doi.org/10.1007/978-3-319-67217-5_8

Authors and Affiliations

Francisco de Arriba-Pérez[1] · Silvia García-Méndez[1] · Fátima Leal[2] · Benedita Malheiro[3,4] · Juan Carlos Burguillo[1]

✉ Francisco de Arriba-Pérez
farriba@gti.uvigo.es

Silvia García-Méndez
sgarcia@gti.uvigo.es

Fátima Leal
fatimal@upt.pt

Benedita Malheiro
mbm@isep.ipp.pt

Juan Carlos Burguillo
J.C.Burguillo@uvigo.es

[1] atlanTTic, University of Vigo, Information Technologies Group, Campus Universitario de Vigo, Lagoas-Marcosende, 36310 Vigo, Spain

[2] REMIT, Universidade Portucalense, Rua Dr. António Bernardino de Almeida, 4200-072 Porto, Portugal

[3] ISEP, Polytechnic of Porto, Rua Dr. António Bernardino de Almeida, 4249-015 Porto, Portugal

[4] INESC TEC, Campus da Faculdade de Engenharia da Universidade do Porto, Rua Dr. Roberto Frias, 4200-465 Porto, Portugal