



A network-based positive and unlabeled learning approach for fake news detection

Mariana Caravanti de Souza, et al. [full author details at the end of the article]

Received: 4 March 2021 / Revised: 19 October 2021 / Accepted: 21 October 2021 /
Published online: 18 November 2021

© The Author(s), under exclusive licence to Springer Science+Business Media LLC, part of Springer Nature 2021

Abstract

Fake news can rapidly spread through internet users and can deceive a large audience. Due to those characteristics, they can have a direct impact on political and economic events. Machine Learning approaches have been used to assist fake news identification. However, since the spectrum of real news is broad, hard to characterize, and expensive to label data due to the high update frequency, One-Class Learning (OCL) and Positive and Unlabeled Learning (PUL) emerge as an interesting approach for content-based fake news detection using a smaller set of labeled data than traditional machine learning techniques. In particular, network-based approaches are adequate for fake news detection since they allow incorporating information from different aspects of a publication to the problem modeling. In this paper, we propose a network-based approach based on Positive and Unlabeled Learning by Label Propagation (PU-LP), a one-class and transductive semi-supervised learning algorithm that performs classification by first identifying potential interest and non-interest documents into unlabeled data and then propagating labels to classify the remaining unlabeled documents. A label propagation approach is then employed to classify the remaining unlabeled documents. We assessed the performance of our proposal considering homogeneous (only documents) and heterogeneous (documents and terms) networks. Our comparative analysis considered four OCL algorithms extensively employed in One-Class text classification (k -Means, k -Nearest Neighbors Density-based, One-Class Support Vector Machine, and Dense Autoencoder), and another traditional PUL algorithm (Rocchio Support Vector Machine). The algorithms were evaluated in three news collections, considering balanced and extremely unbalanced scenarios. We used Bag-of-Words and Doc2Vec models to transform news into structured data. Results indicated that PU-LP approaches are more stable and achieve better results than other PUL and OCL approaches in most scenarios, performing similarly to semi-supervised binary algorithms. Also, the inclusion of terms in the news network activate better results, especially when news are distributed in the feature space considering veracity and subject. News representation using the Doc2Vec achieved better results than the Bag-of-Words model for both algorithms based on vector-space model and document similarity network.

Keywords Fake news · One-class learning · Positive and unlabeled learning · Transductive semi-supervised learning · Graph-based learning · Document management · Text processing

1 Introduction

The advancement of technology, especially mobile technology, has brought a revolution in how users consume information. In a scenario where there is an urge to be continuously updated with recent news, web-based resources, such as news portals and social networks, are important sources of information acquisition and sharing (Deepak et al., 2021; Sharma et al., 2019). This situation raises some uncertainties associated with the credibility and the source of shared information (Osatuyi, 2013; Zhang & Ghorbani, 2020).

Fake news detection research is essential for society, especially because fake news constantly involves writing style and presentation of content, which makes the identification of true and false content difficult even for humans (DePaulo et al., 1997; Frank et al., 2004; Greifeneder et al., 2021; Shu et al., 2017).

Machine Learning algorithms have been used to classify news as real or fake to minimize the spread of deceptive content. The most common way to deal with this problem is to characterize fake news detection as a binary classification problem (Meel & Vishwakarma, 2019; Zhang & Ghorbani, 2020). However, training sets to achieve satisfactory classification performances should cover the entire news spectrum, addressing different subjects, sources, and truth levels. Furthermore, a large and balanced number of labeled fake and real news is necessary to achieve good classification performances, which is not suitable in a constantly changing scenario (Deepak et al., 2021; Sharma et al., 2019).

Considering the challenges of labeling training sets for fake news detection presented before, One-Class Learning (OCL) (Perera et al., 2021; Wang et al., 2019; Khan & Maden, 2014; Tax, 2001) can be a promising approach (Faustini & Covões, 2019). OCL employs labeled examples of the interest class (or positive class) as input to the learning algorithm, contributing to scenarios where it is difficult to label examples of non-interest categories due to the large spectrum (Bellinger et al., 2017). As interesting as OCL approaches for fake news detection, Positive and Unlabeled Learning (PUL) (Bekker & Davis, 2020) can make use of unlabeled data to increase the classification performance. PUL has attracted growing interest in the Machine Learning literature (Bekker & Davis, 2020; Jaskie & Spanias, 2019), and recently it has proven to be an interesting approach for the fake news detection task (Liu & Wu, 2020).

Although OCL and PUL can mitigate the problems of binary learning algorithms to detect fake news, these algorithms are still underexplored for this scenario. Also, besides the benefits of using those learning approaches, the model to represent the data can impair the classification, mainly when making use of unlabeled data (Van Engelen & Hoos, 2020). On the other hand, network-based representation can effectively fuse information across multiple data types, modeling complex objects and their relations, with rich semantics (Shi & Philip, 2017; Yang et al., 2020). Moreover, network representations allow extracting class patterns that are hardly captured by vector space models (Breve et al., 2012) and have been demonstrated to be useful for semi-supervised learning (Rossi et al., 2016). Usually, the learning process in a network is performed by a label propagation algorithm, in which the labeled nodes propagate their label to

the unlabeled ones (Liang & Li, 2018; Zhu & Goldberg, 2009). The iterative procedure applied by label propagation algorithms allows dealing with large volumes of data (Hua et al., 2021; Zhu & Goldberg, 2009).

In this work, we propose an approach based on Positive and Unlabeled Learning by Label Propagation (PU-LP) (Ma & Zhang, 2017), a Positive and Unlabeled Learning algorithm based on a homogeneous network representation and label propagation to identify potential interest and non-interest documents present into the unlabeled data. The PU-LP approach was originally proposed for numerical data and considering a similarity-based network (homogeneous network). Therefore, we adapted PU-LP for text classification and applied it to the fake news detection scenario, considering the class “fake” as the interest class.

Besides the adaptation for text classification, and considering the problem complexity and the fact that enriching the representation model with different types of relations can increase the performance of text classifiers (Deepak et al., 2021; Rossi et al., 2015; Shi & Philip, 2017; Yang et al., 2020), we proposed different network representations than the originally used in Ma and Zhang (2017). In our network proposal, we first generate a similarity-based network considering the news as objects. Then we add and connect representative unigrams and bigrams to objects that represent the news, making a heterogeneous network. We highlight that the relations among documents and terms can be considered generic, since we can extract representative terms from any fake news collection (taken from microblogs, news portals or social networks). Also, we added the terms since: (i) the inclusion of terms in document networks make the labeling process more accurate; (ii) the connections between documents and terms can increase the scores of documents for their real classes; and (iii) the terms are often used to discriminate true and false content (Aggarwal & Li, 2011; Ahmed et al., 2017; Hassan et al., 2020; Mihalcea et al., 2010; Pérez-Rosas et al., 2017; Rossi et al., 2015; Rubin et al., 2016; Yan et al., 2020).

To assess the performance of our approach, we use four OCL algorithms that perform well in the literature for text classification: an approach based on k -Means (Tan et al., 2019), k -Nearest Neighbors Density-based (k -NND) (Tan et al., 2019), One-Class Support Vector Machine (OCSVM) (Manevitz & Yousef, 2001), and Dense Autoencoder (DAE) (Manevitz and Yousef 2007). Also, we used a well-known PUL algorithm: Rocchio Support Vector Machine (RC-SVM) (Li & Liu, 2003). Finally, to assess the performance of OCL and PUL approaches against a scenario in which there are news labeled as real and fake, we also used a semi-supervised binary approach. We name this approach as reference model. These algorithms were applied in three fake news datasets considering different scenarios and languages: two Portuguese datasets with a balanced number of news, one of them covering political issues and the other composed of news on several subjects; and one English dataset with unbalanced data, in which less than 30% are fake news.

In order to generate representations for the OCL and PUL algorithms used in the comparison, we used two textual models that consider the complete text of the news to transform them into structured data: Bag-of-Words (BoW) (Salton, 1989) and document embeddings generated with Doc2Vec (D2V) (Le & Mikolov, 2014). The same representations were used to compute similarities and generate relations among documents in a network. The algorithms were evaluated and compared considering different percentages of labeled fake news: 10%, 20%, and 30%. We evaluated the algorithms considering the macro F_1 and F_1 measure of the interest class (interest- F_1). With our experimental evaluation, we answered the following research questions:

1. **Did the Doc2Vec representation model provide better performance compared to traditional Bag-of-Words?** Doc2Vec provides better overall performance, inferring that representation models grouped real and fake news more efficiently, favoring the performance of OCL and PUL classifiers, especially those based on density and clustering.
2. **Did PUL algorithms perform better than OCL algorithms?** For the FakeNewsNet dataset, with extreme unbalance between classes, OCL activated better interest- F_1 , especially k -NND and k -Means algorithms. However, PUL had a better overall performance for the other news datasets, reaching more than 90% of interest and macro F_1 for the Fact-checked News Policy dataset.
3. **Did the proposed approach outperform other PUL, OCL, and the reference model for fake news detection?** In an average ranking analysis, we show that our proposed approach, which adds representative terms in the news network of PU-LP, achieves better overall performance.
4. **Did the amount of labeled fake news significantly increase the interest- F_1 ?** The results obtained did not vary greatly when considering 10% and 30% of labeled fake news. Therefore, labeling a smaller amount of news becomes more advantageous, especially considering the constant updating of the news scenario, and the difficulty to label large volumes of data.
5. **Which parameters of the proposed approach achieved the best results for interest- F_1 ?** We also present an analysis of the parameters that achieved the best interest- F_1 values considering each dataset and the size of the initially labeled set.

The remaining of this article is organized as follows: Section 2 presents related work. Section 3 presents the background of the OCL and PUL algorithms used for news classification. Section 4 describes the proposed PU-LP approach. Section 5 presents the experimental evaluation, information on news collections, the results, and associated discussions. Finally, Sect. 6 presents conclusions and future works.

2 Related work

This section discusses existing gaps in approaches proposed in the literature for Fake News detection, One-Class Learning, and Positive and Unlabeled Learning. Furthermore, we justify the proposed approach based on PU-LP and we discuss the limitations of existing OCL and PUL algorithms for fake news detection.

2.1 Fake news detection

In order to identify and mitigate the spread of fake news, several automatic approaches have been proposed in the literature in recent years (Bondielli & Marcelloni, 2019; Deepak et al., 2021; Meel & Vishwakarma, 2019; Meneses Silva et al., 2021; Sharma et al., 2019). Those approaches can be divided into three distinct groups: (i) identification of fake news by text content; (ii) news classification methods based on context information; and (iii) intervention-based solutions, that are dedicated to restricting the spread of false information.

In the news texts, group (i) approaches usually analyze lexical, syntactic, and semantic aspects. Lexical approaches analyze words used in textual content with n -grams (Ahmed

et al., 2017; Hassan et al., 2020). For example, n -grams can indicate humor, abbreviations, or useful expressions to determine the veracity of the information (Mihalcea et al., 2010; Rubin et al., 2016). Syntactic characteristics involve word counts (like subjects, adjectives, and verbs), presence and frequency of specific patterns, and analysis of text readability (Pérez-Rosas et al., 2017; Reis et al., 2019; Volkova & Jang, 2018). Indicators such as pausality, uncertainty, and redundancy (Silva et al., 2020; Zhou et al., 2004b) can also be explored. Semantic characteristics are related to the meanings expressed by the terms. Thus, advanced Natural Language Processing (NLP) techniques, based on Deep Learning, have become widely used to represent terms and documents like *embeddings* (Aggarwal, 2018; Kang et al., 2020; Li et al., 2019; Wang et al., 2018).

Works related to the group (ii) try to extract relevant characteristics considering the news's entire environment, such as publication source, post metadata, and users involved (Bondielli & Marcelloni, 2019). Lastly, works related to group (iii) (Barbier et al., 2013; Zhao et al., 2020; Shu et al., 2019) compare patterns of news spread, usually using structures of homogeneous networks (relationships, time diffusion credibility networks) and heterogeneous networks (relationships between different types of entities, such as users, posts, and news) to identify influential users.

Recent studies suggest that fake news content tends to be short, repetitive, and uninformative (Horne & Adali, 2017; Pérez-Rosas et al., 2017), in which social words tend to be used more often. In addition, fake news may focus on the present and future tenses, with less objectivity and a significant presence of adverbs, verbs, and punctuation characters. Studies also show a higher incidence of perceptive words and personal pronouns in the celebrity domain (Pérez-Rosas et al., 2017). Also, first and second-person pronouns and exaggerated words are used more frequently in misleading news. In contrast, reliable news writers try to be impartial, using assertive words and offering concrete data (Rashkin et al., 2017).

The vast majority of the work that analyzes textual content for fake news detection employs binary classification (Bondielli & Marcelloni, 2019; Meel & Vishwakarma, 2019; Meneses Silva et al., 2021; Sharma et al., 2019). As previously discussed, this approach requires a large and balanced set of labeled news to achieve satisfactory performance. Also, the algorithms are evaluated considering datasets that do not reflect real-world scenarios - considering balanced scenarios, for example. Therefore, there is also a need for approaches based on semi-supervised learning that minimize the effort of news labeling, the lack of algorithms that manipulate vast volumes of data, or prove their effectiveness in different news datasets (Zhang & Ghorbani, 2020).

To avoid the effort of labeling real news, Faustini and Covões (2019) was the first paper to identify deceptive content using One-Class-based algorithms. They proposed the DCDistanceOCC algorithm, in which the news is classified as fake if its distance to the class vector is below a threshold. For each news item, linguistic features were extracted, such as the proportion of capital letters and the number of words per sentence. The performance of the DCDistanceOCC was compared to EcoOCC (k -Means-based) (Salmazzo, 2016), Naïve Bayes Positive Class (Datta, 1998) and One-Class SVM (Manevitz & Yousef, 2001). The algorithms were performed using 90% of fake news for training. The results ranged from 54% to 67% of F_1 for fake data, especially considering the Fake.BR news dataset (Silva et al., 2020). Other authors have used OCL to find rumors on social networks, treating them as anomalies. Chen et al. (2016) performed a factor analysis of mixed data using Euclidean distance and cosine similarity to describe the deviation degree of documents in rumor datasets, reaching 68 to 81% of precision. Also, Zhao et al. (2014) present an interactive visualization system for analyzing anomalous information spreading. They

measure patterns of information dissemination, assigning anomaly scores to unlabeled data with One Class Conditional Random Fields.

Liu and Wu (2020) propose the Fake News Early Detection (FNED), in which a status-sensitive crowd response feature extractor extracts textual and user features, and an attention mechanism highlights important user responses. The method detects fake news with 90% of accuracy, using 10% of labeled fake news (set P), and a PUL approach. From the unlabeled set, pseudo-true news are randomly selected, composing set N' , with $|N'| = |P|$. These two sets are used to train a neural classifier. The process is repeated k times, and all the trained classifiers are joined to form a final classifier. The final classifier labels the unlabeled news set, and the top n news classified as fake increments the set P . The approach is evaluated on two datasets, containing 680 and 4664 news.

Some related works use network-based representation models for news classification. De Sales Santos and Pardo (2020) propose a knowledge network for fact-checking, in which the entry is a triple (subject-predicate-object). The graph is constructed with information from Wikipedia, in which given a sentence, it is considered authentic if its triple exists within the knowledge graph. The approach achieves approximately 74% accuracy. Guacho et al. (2018) propose a network for the classification of news and malicious users. News are represented as embeddings, and the Fast Belief Propagation algorithm is used to propagate labeled news information to unlabeled news according to its proximity relationships. The authors achieve 75.43% of accuracy using 30% of real and fake news, and two small datasets containing approximately 150 news items each.

Despite the benefits of network representations (see Sect. 1), semi-supervised network-based algorithms for OCL, or Positive and Unlabeled Learning, have been little explored in the literature for detecting fake news. In the next sections, we discuss some OCL and PUL approaches proposed in the last years for text classification, and their limitations for the news classification context.

2.2 One-class learning

One-Class Learning, Concept Learning, Outlier Detection, or Anomaly Detection are different terms resulting from applications in which learning models are inferred considering only examples of the interest class (Khan & Madden, 2014; Perera et al., 2021; Tax, 2001; Wang et al., 2019). One-Class classifiers must describe a set of objects and determine which new objects are similar to this set. In this approach, the training examples have high representativeness of the interest class, called objects of interest, while non-interest objects may be limited or absent, considered anomalies or outliers.

There are different algorithms proposed in the literature for OCL that address distance, clustering, density, ensembles, and learning-based methods divided into Active Learning (Pimentel et al., 2018), graphs (Eltanbouly et al., 2020; Akoglu et al., 2015), and Deep Learning (Chalapathy & Chawla, 2019; Manevitz and Yousef 2007). However, there are still few studies that apply OCL for text classification (Alam et al., 2020; Khan & Madden, 2014; Perera et al., 2021; Wang et al., 2019).

In Gôlo et al. (2020), OCL algorithms were also extensively employed and evaluated in text classification. Three of the algorithms with the highest classification performances were k -Means (Tan et al., 2019), k -Nearest Neighbors Density-based (k -NND) (Tan et al., 2019), and One-Class Support Vector Machine (OCSVM) (Manevitz & Yousef, 2001). K -NND assigns a score to a new text through the average of the similarities with the k -Nearest Neighbors, and k -Means assigns a score considering the lowest

similarity with a centroid group. A score is compared with a threshold to define if a new text belongs to the interest class. The algorithm OCSVM treats the origin or points near the origin as examples of the non-interest class (or negative class), and then the standard two-class SVM algorithm is employed. The results presented in Gôlo et al. (2020) also demonstrate that *tf-idf* was the term-weighting scheme that provided the best classification results, and that the dimensionality reduction techniques do not necessarily have a positive impact on the classification performance.

Manevitz and Yousef (2007) proposed a Dense Autoencoder to filter examples of interest in a document collection through a simple Neural Network feed-forward. The method demonstrated superior behavior to conventional Naïve Bayes, OCSVM, and methods based on nearest neighbors.

Ruff et al. (2019) proposed an anomaly detection approach based on word embeddings to learn sentence representation by attention mechanisms, called Context Vector Data Description (CVDD). The method uses training sets with approximately 100 examples from each class of the *Reuters* dataset. One class represents the interest data, and the others the anomalous examples. AUC results show that the proposed method is competitive to OCSVM.

Lazhar (2019) used *Fuzzy* clustering to detect anomalous text documents. The method assumes that documents assigned to different groups with very close percentages are candidates for being outliers. After removing anomalous documents, Naïve Bayes and SVM classifiers' accuracy achieved a slight improvement.

Sonbhadra et al. (2020) proposed extracting trends from 45,000 Covid-19 articles containing 75 categories to help the scientific community explore prevention and treatment techniques. Clustering algorithms, such as *k*-Means, were used to cluster articles considering similar subjects. Each cluster was individually trained with OCSVM, which associates the most appropriate articles with the required information in a search string. The approach reached more than 89% of F_1 score.

Although these algorithms have good performance in the literature for multidimensional data classification, they have some limitations if used in the context of news classification. Table 1 present these limitations. *k*-Means, *k*-NND, and OCSVM are highly dependent on parameters and the used datasets. For Dense autoencoder, the number of parameters to be learned can be very high. CVDD does not present many gains over the OCSVM model. Fuzzy clustering is designed to improve the performance of binary classifiers, but the results are not significantly improved. The proposed approach in Sonbhadra et al. (2020) is evaluated considering only different clustering algorithms but it is not compared with other OCL algorithms in the literature. Also, recent works highlight the following gaps and challenges considering high dimensional datasets and OCL (Perera et al., 2021; Wang et al., 2019): (i) need for new semi-supervised OCL approaches that perform well in multiple datasets; (ii) examine the influence of characteristics that differ in the training examples to select those appropriate for anomaly detection, carrying out experimental evaluation with different characteristics; and (iii) most algorithms need thresholds to be set for classifying examples, which is difficult in real-world scenarios.

Among the algorithms, we selected four algorithms that have been demonstrated competitive results for text classification to be compared with PU-LP: *k*-Means, *k*-Nearest Neighbors Density-based (*k*-NND), One-Class Support Vector Machine (OCSVM), and Dense Autoencoder (DAE). The results obtained will be used for the evaluation of the proposed PUL approach based on PU-LP. More details are discussed in Sect. 3.

Table 1 Limitations of OCL and PUL approaches for the fake news detection

Citation	Type	Model	Dataset Language	Limitations for Fake News Detection
DCDistanceOCC (Faustini & Covões, 2019)	OCL	vector-space	Portuguese	Threshold calibration.
<i>k</i> -Means (Tan et al., 2019; Gólo et al., 2020)	OCL	vector-space	English	Results are strongly dependent on the number of groups, the shape of the groups, and the thresholds used to assign an example to a group.
<i>k</i> -NND (Tan et al., 2019; Gólo et al., 2020)	OCL	vector-space	English	Define the optimal value of <i>k</i> and the calibration of the threshold.
OCSVM (Manevitz & Yousef, 2001)	OCL	vector-space	English	Results proved to be very sensitive to parameters, especially to the kernel chosen and the dimension of the representation model.
Dense Autoencoder (Manevitz & Yousef, 2007)	OCL	vector-space	English	Defining an adequate threshold for the interest class, number of parameters to be trained, and the algorithm's performance relies on unlabeled documents of the class of interest to be very similar to the initially labeled set.
Context Vector Data Description (Ruff et al., 2019)	OCL	vector-space	English	Two datasets are analyzed and the algorithms' results are similar to OCSVM.
Fuzzy clustering (Lazhar 2019)	OCL	vector-space	English	The method is used only to find outliers in textual datasets to improve classification results, and it does not apply to identifying interest-class elements.
Clustering with OCSVM (Sonbhadra et al., 2020)	OCL	vector-space	English	The approach is evaluated considering different clustering algorithms, but is not compared to other OCL algorithms in the literature.
RC-SVM (Li & Liu, 2003)	PUL	vector-space	English	The iterative application of SVM (self-training) increases the computational cost and can increase the classification performance.
FNED (Liu & Wu, 2020)	PUL	vector-space	English and Chinese	Although the model performs well for both tested news datasets, the model's uninterpretability and unexplainability is a limitation for news classification, the iterative training of a neural network causes a high increase in the computational cost.

Table 1 (continued)

Citation	Type	Model	Dataset Language	Limitations for Fake News Detection
S-EM (Liu et al., 2002)	PUL	vector-space	English	S-EM only performs well with very few interest examples in the unlabeled set. As Naïve Bayes tolerates a small noise in the training set, it tends to classify most documents in the unlabeled set as negative.
PE-PUC (Yu & Li, 2007)	PUL	vector-space and network	English	If the set of interest documents initially labeled is too small, the set is not sufficient to represent the distribution of the interest class and most of the examples in the unlabeled set are classified as non-interest.
PUL-SAAT (Xu et al., 2019)	PUL	vector-space	English	The approach is compared only with binary supervised algorithms, and its performance highly depend on the random perturbation value.

2.3 Positive and unlabeled learning

PUL algorithms learn models considering a set of interest labeled and unlabeled documents to train a classifier (inductive semi-supervised learning) or to classify unlabeled known documents (transductive semi-supervised learning) (Bekker & Davis, 2020; Li & Liu, 2003; Liu et al., 2002). The purpose of using unlabeled documents during learning is to improve the classification performance, as in multiclass semi-supervised learning (Zhu & Goldberg, 2009). Besides, since the unlabeled documents are easy to collect and a user has to label only a few documents of the interest class, PUL has gained attention in the last years (Bekker & Davis, 2020; Jaskie & Spanias, 2019; Li et al., 2014; Ma & Zhang, 2017; Zhang et al., 2019).

The most common PUL approaches are those which perform learning in two stages. In the first stage, a set of non-interest documents is generated by the extraction of reliable outlier examples. In addition, the set of interest examples can be increased with reliable interest examples. Once there are interest, non-interest, and unlabeled examples, an inductive or transductive learning algorithm is applied to infer the label of the remaining unlabeled examples in the second stage (Bekker & Davis, 2020; Li & Liu, 2003; Liu et al., 2002).

One of the most intuitive PUL algorithms used for text classification is the RC-SVM, proposed by Li and Liu (2003). The algorithm extracts reliable outlier documents from unlabeled data using Rocchio (Salton et al., 1971) and applies Support Vector Machine (SVM) iteratively to build and select a classifier. The algorithm was applied to the Reuters dataset, containing 21,578 text documents, reaching an average accuracy of 97%. Li and Liu (2003) also propose a version of RC-SVM with clustering. A cluster partitions the set of reliable outliers into groups of similar documents. Rocchio then builds a classifier that considers each cluster and the set of interest labeled documents to identify likely positive documents in the clusters and delete them. RC-SVM and clustering RC-SVM perform consistently considering different percentages of labeled documents. However, RC-SVM is more efficient for its simplicity. Other PUL algorithms perform similar processes to classify texts, like Mapping Convergence (Manevitz & Yousef, 2001) and Spy-Expectation-Maximization (S-EM) (Liu et al., 2002).

Liu et al. (2002) proposed the Spy-Expectation-Maximization (S-EM) algorithm. In this approach, some documents of the interest class (spy) are added to the set of unlabeled documents. A classification model through Naïve Bayes is built and the probabilities assigned to the spy documents are used to set thresholds in order to infer reliable negative documents. Then, the EM algorithm is used to classify the remaining unlabeled documents. The author applied the approach in 30 sets of public domain documents, reaching an average accuracy of 92%.

Xu et al. (2019) proposed a PUL algorithm using adversary training and Long short-term memory (LSTM) for sentiment analysis in more recent works. From unlabeled examples, positive and potential negative reviews are extracted by the Rocchio algorithm. To minimize the risk of incorrect document labeling, the author adds a random perturbation to the document representations using Gaussian distribution. The other examples are classified with LSTM. For two datasets, 20 to 40% of positive reviews made up the labeled set, while the rest of the reviews were considered unlabeled. The algorithm achieves 65 to 83% of macro-averaging F_1 .

The PUL algorithms presented above are based on vector-space representations, such a self-training employed by RC-SVM or the Expectation-Maximization applied by

S-EM. Both approaches have demonstrated lower performances than other approaches in a multi-class learning scenario for text classification (Rossi et al., 2014, 2016). S-EM only performs well with very few interest examples in the unlabeled set. Thus, NB assumes that all unlabeled examples are outliers, as it tolerates a small amount of noise in the training set. Xu et al. (2019) does not compare their proposal with other PUL approaches. Furthermore, the algorithm's success highly depends on the proper choice of the random perturbation.

To address the limitations of approaches based on the space-vector model, network representations have been demonstrated to be good alternatives (Rossi et al., 2016, 2015; Shi & Philip, 2017; Yang et al., 2020). Even so, there are few network-based PUL algorithms, such as *Positive documents Enlarging PU Classifier* (PE-PUC) (Yu & Li, 2007) and *Positive and Unlabeled Learning by Label Propagation* (PU-LP) (Ma & Zhang, 2017). PE-PUC uses NB for extracting reliable negative examples. Unlabeled examples are all considered outliers, while the initially labeled set represents the class of interest. The model learned by NB classifies the unlabeled examples and those classified as negative form the set of reliable negative. PE-PUC uses network representations to increase the set of interest documents with reliable positive documents. The network is not used in the classification step. A limitation of the PE-PUC is that if the set of interest documents initially labeled is too small, the set is not sufficient to represent the distribution of the interest class. So, most of the the unlabeled documents will be classified as reliable non-interest documents, which limits the performance of the algorithm. On the other hand, PU-LP is an algorithm entirely based on network. In PU-LP, interest and non-interest documents are extracted considering an interest labeled set and a global path-based similarity measure, which causes vertices with many common neighboring nodes to be classified with the same label. Then a label propagation approach is employed to classify the remaining unlabeled documents. PU-LP performs well in numerical datasets using only 10% of labeled data, in addition to not requiring a threshold to be defined for classifying examples of interest.

According to the limitations presented in Table 1 and the limitations presentend in Becker and Davis (2020); Jaskie and Spanias (2019), we can observe that: (i) few works in the literature apply PUL for real-world tasks; (ii) empirical comparisons of PUL approaches are needed, showing which assumptions are reasonable to obtain good performance in practice; (iii) algorithms do not perform well when the unlabeled set has a large number of examples of the interest class, or when the initially labeled interest set has a limited number of labeled data; and (iv) there is a lack of relational algorithms proposed for PUL that are adequate in the context of news classification represented in multidimensional data.

So, since graph-based semi-supervised learning has been demonstrated to be effective to make use of multidimensional unlabeled data to improve the classification performance, allows the modeling of different objects and relations with rich semantics, achieve satisfactory classification performances even with few labeled data in a semi-supervised learning scenario, and does not require setting thresholds (Rossi et al., 2016; Shi & Philip, 2017), in this article we propose an approach based on PU-LP applied to fake news detection (see Sect. 4).

3 OCL and PUL algorithms for fake news detection

In this section we present the definitions of OCL and PUL, and the details of the algorithms used in the experimental evaluation. One-Class Learning (OCL) algorithms learn with examples labeled with a single class, usually the interest class (also treated as a positive

class) (Khan & Madden, 2014; Tax, 2001). Thus, the set of training documents is $\mathcal{D} = \mathcal{D}^+$, in which \mathcal{D}^+ corresponds to the set of interest documents. In this work, the interest class is “fake”. After learning a classification model, OCL algorithms classify a new document \mathbf{d}_i according to the assigned score $f(\mathbf{d}_i)$, as presented in Eq. 1:

$$\mathcal{C} = \begin{cases} f(\mathbf{d}_i) \geq \text{threshold} \rightarrow \text{interest} \\ f(\mathbf{d}_i) < \text{threshold} \rightarrow \text{not interest} \end{cases} \tag{1}$$

in which \mathcal{C} corresponds to the set of class labels. As mentioned before, we will use the OCL algorithms k -Means, k -Nearest Neighbors Density-based (k -NND), and One-Class Support Vector Machine (OCSVM). Both k -Means and k -NND are similarity-based algorithms. The score assigned by k -Means to a new document \mathbf{d}_i is given by the similarity to the closest centroid group, i.e., (Tan et al., 2019):

$$f(\mathbf{d}_i) = \max_{\mathcal{G}_j \subset \mathcal{D}} \text{cos}(\mathbf{d}_i, \mathbf{g}_j), \tag{2}$$

in which $\mathcal{D} = \mathcal{G}_1 \cup \mathcal{G}_2 \cup \dots \cup \mathcal{G}_k$, $\mathcal{G}_i \subset \mathcal{D}$ is a group of texts, and \mathbf{g}_j is the centroid of group \mathcal{G}_j .

k -NND assigns a score to a new text through the average of the similarities with the k -Nearest Neighbors, i.e., (Tan et al., 2019):

$$f(\mathbf{d}_i) = \sum_{\mathbf{d}_j \in \mathcal{N}_{(\mathbf{d}_i, k)}} \text{sim}(\mathbf{d}_i, \mathbf{d}_j) / |\mathcal{N}_{(\mathbf{d}_i, k)}|, \tag{3}$$

in which $\mathcal{N}_{(\mathbf{d}_i, k)}$ is the set of the k nearest neighbors of \mathbf{d}_i , and $\text{sim}(\mathbf{d}_i, \mathbf{d}_j)$ returns the similarity between the documents \mathbf{d}_i and \mathbf{d}_j .

One-Class Support Vector Machine (OCSVM) (Manevitz & Yousef, 2001) generates fictitious points near the origin and considers them as points from the non-interest class. Then, a maximal margin hyperplane is generated, such as in SVM. The optimization function to obtain the maximal margin hyperplane is presented in Eq. 4, in which \mathbf{h} are the coefficients of the hyperplane, ϵ_{d_j} is the classification error, ρ and ν are user-defined parameters.

$$\min \frac{1}{2} \|\mathbf{h}\|^2 + \frac{1}{\nu \cdot |\mathcal{D}|} \sum_{\mathbf{d}_j \in \mathcal{D}} \epsilon_{d_j} - \rho, \tag{4}$$

The score assigned to a new document d_i is presented in Eq. 5, in which $\Phi(\mathbf{d}_i)$ maps the original space to a new space in which examples of distinct classes are linearly separable, and $\text{sgn}(\cdot)$ returns 1 if the function value is ≥ 0 and 0 otherwise.

$$f(\mathbf{d}_i) = \text{sgn}(\mathbf{h} \cdot \Phi(\mathbf{d}_i) - \rho) \tag{5}$$

The use of an autoencoder for one-class text classification is proposed in Manevitz and Yousef (2007). The autoencoder is a neural network in which the goal is to produce an output similar to the input, i.e., the goal is to minimize the regularization function:

$$J(\Theta) = \frac{1}{n} \sum_{\mathbf{d}_i \in \mathcal{D}} \|\mathbf{w}_{\mathbf{d}_i} - \mathbf{y}_{\mathbf{d}_i}\|_2^2, \tag{6}$$

in which $\mathbf{y}_{\mathbf{d}_i}$ is the output of the neural network given a document \mathbf{d}_i , and Θ is the neural network parameters. Thus, given a new example, if the similarity between input and output

exceeds a threshold, the document is classified as the interest class. The proposed approach presented in Manevitz and Yousef (2007) is a feed-forward neural network composed of m inputs, h neurons in the hidden layer, and m outputs. This neural is also called “Bottleneck” or Dense Autoencoder (DAE), all layers are dense, and the activation functions of the neurons are all sigmoids.

For PUL algorithms, let the training documents set $\mathcal{D} = \{\mathbf{d}_1, \dots, \mathbf{d}_l, \mathbf{d}_{l+1}, \dots, \mathbf{d}_{l+u}\}$ be a news set, and $\mathcal{C} = \{\textit{interest}, \textit{non-interest}\}$ be a set of class labels. The first l elements of \mathcal{D} are labeled news from the interest class, composing the interest labeled set \mathcal{D}^+ . The remaining u elements are unlabeled news, composing the set \mathcal{D}^U , and usually $|\mathcal{D}^U| \gg |\mathcal{D}^+|$.

RC-SVM treats the unlabeled documents set \mathcal{D}^U as belonging to the non-interest class (Li & Liu, 2003). The interest set \mathcal{D}^+ and the unlabeled documents \mathcal{D}^U are used as the training data to build a Rocchio classifier, which is used to classify \mathcal{D}^U . The documents that are classified as negative are considered non-interest data, composing the set RN . Rocchio’s steps are described in the Algorithm 1.

A classifier is built using interest and not interest prototype vectors \mathbf{c}^+ and \mathbf{c}^- . The parameters α and β adjust the relative impact of the interest and non-interest data training examples (Li & Liu, 2003), and $\textit{sim}(\cdot)$ is the similarity between the document and the prototype.

Algorithm 1 Rocchio Support Vector Machine

Input:
 \mathcal{D}^+ , a set of labeled interest documents
 \mathcal{D}^U , a set of unlabeled documents

Output:
 RN , a set of extracted reliable non-interest documents

- 1: Assign the unlabeled set \mathcal{D}^U the non-interest class, and the interest set \mathcal{D}^+ the interest class
- 2: Let $\mathbf{c}^+ = \alpha \frac{1}{|\mathcal{D}^+|} \sum_{\mathbf{d}_i \in \mathcal{D}^+} \frac{\mathbf{d}_i}{\|\mathbf{d}_i\|} - \beta \frac{1}{|\mathcal{D}^U|} \sum_{\mathbf{d}_i \in \mathcal{D}^U} \frac{\mathbf{d}_i}{\|\mathbf{d}_i\|}$
- 3: Let $\mathbf{c}^- = \alpha \frac{1}{|\mathcal{D}^U|} \sum_{\mathbf{d}_i \in \mathcal{D}^U} \frac{\mathbf{d}_i}{\|\mathbf{d}_i\|} - \beta \frac{1}{|\mathcal{D}^+|} \sum_{\mathbf{d}_i \in \mathcal{D}^+} \frac{\mathbf{d}_i}{\|\mathbf{d}_i\|}$
- 4: **for all** document \mathbf{d}_i in \mathcal{D}^U **do**
- 5: **if** $\textit{sim}(\mathbf{c}^+, \mathbf{d}_i) \leq \textit{sim}(\mathbf{c}^-, \mathbf{d}_i)$ **then**
- 6: $RN \leftarrow RN \cup \{\mathbf{d}_i\}$
- 7: **end if**
- 8: **end for**
- 9: **return** RN

Step two of the RC-SVM algorithm consists of building a final classifier using Support Vector Machine iteratively with the \mathcal{D}^+ and RN document sets, described in Algorithm 2. A SVM classifier is used at each iteration to extract reliable non-interest documents from Q , $Q = \mathcal{D}^U - RN$. The iterations end when there are no more non-interest documents to be extracted. If many interest documents are included in the RN set, the last classifier will be extremely poor. In this way, the first classifier S_1 will be chosen (Li & Liu, 2003).

Algorithm 2 Classifier building

Input:
 \mathcal{D}^+ , a set of labeled interest documents
 \mathcal{D}^U , a set of unlabeled documents
 RN , a set of extracted reliable non-interest documents

- 1: Every document in \mathcal{D}^+ is assigned the class label *interest*;
- 2: Every document in RN is assigned the label *not interest*;
- 3: $Q = \mathcal{D}^U - RN$;
- 4: **for** $i \leftarrow 1$ initially and $i \leftarrow i + 1$ **do**
- 5: Use \mathcal{D}^+ and RN to train a SVM classifier S_i ;
- 6: Classify Q using S_i . Let the documents in Q that are classified as non-interest be W ;
- 7: **if** $W = \{\}$ **then** stop;
- 8: **else** $Q \leftarrow Q - W$
- 9: $RN \leftarrow RN \cup W$
- 10: **end if**
- 11: **end for**
- 12: Use the last SVM classifier S_{last} to classify \mathcal{D}^+ ;
- 13: **if** more than 5% interest documents are classified as negative **then**
- 14: use S_1 as the final classifier;
- 15: **else** use S_{last} as the final classifier;
- 16: **end if**

As presented in this section, the main OCL or PUL algorithms are based on the vector space model. So, the benefits of using networks in semi-supervised learning scenarios and modeling different types of relations are disregarded. In the next, we present our adaptation of PU-LP to perform a network-based positive and unlabeled learning.

4 Proposed approach: positive and unlabeled learning by label propagation for fake news detection

In order to make use of the benefits of network representations to model different types of relations in a text collection and for the learning in a semi-supervised scenario, we proposed a network-based approach for positive and unlabeled learning to detect fake news. Our proposal is based on PU-LP algorithm, applied to textual data. In addition to reducing labeling efforts, PU-LP extracts relevant information from unlabeled documents that assists in classifying news of interest (fake) and uninteresting (true) news. Then, a label propagation algorithm classifies the remaining unlabeled documents. As PU-LP is based on similarity networks, in which nodes in our approach are documents, we propose the inclusion of a new type of object in the network: representative terms (unigrams and bigrams) of the document collection. Unigrams and bigrams are widely used to discriminate true and false content (see Sect. 2.1). Also, works in the literature show that the combination of patterns obtained through the inclusion of documents and representative terms in heterogeneous networks increase the label propagation algorithms performance (Aggarwal & Li, 2011; Chakravarthy et al., 2014). Figure 1 presents the proposed approach of PU-LP for semi-supervised fake news classification. Each stage of Fig. 1 is described below.

4.1 Data structuring

In Fig. 1, Stage 1, the set of labeled news items is composed of fake news, while the other news items are not labeled. The labeled news set is much smaller than the unlabeled set. Initially, PU-LP builds an adjacency matrix with the full set of examples \mathcal{D} . According to

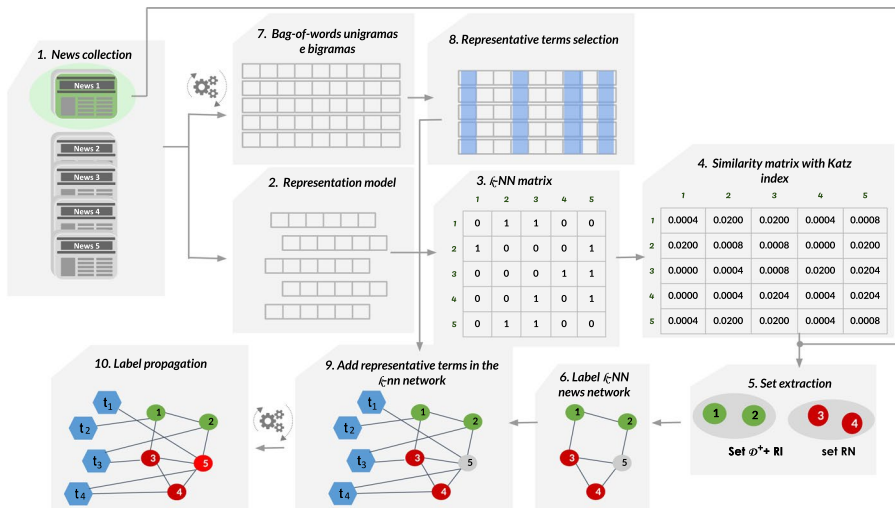


Fig. 1 Proposed approach for detecting false news based on the semi-supervised PU-LP algorithm

Fig. 1—Stage 2, the news collection must be preprocessed, and a representation model, such as Bag-of-Words (Salton, 1989) or document embeddings (Le & Mikolov, 2014), must be adopted to transform news into structured data.

4.2 Graph construction

An adjacency matrix is calculated using the representation model and a distance metric. In the adjacency matrix, news with similar content has a low distance between them. So, in Fig. 1—Stage 3, the adjacency matrix is used to build a matrix k -Nearest Neighbors (k -NN), called A . In matrix A , if the news \mathbf{d}_j is one of the k nearest neighbors to the news \mathbf{d}_i , then $A_{i,j} = 1$. Otherwise, $A_{i,j} = 0$. A network is also created through the k -NN algorithm, in which the nodes are news, and edges connect similar nodes with weight $w_{\mathbf{d}_i, \mathbf{d}_j} = 1$.

Some assumptions can be presented from the k -NN network-based representation (Katz 1953; Lü et al., 2009; Ma & Zhang, 2017): each edge is independent and has the same weight in the network as the other edges. In addition, if two news items are directly connected, they will generally be considered to belong to the same class. Likewise, labels can propagate along paths in the graph so that if two nodes have many neighbors in common, they are likely to be of the same class. Therefore, Ma and Zhang (2017) proposes the use of the Katz index, as in Fig. 1—Stage 4. Katz is a global similarity measure that calculates the similarity between pairs of nodes according to all possible paths that connect them in the network. Thus:

$$\text{sim}(\mathbf{d}_i, \mathbf{d}_j) = \sum_{h=1}^{\infty} \alpha^h \cdot |\text{path}_{\mathbf{d}_i, \mathbf{d}_j}^{<h>}| = \alpha A_{i,j} + \alpha^2 (A^2)_{i,j} + \alpha^3 (A^3)_{i,j} + \dots \quad (7)$$

In Eq. 7, α is a free parameter that controls the influence of paths in the network, thus, longer paths contribute less than short ones. When $\alpha < 1/\epsilon$, in which ϵ is the biggest eigenvalue for the matrix A , Eq. 7 converges and can be calculated according to the Eq. 8, in

which $W = (|D| \times |D|)$ and I denotes the identity matrix. Thus, $W_{ij} \in \mathbb{R}$ denotes the similarity between the news \mathbf{d}_i and \mathbf{d}_j considering the Katz index.

$$W = (I - \alpha A)^{-1} - I, \tag{8}$$

In Fig. 1—Stage 5, the similarity matrix W and the labeled news in \mathcal{D}^+ are used to infer the set RI of reliable interest news and the set RN of reliable non-interest news. While RI contains news from the set \mathcal{D}^U that are most similar to examples from \mathcal{D}^+ , RN has news from $\mathcal{D}^U - RI$ that are most dissimilar to the set $\mathcal{D}^+ \cup RI$.

The inference of the set RI is divided into m iterations. In each of the m iterations, $(\lambda/m) \times |\mathcal{D}^+|$ reliable interest news will be extracted, in which λ controls the size of the set. So, based on W , news in \mathcal{D}^U are ranked according to their average similarities for all news in \mathcal{D}^+ . The $(\lambda/m)|\mathcal{D}^+|$ most similar news are taken from \mathcal{D}^U , forming the set RI' . After each iteration ends, RI is incremented with the elements of RI' . Thus, $RI \leftarrow RI \cup RI'$ (Ma & Zhang, 2017).

For the inference of the reliable non-interest set, news in $\mathcal{D}^U - RI$ are ranked according to their average similarities for all news in $\mathcal{D}^+ \cup RI$, based on W matrix. PU-LP extracts the $|\mathcal{D}^+ \cup RI|$ most dissimilar news, forming the set RN . After getting the set RN , the sets $\mathcal{D}^+ \cup RI$, RN and $\mathcal{D}^U \leftarrow (\mathcal{D}^U - RI - RN)$ are used as input by label propagation algorithms based on transductive semi-supervised learning. In the traditional PU-LP algorithm (Algorithm 3) which considers only the news network, we use homogeneous networks such as those described in Sect. 4.3.1 (Fig. 1—Stage 6). From the labeled vertices, which are fixed, the algorithm gradually labels the set of non-labeled vertices (Ma & Zhang, 2017).

Algorithm 3 *Positive and Unlabeled Learning by Label Propagation*

Input:
 \mathcal{D}^+ , a set of labeled interest documents
 \mathcal{D}^U , a set of unlabeled documents
 m , number of iterations
 λ , controls the size of set RI , $\lambda \in (0, 1)$
 W , similarity matrix calculated by Katz index

Output:
 RI , a set of extracted reliable interest documents
 RN , a set of extracted reliable non-interest documents

- 1: $RI \leftarrow \emptyset, RN \leftarrow \emptyset$
- 2: **for** $k \leftarrow 1 : m$ **do**
- 3: based on W , calculate $S_{\mathbf{d}_i}, \mathbf{d}_i \in \mathcal{D}^U, S_{\mathbf{d}_i} = \frac{\sum_{j=1}^{|\mathcal{D}^+|} W_{i,j}}{|\mathcal{D}^+|}, S_{\mathbf{d}_i}$ is the mean similarity between an unlabeled document \mathbf{d}_i and all labeled interest documents;
- 4: rank each document \mathbf{d}_i according to the $S_{\mathbf{d}_i}, \mathbf{d}_i \in \mathcal{D}^U$;
- 5: $RI' \leftarrow$ the top $\frac{\lambda}{m} \times |\mathcal{D}^+|$ ranked examples in \mathcal{D}^U ;
- 6: $RI \leftarrow RI \cup RI', \mathcal{D}^U \leftarrow \mathcal{D}^U - RI'$
- 7: **end for**
- 8: based on W , calculate $S_{\mathbf{d}_i}, \mathbf{d}_i \in \mathcal{D}^U - RI, S_{\mathbf{d}_i} = \frac{\sum_{j=1}^{|\mathcal{D}^+ \cup RI|} W_{i,j}}{|\mathcal{D}^+ \cup RI|}$;
- 9: rank each document \mathbf{d}_i according to the $S_{\mathbf{d}_i}, \mathbf{d}_i \in \mathcal{D}^U - RI$;
- 10: $RN \leftarrow$ the bottom $|\mathcal{D}^+ \cup RI|$ ranked examples in $\mathcal{D}^U - RI$;
- 11: **return** RI, RN

In order to improve the classification performance of PU-LP, we also add representative terms to the news network that assist in the label propagation after the inference of the RI and RN sets. Relations among documents (news) and terms are generic since terms naturally compose the news in any type of news collection. We selected representative unigrams and bigrams from the Bag-of-Words representation model, considering tf-idf as term-weighting scheme (Feldman et al., 2007; Manning et al., 2008; Yan et al., 2020),

which weighs the frequency of the term by the inverse of the number of documents in which the term occurs.

For the selection of relevant terms, we remove stopwords and apply stemming to the remaining terms. Bag-of-Words is created by the library `Feature_extraction` from Sklearn (Pedregosa et al., 2011), considering the following parameters: `n-gram_range`, which determines whether the matrix will be created considering only unigrams (1,1) or unigrams and bigrams (1,2); and `min_df`, which ignores terms that have a document frequency strictly below the informed limit. With the Bag-of-Words calculated, we select terms with a tf-idf value above a threshold ℓ to be included in the k -NN news network (see Sect. 4.2) as new unlabeled nodes (Fig. 1—Stage 9). It is important to limit the vocabulary to more discriminating words so that the size of the network and the computational complexity are reduced (Aggarwal & Li, 2011; Chakravarthy et al., 2014; Deepak et al., 2021; Yan et al., 2020). The corresponding tf-idf value gives the weight of the news-term edge in the Bag-of-Words. After building the news and terms network, the next stage (Fig. 1—Stage 10) is to carry out the label propagation using transductive algorithms for heterogeneous networks, such as those described in Sect. 4.3.2.

4.3 Label propagation

We propose the use of regularization-based transductive algorithms for fake news classification, which seek to minimize a cost function that satisfies two premises: (i) the class information of neighboring documents must be similar; and (ii) the class information of the labeled documents assigned during the classification process must be similar to the real class information (Rossi, 2016). Such premises can be expressed in a regularization framework, in which the first term is the regularization function and the second term is the cost function (Delalleau et al., 2005):

$$Q(\mathbf{F}) = \frac{1}{2} \sum_{\mathbf{d}_i, \mathbf{d}_j \in \mathcal{D}} w_{\mathbf{d}_i, \mathbf{d}_j} \Omega(\mathbf{f}_{\mathbf{d}_i}, \mathbf{f}_{\mathbf{d}_j}) + \mu \sum_{\mathbf{d}_i \in \mathcal{D}^L} \Omega'(\mathbf{f}_{\mathbf{d}_i}, \mathbf{y}_{\mathbf{d}_i}). \quad (9)$$

In Eq. 9, the vector $\mathbf{f}_{\mathbf{d}_i} = \{f_{interest}, f_{not-interest}\}$, that is $\mathbf{f}_{\mathbf{d}_i} = \{f_{fake}, f_{real}\}$, gives how much a document \mathbf{d}_i belongs to each of the classes in \mathcal{C} (class information). The vector $\mathbf{y}_{\mathbf{d}_i}$ has the same dimensions as the vector \mathbf{f} , but gives the original label of the document \mathbf{d}_i , in which the position of the vector corresponding to its label is filled with 1. The term $w_{\mathbf{d}_i, \mathbf{d}_j}$ denotes the weight of the edge connecting document \mathbf{d}_i to document \mathbf{d}_j on the network. The term \mathcal{D}^L refers to the set of labeled documents. Furthermore, μ is the regularization parameter that defines the importance given to each of the premises; $\Omega(\dots)$ and $\Omega'(\dots)$ are distance functions.

The function $\Omega(\dots)$ calculates the proximity between the class information vectors and each pair of related documents on the network. Next, the function $\Omega'(\dots)$ calculates the proximity between the class information of labeled documents and their real class information. The equation can be solved using iterative solutions, called “label propagation”, in which documents gradually spread their class information to neighboring documents. The propagation is done until there are no more changes in the network nodes’ class information or by defining a maximum number of iterations for the algorithm.

We use iterative versions of Gaussian Fields and Harmonic Functions (GFHF) (Zhu et al., 2003) and Learning With Local and Global Consistency (LLGC) (Zhou et al., 2004a) as label propagation algorithms in PU-LP applied to homogenous networks (news

network). Both GFHF and LLGC are well established in the literature (Hua et al., 2021). For the proposed approach, which considers news and terms relations, we use Label Propagation through Heterogeneous Networks (LPHN) (Rossi 2016) and GNetMine (GNM) (Ji et al., 2010) algorithms, which are extensions of the GFHF and LLGC, respectively.

4.3.1 Label propagation in homogeneous networks

The GFHF algorithm uses a harmonic function that determines a document class information based on the class information average of neighboring documents. It is important to note that the GFHF algorithm was also used to propagate labels in the work developed by Ma and Zhang (2017). Equation 10 gives the class information of a document \mathbf{d}_i .

$$\mathbf{f}_{\mathbf{d}_i} = \frac{\sum_{\mathbf{d}_j \in \mathcal{D}} w_{\mathbf{d}_i, \mathbf{d}_j} \mathbf{f}_{\mathbf{d}_j}}{\sum_{\mathbf{d}_j \in \mathcal{D}} w_{\mathbf{d}_i, \mathbf{d}_j}}. \tag{10}$$

The GFHF algorithm does not allow class information for documents previously labeled in the network to be changed during the propagation phase. This premise is guaranteed by $\lim_{\mu \rightarrow \infty}$ in the regularization function to be minimized, presented in Eq. 11.

$$Q(\mathbf{F}) = \frac{1}{2} \sum_{\mathbf{d}_i, \mathbf{d}_j \in \mathcal{D}} w_{\mathbf{d}_i, \mathbf{d}_j} \|\mathbf{f}_{\mathbf{d}_i} - \mathbf{f}_{\mathbf{d}_j}\|^2 + \lim_{\mu \rightarrow \infty} \sum_{\mathbf{d}_i \in \mathcal{D}^L} \|\mathbf{f}_{\mathbf{d}_i} - \mathbf{y}_{\mathbf{d}_i}\|^2. \tag{11}$$

Class Mass Normalization (CMN), presented in Eq. 12, is used to classify the documents (Zhu et al., 2003) considering the final values of $\mathbf{f}_{\mathbf{d}_i}$ vectors for $\mathbf{d}_i \in \mathcal{D}^U$. The label propagation to minimize Eq. 11 is presented in Algorithm 4.

$$class(\mathbf{d}_i) = \arg \max_{1 \leq l \leq |C|} Pr[c_l] \cdot \frac{f_{\mathbf{d}_i, c_l}}{\sum_{\mathbf{d}_j \in \mathcal{D}} f_{\mathbf{d}_j, c_l}} \tag{12}$$

Algorithm 4 *Gaussian Fields and Harmonic Functions*

Input:
 \mathcal{D} , a set of documents
 \mathbf{Y} , real class information
 \mathbf{W} , connection weights between documents

Output:
 $\mathbf{F}(\mathcal{D}^U)$, class information for unlabeled documents

- 1: $\mathbf{D} = \text{diag}(\mathbf{W} \cdot \mathbf{I}_{|\mathcal{D}|})$; $\text{diag}(\dots)$ is the matrix diagonal operator
- 2: $\mathbf{P} \leftarrow (1/\mathbf{D}) \cdot \mathbf{W}$
- 3: **while** there is no convergence or maximum number of iterations not being reached **do**
- 4: $\mathbf{F}(\mathcal{D}) \leftarrow \mathbf{P} \cdot \mathbf{F}(\mathcal{D})$;
- 5: $\mathbf{F}(\mathcal{D}^L) \leftarrow \mathbf{Y}(\mathcal{D}^L)$;
- 6: **end while**
- 7: **return** $\mathbf{F}(\mathcal{D}^U)$

LLGC (Zhou et al., 2004a) decreases the influence of objects with a high degree in the definition of neighboring objects’ class information. The algorithm allows the class information of neighboring documents to change during the classification process since objects can be wrongly labeled, deteriorating the classification performance. In the label propagation

process, to calculate the class information of an object, the algorithm considers both information related to the source document's degree and the degree of the destination document. This characteristic is expressed in the normalization of the first term in the regularization function, in Eq. 13:

$$Q(\mathbf{F}) = \frac{1}{2} \sum_{\mathbf{d}_i, \mathbf{d}_j \in \mathcal{D}} w_{\mathbf{d}_i, \mathbf{d}_j} \left\| \frac{\mathbf{f}_{\mathbf{d}_i}}{\sqrt{\sum_{\mathbf{d}_k \in \mathcal{D}} w_{\mathbf{d}_i, \mathbf{d}_k}}} - \frac{\mathbf{f}_{\mathbf{d}_j}}{\sqrt{\sum_{\mathbf{d}_k \in \mathcal{D}} w_{\mathbf{d}_j, \mathbf{d}_k}}} \right\|^2 + \mu \sum_{\mathbf{d}_i \in \mathcal{D}^L} \|\mathbf{f}_{\mathbf{d}_i} - \mathbf{y}_{\mathbf{d}_i}\|^2 \quad (13)$$

The documents are classified considering the arg-max of the final value of $\mathbf{f}_{\mathbf{d}_i}$ vectors for $\mathbf{d}_i \in \mathcal{D}^U$. The label propagation solution to minimize Eq. 13 is presented in Algorithm 5.

Algorithm 5 Learning with Local and Global Consistency

Input:
 \mathcal{D} , a set of documents
 \mathbf{Y} , real class information
 \mathbf{W} , connection weights between documents
 α , LLGC's parameter to attenuate differences of the class information of labeled documents in consecutive iterations
Output:
 $\mathbf{F}(\mathcal{D}^U)$, class information for unlabeled documents

- 1: $\mathbf{D} = \text{diag}(\mathbf{W} \cdot \mathbf{I}_{|\mathcal{D}|})$; $\text{diag}(\dots)$ is the matrix diagonal operator
- 2: $\mathbf{S} = \mathbf{D}^{-1/2} \cdot \mathbf{W} \cdot \mathbf{D}^{-1/2}$
- 3: **while** there is no convergence or maximum number of iterations not being reached **do**
- 4: $\mathbf{F}(\mathcal{D}) \leftarrow \alpha \cdot \mathbf{S} \cdot \mathbf{F}(\mathcal{D}) + (1 - \alpha) \cdot \mathbf{Y}(\mathcal{D})$
- 5: **end while**
- 6: **return** $\mathbf{F}(\mathcal{O}^U)$

4.3.2 Label propagation in heterogeneous networks

The LPHN algorithm extends the GFHF algorithm (see Sect. 4.3.1) for transductive classification in heterogeneous networks. The LHPN regularization function is analogous to the GFHF function (Eq. 11). However, it considers different types of relations in the first term of the regularizer. Consider, for example, a network containing two objects, in which \mathcal{O}_1 are documents and \mathcal{O}_2 are terms of a news collection. According to Eq. 14, the algorithm propagates labels from a set of objects \mathcal{O}_i to a set of objects \mathcal{O}_j .

$$Q(\mathbf{F}) = \sum_{\mathcal{O}_i, \mathcal{O}_j \in \mathcal{O}} \frac{1}{2} \sum_{o_i \in \mathcal{O}_i} \sum_{o_j \in \mathcal{O}_j} w_{o_i, o_j} (\mathbf{f}_{o_i} - \mathbf{f}_{o_j})^2 + \lim_{\mu \rightarrow \infty} \sum_{o_i \in \mathcal{O}^L} (\mathbf{f}_{o_i} - \mathbf{y}_{o_i})^2. \quad (14)$$

Equation 14 can be solved using iterative methods, such as label propagation, as presented in Algorithm 6.

Algorithm 6 *Label Propagation through Heterogeneous Networks*

Input:
 \mathcal{O} , set of network objects
 \mathbf{Y} , real class information
 \mathbf{W} , connection weights between objects
 \mathbf{D} , degree of objects

Output:
 $\mathbf{F}(\mathcal{O}^U)$, class information for unlabeled objects

```

1:  $\mathbf{P} \leftarrow (\mathbf{D}^{-1}) \cdot \mathbf{W}$ ;
2: while there is no convergence or maximum number of iterations not being reached do
3:   for all  $\mathcal{O}_i, \mathcal{O}_j \subset \mathcal{O}$  (subset of type  $i$  and  $j$  objects) do
4:      $\mathbf{F}(\mathcal{O}_i, \mathcal{O}_j) \leftarrow \mathbf{P}(\mathcal{O}_i, \mathcal{O}_j) \cdot \mathbf{F}(\mathcal{O}_i, \mathcal{O}_j)$ ;
5:      $\mathbf{F}(\mathcal{O}_i^L) \leftarrow \mathbf{Y}(\mathcal{O}_i^L)$ ;
6:      $\mathbf{F}(\mathcal{O}_j^L) \leftarrow \mathbf{Y}(\mathcal{O}_j^L)$ ;
7:   end for
8: end while
9: return  $\mathbf{F}(\mathcal{O}^U)$ 

```

The GNetMine (GNM) Ji et al. (2010) algorithm is an extension of the LLGC algorithm (See Sect. 4.3.1). In addition to the relations between different objects having different importance, the algorithm allows the reliability of the training object labels to be reduced. In this way, the label of an object can be changed during the classification process if information from neighboring objects diverges from the class of the object initially labeled. Equation 15 describes the regularization function of GNM. The term λ defines the object’s importance initially of the relationship between objects of the type \mathcal{O}_i and \mathcal{O}_j , which varies between 0 and 1. The term $\alpha_{o_j} \in \mathcal{O}^L$ defines the importance of the object initially labeled o_j , also ranging between 0 and 1.

$$\begin{aligned}
 Q(\mathbf{F}) = & \sum_{\mathcal{O}_i, \mathcal{O}_j \in \mathcal{O}} \lambda_{\mathcal{O}_i, \mathcal{O}_j} \sum_{o_k \in \mathcal{O}_i} \sum_{o_l \in \mathcal{O}_j} w_{o_k, o_l} \left\| \frac{\mathbf{f}_{o_k}(\mathcal{O}_i)}{\sqrt{\sum_{o_m \in \mathcal{O}_i} w_{o_k, o_m}}} - \frac{\mathbf{f}_{o_l}(\mathcal{O}_j)}{\sqrt{\sum_{o_m \in \mathcal{O}_j} w_{o_l, o_m}}} \right\|^2 \\
 & + \sum_{o_j \in \mathcal{O}^L} \alpha_{o_j} (\mathbf{f}_{o_j} - \mathbf{y}_{o_j})
 \end{aligned} \tag{15}$$

in which $\lambda_{\mathcal{O}_i, \mathcal{O}_j} (0 \geq \lambda_{\mathcal{O}_i, \mathcal{O}_j} \geq 1)$ is the importance given to the relationship between objects of the types \mathcal{O}_i and \mathcal{O}_j , and $\alpha_{o_i} (0 \geq \alpha_{o_i} \geq 1)$ is the importance given to the real class information of an object $o_j \in \mathcal{O}^L$ (set of labeled objects).

Equation 15 can be solved using iterative methods as presented in the Algorithm 7. In line 2, a confidence value of the connection between two types of objects is defined. In lines 7 to 9, the object class information is updated based on information from neighboring objects, as well as based on the importance of each relationship and the reliability of the labeled information.

Algorithm 7 *GNetMine*

Input:
 \mathcal{O} , set of network objects
 \mathbf{Y} , real class information
 \mathbf{W} , connection weights between objects
 \mathbf{D} , degree of objects
 α , reliability of labeled objects
 $\lambda[\cdot]$, importance of relationships between different objects in the network

Output:
 $\mathbf{F}(\mathcal{O}^U)$, class information for unlabeled objects

- 1: **for all** $\mathcal{O}_i, \mathcal{O}_j \subset \mathcal{O}$ (subset of type i and j objects) **do**
- 2: $\mathbf{S}(\mathcal{O}_i, \mathcal{O}_j) \leftarrow \mathbf{D}(\mathcal{O}_i, \mathcal{O}_j)^{-\frac{1}{2}} \cdot \mathbf{W}(\mathcal{O}_i, \mathcal{O}_j) \cdot \mathbf{D}(\mathcal{O}_i, \mathcal{O}_j)^{-\frac{1}{2}}$;
- 3: **end for**
- 4: **while** there is no convergence or maximum number of iterations not being reached **do**
- 5: **for all** $\mathcal{O}_i \subset \mathcal{O}$ **do**
- 6: **for all** $o_j \subset \mathcal{O}_i$ **do**
- 7: $\mathbf{f}_{o_j} \leftarrow \sum_{\mathcal{O}_k \subset \mathcal{O}, o_j \notin \mathcal{O}_k} \lambda_{\mathcal{O}_i, \mathcal{O}_k} \cdot \sum_{o_m \in \mathcal{O}_k} \mathbf{S}(\mathcal{O}_i, \mathcal{O}_k)_{o_j, o_m} \cdot \mathbf{f}_{o_m}$;
- 8: $\mathbf{f}_{o_j} \leftarrow \mathbf{f}_{o_j} + (2 \cdot \lambda_{\mathcal{O}_i, \mathcal{O}_i} \cdot \sum_{o_m \in \mathcal{O}_i} \mathbf{S}(\mathcal{O}_i, \mathcal{O}_i)_{o_j, o_m} \cdot \mathbf{f}_{o_m}) + \alpha_{o_j} \cdot \mathbf{y}_{o_j}$;
- 9: $\mathbf{f}_{o_j} \leftarrow \frac{\mathbf{f}_{o_j}}{\sum_{\mathcal{O}_k \subset \mathcal{O}, o_j \notin \mathcal{O}_k} \lambda_{\mathcal{O}_i, \mathcal{O}_k} + 2 \cdot \lambda_{\mathcal{O}_i, \mathcal{O}_i} + \alpha_{o_j}}$;
- 10: **end for**
- 11: **end for**
- 12: **end while**
- 13: **return** $\mathbf{F}(\mathcal{O}^U)$

Considering how vital is the classification effectiveness to a fake news detection algorithm, we measure the classification performance considering $F_1 = (2 \cdot \text{precision} \cdot \text{recall}) / (\text{precision} + \text{recall})$ only for fake news as interest set (interest- F_1), and an average of F_1 considering fake and true news as positive set (macro F_1).

5 Experimental evaluation

To assess the performance of our proposed approach for news classification based on PU-LP, we propose an experimental evaluation that considers: a comprehensive set of parameters; two models to generate structured representations (Bag-of-Words and Doc2Vec); and datasets with different scenarios, considering both languages (Portuguese and English) and class balancing. We also consider four traditional OCL algorithms and two PUL algorithms - the traditional PU-LP and RC-SVM.

5.1 News collections

This paper aims to evaluate the PUL and OCL algorithms considering balanced and unbalanced collections, news in Portuguese and in English, and collections containing only one subject or multiple subjects. The first collection was acquired from FakeNewsNet repository¹ (Shu et al., 2020), which contains news of famous people fact-checked by the *GossipCop*² website. The dataset has 16095 real news and 4937 fake news. However, after an initial analysis, we found that the news presented a significant imbalance in the number of tokens (words present in the news after removing unnecessary characters and stopwords),

¹ <https://github.com/KaiDMML/FakeNewsNet>.

² <https://www.gossipcop.com/>.

Table 2 Summary of the news collections used in the experimental evaluation

	FakeNewsNet	Fake.BR	Fact-checked News
Language	English	Portuguese	Portuguese
Fake news	1705	3600	1044
Real news	5298	3600	1124
Subjects	Celebrities	Science, society, politics, religion, celebrities and economy	Politics
Avg number of tokens for fake news	299.49	103.74	286.94
Avg number of tokens for real news	317.42	105.35	282.15

caused by crawling errors. Therefore, news ranging from 200 to 600 tokens were selected, remaining 5298 real news and 1705 fake news. FakeNewsNet is the collection with the greatest unbalance in the distribution of classes. The remaining collections present a balanced number of real and fake news.

The second collection, Fake.BR,³ is the first reference corpus in Portuguese for fake news detection. The news was manually collected and labeled. All of them have a textual format, available in their original sizes, and truncated. The truncation in the texts was carried out to have a text dataset with an approximate number of words, avoiding bias in the learning process. The corpus consists of 7200 news items, distributed in 6 categories: politics (58%), TV and celebrities (21.4%), society and daily life (17.7%), science and technology (1.5%), economy (0.7%) and religion (0.7%). This corpus contains 3600 fake news and 3600 true news (Silva et al., 2020).

The third news collection,⁴ also in Portuguese, was the result of a collection on fact-checking news - AosFatos,⁵ Agência Lupa,⁶ Fato ou Fake,⁷ UOL Confere⁸ and G1 - Política.⁹ The collection contains 2168 news, in which 1124 are real and 1044 are fake, and was collected during our project's execution. Some terms that were added after the checking process were removed since they are correlated with the classes: “fato”, “fake”, “verdadeiro”, “falso”, “#fake”, “verificamos”, “montagem”, “erro” e “checagem” (in English: fact, fake, real, check, and montage, respectively).

For each of the mentioned collections, the news was first preprocessed, characters were converted to lowercase, stopwords, links and numbers were removed. The remaining words were reduced to their word stems with PorterStemmer from Natural Language Toolkit—NLTK (Garrette & Klein, 2009). Such changes preserved the word order in each news item. A summary of the characteristics of the news collections after preprocessing is presented in Table 2.

The next stage was to convert each news into a feature vector. For this, we used the traditional approach known as Bag-of-Words, which uses simple words as terms in the

³ <https://github.com/roneysco/Fake.BR-Corpus>

⁴ We make this news collection available in the Github repository: <https://github.com/marianacaravanti/PU-LP-for-fake-news-detection>

⁵ <https://aosfatos.org/noticias/>

⁶ <https://piaui.folha.uol.com.br/lupa/>

⁷ <https://g1.globo.com/fato-ou-fake/>

⁸ <https://noticias.uol.com.br/confere/>

⁹ <https://g1.globo.com/politica/>

document collection, generating the document-term matrix, and Doc2Vec document embeddings (Le & Mikolov, 2014).¹⁰ Next section presents the configurations used for these representation models.

5.2 Experiment configuration and evaluation criteria

This section presents the experiment configuration and evaluation criteria for both the OCL and PUL approaches. To represent each of the datasets, after preprocessing, feature vectors were obtained considering two strategies:

- **Bag-of-Words** we considered unigrams as terms and considered tf-idf as term-weighting scheme.
- **Doc2Vec** (Paragraph Vectors) we used the union of the models Distributed Memory and Distributed Bag-of-Words to generate the document embeddings. For training each of these models, we consider the average and concatenation of the word vectors to create the hidden layer's output. Also, we employed the range of the maximum number of epochs $\in \{100, 1000\}$, $\alpha = 0.025$ and $\alpha_{min} = 0.0001$, number of dimensions of each model = 500, window size = 8, and minimum count = 1 (Le & Mikolov, 2014; Martinčić-Ipšić et al., 2019; Pita & Pappa, 2018). So, the four representation models are as follows:
 - Rep. 1: Method=average, Max epochs=100;
 - Rep. 2: Method=average, Max epochs=1000;
 - Rep. 3: Method=concatenation, Max epochs=100;
 - Rep. 4: Method=concatenation, Max epochs=1000.

The OCL algorithms and their parameters are the following:

- **k -NND** (Tan et al., 2019): $k = 1 + 3 * p, p \in [1..7]$ and cosine as similarity measure.
- **k -Means** (Tan et al., 2019): $k = 1 + 2 * p, p \in [1..9]$, 100 as the maximum number of iterations and cosine as the similarity measure. We performed 10 trials and chose the clustering result with the highest cohesion.
- **One-Class Support Vector Machines (OCSVM)** (Manevitz & Yousef, 2001): $\gamma = 1 * 10^p, p \in [-3..1]$, $\nu = 0.05 + 0.1 * q, q \in [0..9]$, and the *kernels* Linear and Radial Basis Function.
- **DAE**: a single hidden layer with $h \in \{2, 6, 12\}$ (number of neurons in the hidden layer) (Manevitz & Yousef, 2007), ADAM as optimizer with $\eta = 0.01$ (learning rate) (Kingma & Ba, 2015), the weights of the document vectors were normalized to 1 (vectors are divided by the norm) (Manevitz & Yousef, 2007), 200 as the maximum number of iterations, and cosine to measure the similarity between the input and the output. The activation function in the hidden layers is ReLU and in the output layer is a Sigmoid.

k -NND, k -Means, and DAE require a threshold to define if a new text belongs to the interest class. We considered thresholds manually and automatically defined: In case of manually defined, we have $threshold \in \{0.05 \times z, z \in \mathbb{N} : 1 \leq z \leq 19\}$. The 6σ approach was used to set thresholds automatically (Muir, 2005). In this case, the scores $f(\mathbf{d}_i)$ are

¹⁰ <https://radimrehurek.com/gensim/models/Doc2Vec.html>.

generated for the training documents, the average (μ) and standard deviation are computed (σ), and then $threshold \in \{\mu - 3\sigma, \mu - 2\sigma, \mu - 1\sigma, \mu, \mu + 1\sigma, \mu + 2\sigma, \mu + 3\sigma\}$.

For **RC-SVM** Li and Liu (2003), we used $\alpha = \{0.2, 0.4, 0.6, 0.8, 1.0\}$, and $\beta = 1 - \alpha$ for the Rocchio algorithm. We also used $C \in \{0.1, 1.0, 10\}$, and the linear kernel for the SVM algorithm.

For the original **PU-LP** (homogeneous network), in which k -NN network consists of news only (Ma & Zhang, 2017), the parameters used were:

- **k -NN matrix and network:** $k = [5, 7]$ and cosine as similarity measure, suitable for textual domains.
- **Reliable interest and non-interest sets extraction:** $m = \{1, 2\}$, $\lambda = \{0.6, 0.8\}$, $\alpha = \{0.005, 0.01, 0.02\}$. These values were chosen as suggested in Ma and Zhang (2017).
- **GFHF and LLGC label propagation algorithms:** as suggested in Santos (2018), we used a convergence threshold = 0.00005 and a maximum number of iterations = 1,000. For the LLGC we used $\alpha = \{0, 0.5\}$.

For PU-LP with representative unigrams and bigrams extracted from the Bag-of-Words representation model (heterogeneous network), we used the following parameters:

- **Bag-of-Words:** We consider two ways to add representative terms to the news network. The first considers only unigrams generated with BoW. We disregard terms that appear in fewer than two documents. In the second way we use unigrams and bigrams generated with BoW. To avoid the high dimensionality of the representation model, we disregard terms that appear in less than x documents, $x = \{3, 4\}$. For both ways, we use *tf-idf* as term-weighting scheme and we keep the terms in which the *tf-idf* in a document is greater than 0.08 (parameter ℓ). This value was chosen after a statistical analysis of the sample, indicating that about 25% of Bag-of-Words terms had *tf-idf* greater than 0.08.
- For k -NN matrix, network, and reliable interest and non-interest sets extraction, we used the same parameters as homogeneous PU-LP.
- **LPHN and GNM label propagation algorithms:** we used a convergence threshold = 0.00005 and a maximum number of iterations = 1000. For GNM we used $\alpha = \{0, 0.5\}$.

A 10-fold cross-validation adapted to OCL and PUL problems was used as a validation scheme. For each news collection, the set of fake news was randomly divided into 10 folds. In order to simulate a semi-supervised learning environment, in which the number of labeled examples is higher than the unlabeled examples, i.e., $|\mathcal{D}^+| \gg |\mathcal{D}^U|$, we carried out different experiments considering p folds to form the initially labeled set, $p = \{1, 2, 3\}$. The remaining folds and the real news are: (i) considered as test documents for the OCL algorithm; (ii) considered as unlabeled documents for the PUL algorithms.

We proposed a reference model using binary semi-supervised learning (BL) to assess the labeling procedure of reliable-interest and reliable-non-interest examples in PU-LP. For this analysis, the set of real news was randomly divided into ten subsets. In the cross-validation scheme, for each fake news fold used to train the algorithm, one fold of real news was used. That is, the algorithm works with p real news folds and p fake news folds in the initially labeled set, $p = \{1, 2, 3\}$. From the network obtained by the k -NN matrix (Fig. 1—Stage 3) and considering the training set as the set of labeled nodes, the label propagation algorithm infers the class of the remaining news through the network. We considered the values of k ranging in the interval $[5, 7]$. As label propagation algorithms, we used GFHF and LLGC with

the same parameters as homogeneous PU-LP. Next section presents the results of the experiments considering the proposed experimental configuration.

5.3 Results and discussion

In this section, we present the results achieved considering the experimental evaluation presented previously. Our goal is to demonstrate that OCL and PUL approaches can be relevant for detecting fake news, particularly because they learn classification models using a small set of labeled fake news, eliminating the need to label news from non-interesting classes. Furthermore, we want to show that our proposed approach, which uses PU-LP and heterogeneous networks, can achieve results as good as the binary reference model, and that the inclusion of terms in the news network improves the performance of label propagation algorithms, especially when the fake news is distributed in the feature space.

Tables 3 and 4 present the best results obtained by each algorithm, using the parameter set defined in Sect. 5.2, the Bag-of-Words and Doc2Vec representation models and the three news collections. The first three columns correspond to interest- F_1 , and the last three columns refer to the macro F_1 . 10%, 20%, and 30% indicate the percentage of fake news used to train the algorithms. The first two lines represent our approach, the heterogeneous PU-LP (HT) containing news and terms, using the label propagation algorithms LPHN and GNM. The third and fourth lines represent the homogeneous PU-LP (HM), using GFHF and LLGC algorithms. In the following lines, there are the OCL and PUL algorithms used to validate the approach. The last two rows refer to the results obtained by the binary semi-supervised reference model (BL), which has real and fake news labeled in the training set, considering the GFHF and LLGC label propagation algorithms. We also highlight in bold the best performances considering OCL and PUL approaches, the news collections, the initially labeled set, and F_1 measures.

With BoW (see Table 3) our PU-LP-LPHN approach presents better interest and macro F_1 , both for Fake.BR and for Fact-checked News collections. For FakeNewsNet, k -NND and PULP-GFHF show better interest- F_1 , while DAE stands out in macro F_1 . Considering D2V (see Table 4), our approach still stands out in the Fake.BR dataset. OCSVM also tends to increase its performance, especially considering 10% of labeled data and both the Fake.BR and FakeNewsNet datasets. Models based on clustering and density outperforms even the reference model using FakeNewsNet, achieving best interest- F_1 . For Fact-checked News, homogeneous PU-LP tends to obtain better performance.

We deepen our discussion of the results by answering the following research questions: (i) Did the Doc2Vec representation model provide better performance compared to traditional Bag-of-Words? (ii) Did PUL algorithms perform better than OCL algorithms? (iii) Did the proposed approach can surpass other PUL, OCL and the reference model for fake news detection? (iv) Did the amount of labeled fake news significantly increase the interest- F_1 ? (v) Which parameters of the proposed approach achieved the best results for interest- F_1 ? Next sections present the answers to those question.

5.3.1 Did the Doc2Vec representation model provide better performance compared to traditional Bag-of-Words?

To help answer this question, Figs. 2, 3, and 4 present bar plots that compare the interest- F_1 obtained for each dataset, considering the BoW (in orange) and D2V (in teal) models, for 10%, 20%, and 30% of labeled fake news.

Table 3 Results using Bag-of-Words as representation model, comparing the proposed approach, the traditional PUL and OCL algorithms and the binary semi-supervised reference model for detecting fake news

	Interest-F1			Macro F1		
	10%	20%	30%	10%	20%	30%
<i>FakeNewsNet</i>						
PU-LP-LPHN (HT)	0.5106	0.5135	0.4727	0.6690	0.6618	0.6213
PU-LP-GNM (HT)	0.4909	0.4862	0.4523	0.6428	0.6216	0.5860
PU-LP-GFHF (HM)	0.4709	0.5379	0.5556	0.6304	0.6411	0.6286
PU-LP-LLGC (HM)	0.4815	0.4850	0.4995	0.6410	0.6631	0.6580
RC-SVM	0.1436	0.4340	0.5123	0.2601	0.5316	0.5278
<i>k</i> -NND	0.5492	0.5431	0.5273	0.6547	0.6446	0.6336
<i>k</i> -Means	0.5379	0.5253	0.5106	0.6555	0.6449	0.6333
OCSVM	0.4364	0.4182	0.3932	0.5955	0.5868	0.5745
DAE	0.4968	0.5009	0.4740	0.6791	0.6803	0.6711
BL-GFHF	0.4967	0.5405	0.5581	0.6743	0.6954	0.7040
BL-LLGC	0.4245	0.4810	0.4995	0.6429	0.6716	0.6809
<i>Fake.BR</i>						
PU-LP-LPHN (HT)	0.6782	0.6935	0.6987	0.6700	0.6898	0.7219
PU-LP-GNM (HT)	0.6831	0.6904	0.6888	0.6658	0.6900	0.7161
PU-LP-GFHF (HM)	0.6405	0.6589	0.6555	0.6487	0.6686	0.6848
PU-LP-LLGC (HM)	0.6535	0.6659	0.6601	0.6515	0.6662	0.6903
RC-SVM	0.2530	0.5564	0.6865	0.3263	0.5914	0.7048
<i>k</i> -NND	0.6477	0.6272	0.6005	0.6433	0.6175	0.5899
<i>k</i> -Means	0.6444	0.6193	0.5908	0.6433	0.6175	0.5878
OCSVM	0.5602	0.5363	0.5078	0.5765	0.5505	0.5220
DAE	0.6435	0.6170	0.5848	0.5399	0.6648	0.6620
BL-GFHF	0.5921	0.6664	0.7023	0.6438	0.6884	0.7023
BL-LLGC	0.5001	0.6205	0.6682	0.5955	0.6585	0.6789
<i>Fact-checked news</i>						
PU-LP-LPHN (HT)	0.8957	0.8995	0.8973	0.9000	0.9062	0.9104
PU-LP-GNM (HT)	0.8699	0.8774	0.8751	0.8711	0.8845	0.8876
PU-LP-GFHF (HM)	0.8571	0.8546	0.8657	0.8613	0.8625	0.8839
PU-LP-LLGC (HM)	0.8581	0.8552	0.8672	0.8616	0.8631	0.8863
RC-SVM	0.0210	0.1604	0.7388	0.0239	0.1601	0.7508
<i>k</i> -NND	0.7185	0.7037	0.6772	0.7469	0.7394	0.7269
<i>k</i> -Means	0.7402	0.7250	0.7094	0.7538	0.7373	0.7384
OCSVM	0.7793	0.7832	0.7736	0.7108	0.7043	0.6886
DAE	0.7146	0.7215	0.7055	0.5399	0.6648	0.6620
BL-GFHF	0.8784	0.8983	0.9092	0.8774	0.8977	0.9088
BL-LLGC	0.8809	0.9004	0.9095	0.8790	0.8995	0.9086

HM denotes the use of homogeneous networks algorithms and HT denotes the use of heterogeneous networks algorithms. 10%, 20% e 30% indicate the percentage of fake news used in the labeled set. For our reference algorithm (BL), the labeled set also has the same percentage of real news. The best results considering PUL and OCL algorithms are highlighted in bold

Table 4 Results using Doc2Vec as representation model, comparing the proposed approach, the traditional PUL and OCL algorithms and the binary semi-supervised reference model for detecting fake news

	Interest-F1			Macro F1		
	10%	20%	30%	10%	20%	30%
<i>FakeNewsNet</i>						
PU-LP-LPHN (HT)	0.5345	0.5218	0.5064	0.6864	0.6713	0.6819
PU-LP-GNM (HT)	0.5200	0.4984	0.4828	0.6565	0.6354	0.6328
PU-LP-GFHF (HM)	0.5156	0.5057	0.4943	0.6675	0.6683	0.6615
PU-LP-LLGC (HM)	0.5189	0.5186	0.5249	0.6810	0.6967	0.7012
RC-SVM	0.7027	0.7518	0.6694	0.5217	0.6195	0.5427
<i>k</i> -NND	0.8575	0.8539	0.8499	0.6564	0.6471	0.6342
<i>k</i> -Means	0.8576	0.8537	0.8497	0.6558	0.6460	0.6397
OCSVM	0.7827	0.7560	0.8345	0.6930	0.5523	0.8121
DAE	0.3886	0.3641	0.3411	0.6213	0.6131	0.6053
BL-GFHF	0.4438	0.5202	0.5553	0.6514	0.6957	0.7159
BL-LLGC	0.2594	0.3932	0.4512	0.5633	0.6327	0.6629
<i>Fake.BR</i>						
PU-LP-LPHN (HT)	0.6572	0.7119	0.7152	0.6921	0.7476	0.7707
PU-LP-GNM (HT)	0.6654	0.6930	0.6815	0.6722	0.7071	0.7353
PU-LP-GFHF (HM)	0.6496	0.6749	0.6784	0.6570	0.6909	0.7187
PU-LP-LLGC (HM)	0.6610	0.6783	0.6815	0.6620	0.7019	0.7350
RC-SVM	0.6139	0.6398	0.6273	0.5539	0.6125	0.5863
<i>k</i> -NND	0.6497	0.6323	0.6066	0.6469	0.6259	0.5988
<i>k</i> -Means	0.6460	0.6195	0.5946	0.6454	0.6184	0.5851
OCSVM	0.8244	0.6128	0.6580	0.8172	0.6104	0.6765
DAE	0.6429	0.6156	0.5836	0.6687	0.6634	0.6606
BL-GFHF	0.6858	0.7341	0.7541	0.7177	0.7538	0.7700
BL-LLGC	0.7033	0.7488	0.7689	0.7308	0.7655	0.7818
<i>Fact-checked news</i>						
PU-LP-LPHN (HT)	0.9166	0.9196	0.9208	0.9185	0.9255	0.9315
PU-LP-GNM (HT)	0.9059	0.9140	0.9222	0.9097	0.9208	0.9326
PU-LP-GFHF (HM)	0.9200	0.9175	0.9268	0.9217	0.9223	0.9366
PU-LP-LLGC (HM)	0.9181	0.9161	0.9277	0.9195	0.9223	0.9379
RC-SVM	0.6117	0.5874	0.6416	0.4830	0.5080	0.6016
<i>k</i> -NND	0.6399	0.6162	0.5874	0.6472	0.6225	0.5973
<i>k</i> -Means	0.7116	0.7027	0.7197	0.6865	0.6905	0.6905
OCSVM	0.8490	0.7942	0.8144	0.8098	0.6979	0.7530
DAE	0.6863	0.6780	0.6671	0.6887	0.6842	0.6872
BL-GFHF	0.9306	0.9426	0.9477	0.9301	0.9424	0.9473
BL-LLGC	0.9329	0.9448	0.9501	0.9318	0.9442	0.9497

HM are homogeneous networks algorithms and HT are heterogeneous networks algorithms. 10%, 20% e 30% indicate the percentage of fake news used in the labeled set. For our reference algorithm (BL), the labeled set also has the same percentage of real news. The best results considering PUL and OCL algorithms are highlighted in bold

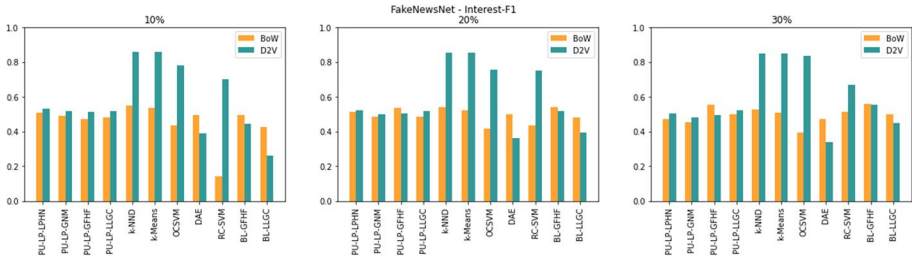


Fig. 2 Comparison of BoW (in orange) and D2V (in teal) representation models for FakeNewsNet, considering the interest- F_1 of the OCL, PUL, and the binary reference models, and 10%, 20% and 30% of labeled fake news (Color figure online)

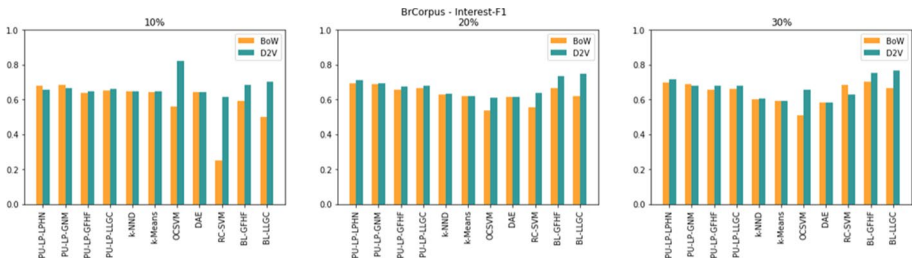


Fig. 3 Comparison of BoW (in orange) and D2V (in teal) representation models for Fake.BR, considering the interest- F_1 of the OCL, PUL and the binary reference model, and 10%, 20% and 30% of labeled fake news (Color figure online)

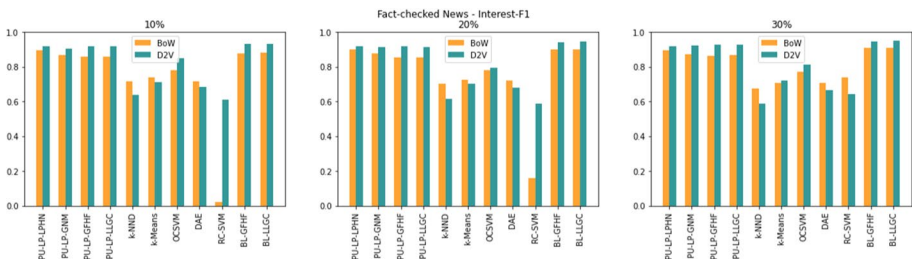


Fig. 4 Comparison of BoW (in orange) and D2V (in teal) representation models for Fact-checked News, considering the interest- F_1 of the OCL, PUL and the binary reference model, and 10%, 20% and 30% of labeled fake news (Color figure online)

We also used Sklearn’s t-SNE tool (Pedregosa et al., 2011) for insights into how the news are distributed in space considering each representation model. T-SNE converts similarities between data points into joint probabilities, trying to minimize the Kullback-Leibler divergence (Van der Maaten & Hinton, 2008) between the joint probabilities of the low dimension embedding and the high dimension data. Two-dimensional views are presented in Figs. 5, 6, 7, and 8. In the figures, we have the news distribution for the Bag-of-Words model, followed by the four representations of document embeddings with Doc2Vec. For FakeNewsNet and Fact-checked news datasets, -1 represents the real news (non-interest class) and 1 the fake news (interest class). For the Fake.

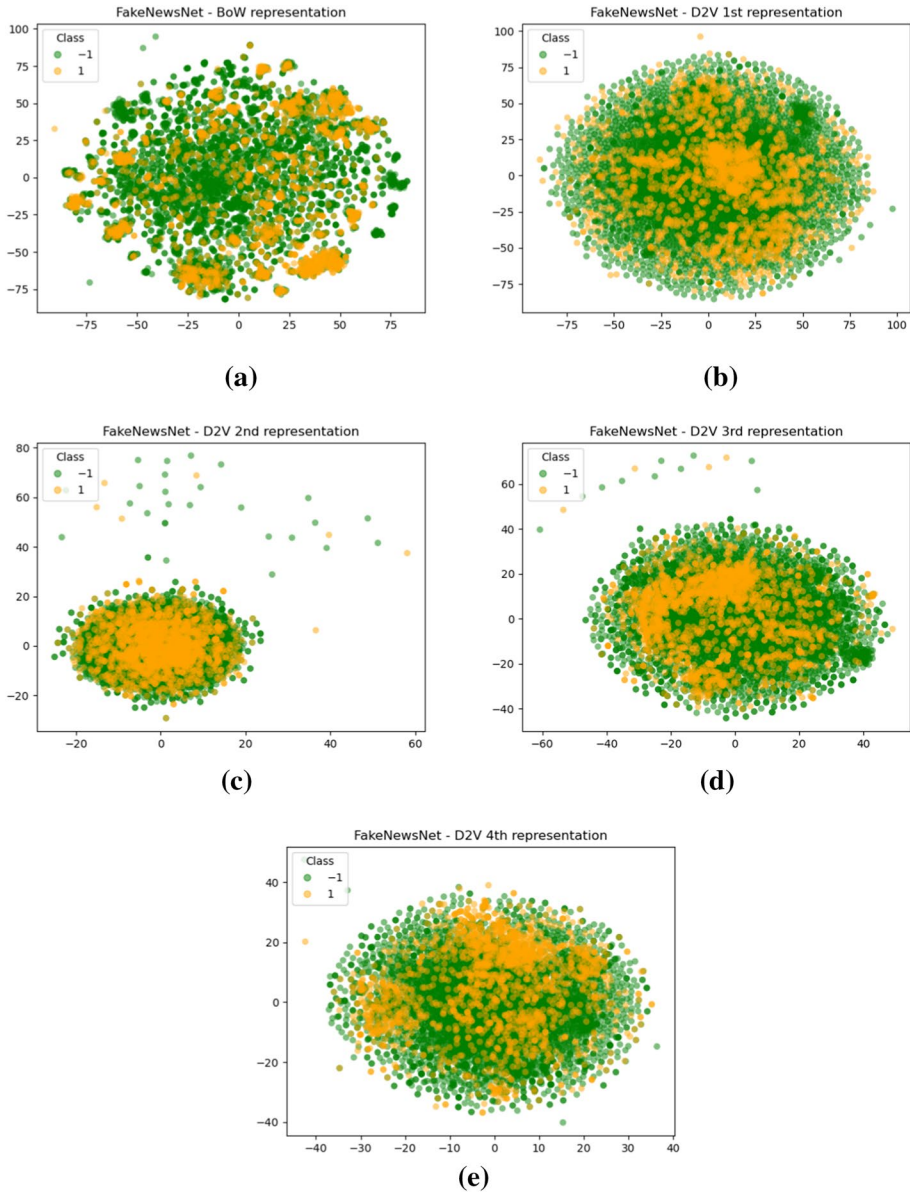


Fig. 5 News from the FakeNewsNet dataset plotted in two dimensions, using the t-SNE tool. **a** corresponds to the Bag-of-Words representation model, **b–e** correspond to the four representations of document embeddings generated with Doc2Vec. Green dots (– 1) are real news and orange dots (1) are fake news (Color figure online)

BR dataset, we plot real and fake news considering the subjects as well. In the figure caption, 1, 2, 3, 4, 5, and 6 correspond respectively to real news from politics, society, celebrities, science and technology, religion and economics. 7, 8, 9, 10, 11, and 12 correspond to fake news, also considering the same order.

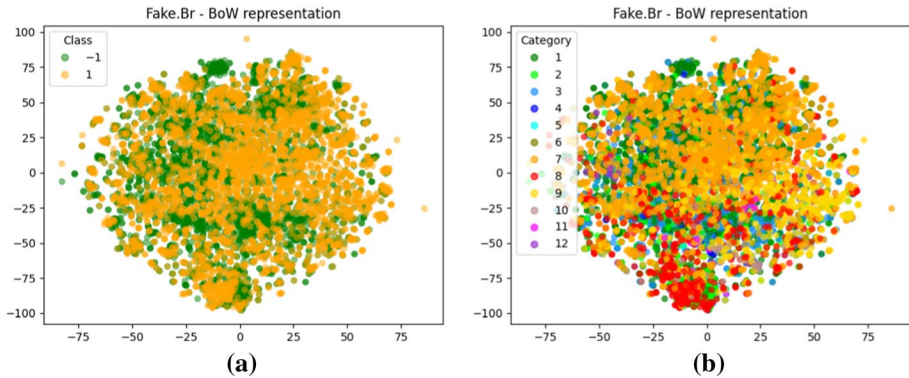


Fig. 6 News from the Fake. BR dataset plotted in two dimensions considering BoW representation, using the t-SNE tool. In **a**, Green dots (−1) are real news and orange dots (1) are fake news. In **b**, news related to the 6 categories are divided into two classes. In caption, 1, 2, 3, 4, 5, and 6 represent real news from politics, society, celebrities, science, religion and economics. 7, 8, 9, 10, 11 and 12 are fake news following the same order (Color figure online)

Overall, D2V representations provide better classification performances for algorithms. For example, the FakeNewsNet collection, which is unbalanced and has only celebrity news, Table 4 shows that the interest- F_1 increases drastically when D2V is used in the k -NND, k -Means, OCSVM and RC-SVM algorithms. Considering the Fig. 5, we can infer that D2V grouped fake news more efficiently in the space of features compared to BoW, increasing the performance of density and clustering based algorithms. For OCSVM, D2V also allowed the inference of a better separation hyperplane. Furthermore, as 75.7% of the news on FakeNewsNet is real, RC-SVM was able to infer a purer set of reliable outliers.

For Fake.BR, which is balanced and has six different subjects, of which politics (58%), celebrities (21.4%), and society (17.7%) together make up 97.1% of the news, D2V it also causes an abrupt increase of interest- F_1 in the RC-SVM, OCSVM and the binary reference model algorithms, considering 10% of labeled news. Our intuition is that this representation model was able to group news in feature space more efficiently, considering both the truthfulness and the subject of the news (see Figs. 6 and 7). RC-SVM, in particular, tends to perform well when unlabeled data covers a wide region of the feature space, and different topics. Binary semi-supervised reference model probably have included news from different topics in the initially labeled set, showing better performance in the label propagation of news represented with D2V.

For the Fact-checked News, RC-SVM presents a big change in its results again, mainly with 10 and 20% of labeled data. Looking at Fig. 8, we can see how D2V better splits real and fake news, especially with representations 3 and 4, in which real news is concentrated above or in the center. From this discussion, we can conclude that representing news with document embeddings is a more promising strategy on the news scenario.

5.3.2 Did PUL algorithms perform better than OCL algorithms?

To help answer this question, we plot for each dataset the interest- F_1 (Fig. 9) and the macro F_1 (Fig. 10) considering the D2V representation model and the algorithms' type. Red bars correspond to PUL algorithms, teal bars correspond to OCL, yellow bars

correspond to the binary semi-supervised reference model, and the notations 1, 2, and 3 represent the number of folds used to train the algorithms (10%, 20%, and 30% of labeled data, respectively).

In general, OCL outperforms PUL only for the FakeNewsNet and considering the interest- F_1 measure. The clustering and density algorithms k -Means and k -NND outperform also the binary reference model by more than 30% of interest- F_1 using the D2V representation model. Our hypothesis for the poor interest- F_1 of PUL approaches in FakeNewsNet is that as fake news are clustered in different regions of space (see Fig. 5), true news from different regions can be considered as reliable fake news depending on the initial labeled news set. The same works for reliable fake news. Despite the poor performance of PUL, we can see that our proposed approach using LPHN as the propagation algorithm (PU-LP-LPHN) is competitive with the reference model (see Table 4).

For the Fake.BR dataset, PUL algorithms outperform OCL approaches, both for interest and macro F_1 . Furthermore, PU-LP-LPHN (HT) and OCSVM stand out the binary reference model. In this dataset, which has fake news from different topics, the inclusion of terms in the news network of PU-LP causes the results to increase by 2 to 6% for macro and interest- F_1 . Thus, we can infer that the presence of different data patterns on the network contributed to differentiating the veracity of the news when real and fake news were present in different regions of the feature space. It is worth mentioning that Faustini and Covões (2019) also used Fake.BR in their OCL algorithms experiments, reaching an interest- F_1 of 67% with 90% fake news labeled. We outperform their approach with only 20% of labeled data (see Table 4). To increase the performance of this dataset, we would have to ensure that the initially labeled set has news from 6 different topics, or perform the classification considering one topic at a time, as the classification errors are mainly associated with news belonging to topics with little representativeness (Silva et al., 2020).

For Fact-checked News dataset, PUL also outperforms OCL approaches. Homogeneous and heterogeneous PU-LP have better overall performances, also close to the binary reference model, reaching more than 90% of macro and interest- F_1 .

Therefore, we can conclude that OCL algorithms based on density and clustering are more advantageous when fake news are grouped in different regions of the feature space and have little representativeness in the dataset. On the other hand, PUL approaches can infer more pure sets of fake and real news and obtain results similar or even surpass binary algorithms considering adequate representation models.

5.3.3 Did the proposed approach outperform other PUL, OCL and the reference model for fake news detection?

Table 5 and Table 6 present the average ranking of the algorithms considering the interest and macro F_1 , and using both representations - Bag-of-Words (BoW) and Doc2Vec (D2V). 10%, 20% and 30% indicate the percentages of labeled data. Last column presents the mean of the average rankings. The best performances considering PUL and OCL algorithms are highlighted in bold.

In Table 5, we can notice that our PU-LP-LPHN approach with heterogeneous networks gets a better average ranking considering 10% and 20% of labeled news. For 30% of labeled data, PU-LP-LPHN ties with RC-SVM, getting second by the standard deviation. Considering the mean of the average rankings our approach stands out, being very close to binary semi-supervised algorithms. Also considering the Table 3, we can see

Fig. 7 News from the Fake.BR dataset plotted in two dimensions considering D2V representation, using the t-SNE tool. In **a, c, e** and **g**, Green dots (−1) are real news and orange dots (1) are fake news. In **b, d, f** and **h**, news related to the 6 categories are divided into two classes. In caption, 1, 2, 3, 4, 5, and 6 represent real news from politics, society, celebrities, science, religion and economics. 7, 8, 9, 10, 11 and 12 are fake news following the same order (Color figure online)

that our approach obtains the best performance for Fake.BR and Fact-checked News collections using BoW. As discussed previously (see Sect. 5.3.1) fake news represented with BoW is more dispersed in the feature space. Therefore, the inclusion of news terms patterns in the news network was able to improve the classification performance.

Given the interest- F_1 and the D2V representation model, OCSVM has a higher average ranking for 10% of labeled data. However, the algorithm presents great instability in the standard deviation. For 20% and 30% of labeled data, PU-LP-LPHN (heterogeneous) and PU-LP-LLGC (homogeneous) have better rankings. Considering the average (last column), PU-LP-LPHN also stands out, with a low standard deviation and close to the binary semi-supervised approach (BL).

Analyzing the results activated in Table 4, we can see that for the Fake.BR, in which news are grouped in the characteristics space both by veracity and by subject (See Fig. 7), the inclusion of terms in the news network also tends to increase the PU-LP classification results. Considering the Fact-checked News, in Fig. 8 we can see that the D2V is able to effectively separate real and false news. Therefore, the inclusion of terms in the network does not tend to improve the results achieved by the homogeneous networks approach.

Considering macro F_1 (Table 6), PU-LP-LPHN stands out using both BoW and D2V. The only exception is for 20% of labeled data, where PU-LP-LLGC gets a better average ranking.

5.3.4 Did the amount of labeled fake news significantly increase the interest- F_1 ?

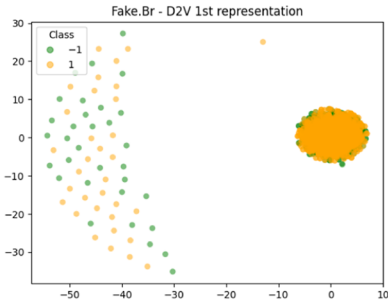
Based on Figs. 9 and 10, we can notice that, in general, the results do not present great variation considering the number of folds used in training. Even algorithms like OCSVM reach maximum macro F_1 and interest- F_1 for Fake.BR using only 10% of labeled data. These results encourage the search for OCL and PUL approaches to news classification capable of detecting fake news with very little labeled data.

5.3.5 Which parameters of the proposed approach achieved the best results for interest- F_1 ?

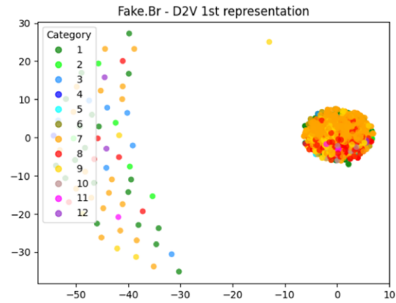
Considering that our proposed approach achieved better overall performance using PU-LP in document and term networks, in Figs. 11, 12 and 13 we present a parameter analysis considering the 100 best results of interest- F_1 obtained by dataset. 10%, 20% and 30% indicate the amount of fake news initially labeled.

Considering the representation models, those that activated the best interest- F_1 were representation model 3 (for FakeNewsNet) and 4 (for Fake.BR and Fact-checked News). Both models use the concatenation of word vectors to create the hidden layer's output. As for adding terms in the news network, adding unigrams and bigrams provided better results for FakeNewsNet and Fake.BR. As for Fact-checked News, the addition of unigrams or bigrams obtained similar results.

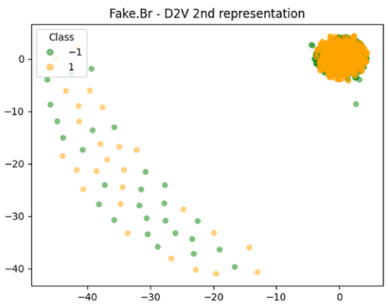
We also analyze some parameters of the PU-LP algorithm. The values of the λ parameter, which controls the size of the generated interest and non-interest reliable sets, provided



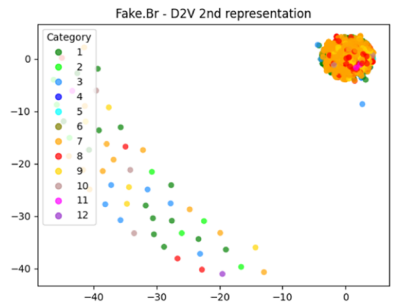
(a)



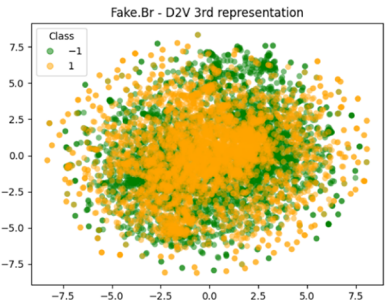
(b)



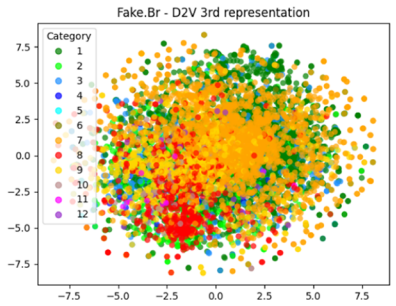
(c)



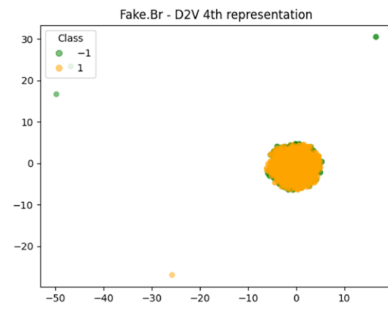
(d)



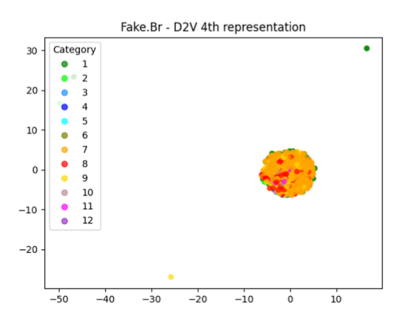
(e)



(f)



(g)



(h)

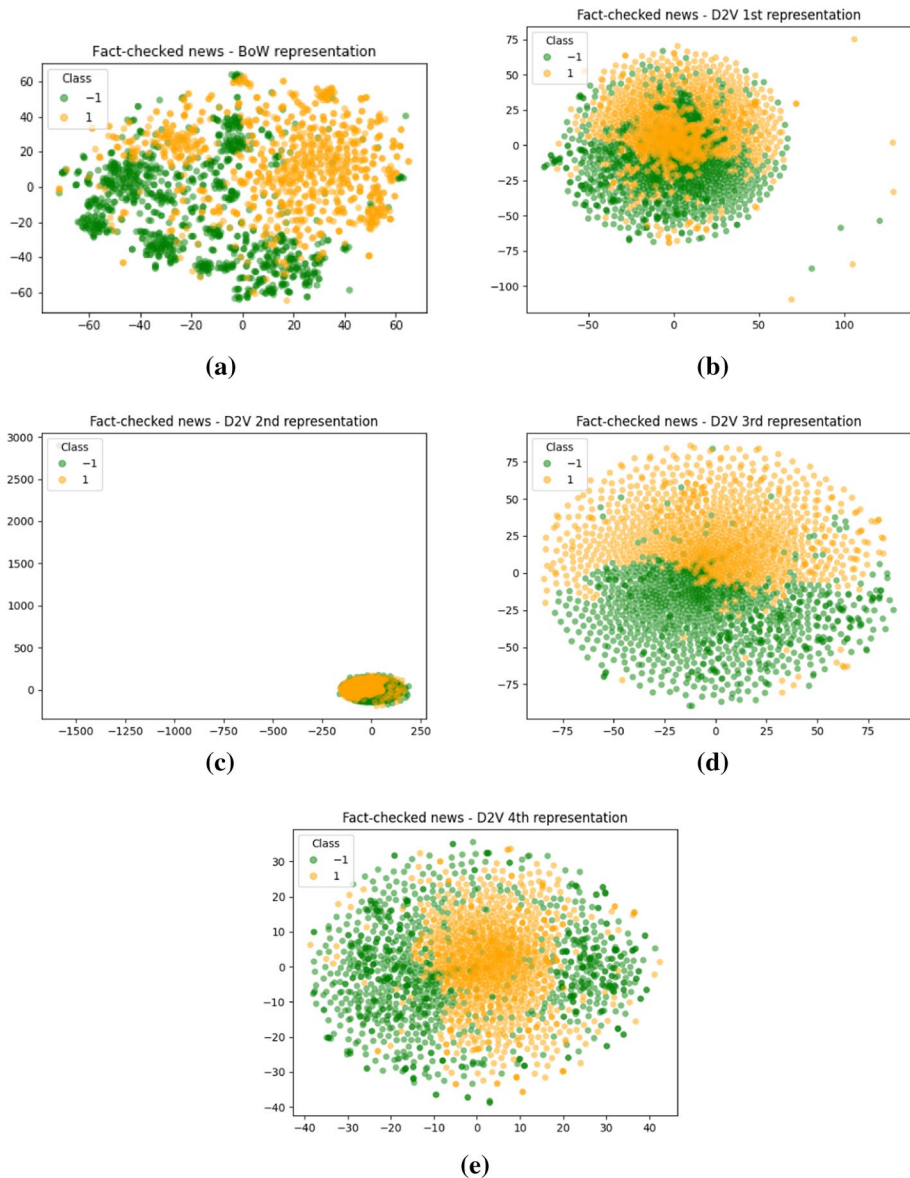


Fig. 8 News from the Fact-checked news dataset plotted in two dimensions, using the t-SNE tool. **a** corresponds to the Bag-of-Words representation model, **b–e** correspond to the four representations of document embeddings generated with Doc2Vec. Green dots (–1) are real news and orange dots (1) are fake news (Color figure online)

similar results in FakeNewsNet. For Fake.BR, the best results were obtained with $\lambda = 0.6$. As for Fact-checked News, the parameter enabled better results with $\lambda = 0.8$. We also analyzed the k parameter, corresponding to the k -NN news network. For FakeNewsNet, 5-NN

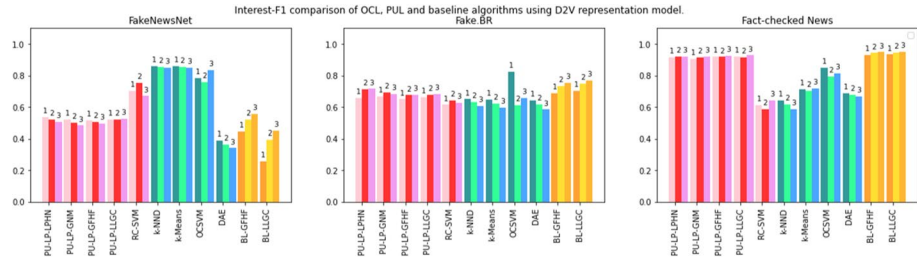


Fig. 9 Interest- F_1 considering D2V representation model and the algorithms’ type. Red bars correspond to PUL algorithms, blue bars correspond to OCL, and yellow bars correspond to the binary semi-supervised reference model, and 1, 2, and 3 represent the number of folds used to train the algorithms (Color figure online)

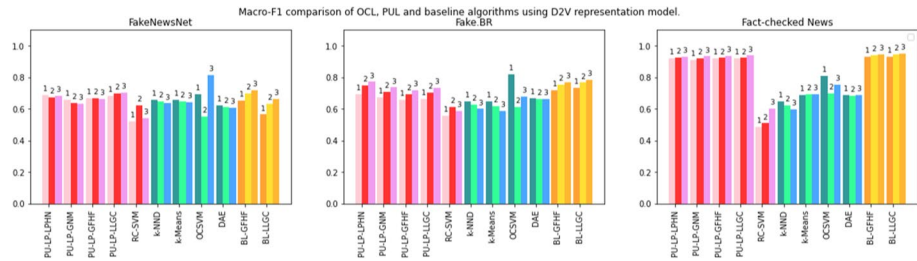


Fig. 10 Macro F_1 considering D2V representation model and the algorithms’ type. Red bars correspond to PUL algorithms, blue bars correspond to OCL, and yellow bars correspond to the binary semi-supervised reference model, and 1, 2, and 3 represent the number of folds used to train the algorithms (Color figure online)

networks performed better, while for Fake.BR and Fact-checked News, the best parameter was $k = 6$.

The parameter α that controls the influence of paths on the Katz index equal to 0.005 provided the best interest- F_1 for FakeNewsNet and Fact-checked News datasets. For the Fake.BR, $\alpha = [0.005, 0.01, 0.02]$ provided similar results. We also analyzed the m parameter, which corresponds to the number of iterations that the task of inferring real and fake reliable sets is performed. For all databases, $m = 2$ provided the best performances.

6 Conclusion and future work

Researches related to the automatic detection of fake news are essential today, mainly due to this task’s intrinsic dynamics characteristics. This work proposed a new approach for detecting fake news based on PU-LP to minimize the news labeling effort. In addition to using a low amount of labeled data, it extracts information from unlabeled data and relevant terms to assist label propagation algorithms in differentiating fake (interest) and real news (not interest). The approach was evaluated considering balanced and unbalanced datasets, with news in Portuguese and English containing only one or several subjects. The results obtained were compared with traditional algorithms in the literature for One-Class learning and Positive and Unlabeled learning. We also used a semi-supervised binary

Table 5 Average ranking and standard deviation of the OCL, PUL and binary (BIN) algorithms, considering 10%, 20% and 30% of labeled data for the interest- F_1 results. Last column presents the mean of the average rankings. The best performances considering PUL and OCL algorithms are highlighted in bold

Interest- F_1						
Rep. Model	Type	Algorithm	10%	20%	30%	Avg \pm Std
BoW	PUL	PU-LP-LPHN	7.7 \pm 3.1	8.0 \pm 4.0	10.7 \pm 6.7	8.8 \pm 4.6
		PU-LP-GNM	9.3 \pm 5.0	10.3 \pm 5.5	11.7 \pm 6.7	10.4 \pm 5.7
		PU-LP-GFHF	15.0 \pm 2.6	10.0 \pm 2.6	10.7 \pm 4.2	11.9 \pm 3.2
		PU-LP-LLGC	11.7 \pm 3.1	12.7 \pm 3.8	12.2 \pm 1.3	12.2 \pm 2.7
		RC-SVM	22.0 \pm 0.0	20.7 \pm 1.5	10.7 \pm 4.0	17.8 \pm 1.9
	OCL	k-NND	11.0 \pm 5.6	12.0 \pm 6.2	14.7 \pm 5.9	12.6 \pm 5.9
		k-Means	11.7 \pm 4.9	13.3 \pm 4.7	15.7 \pm 4.2	13.6 \pm 4.6
		OCSVM	17.3 \pm 3.1	18.7 \pm 4.2	19.0 \pm 4.4	18.3 \pm 3.9
		DAE	14.7 \pm 2.5	16.0 \pm 2.0	18.3 \pm 1.5	16.3 \pm 2.0
	BIN	BL-GFHF	13.7 \pm 5.0	8.0 \pm 1.7	5.7 \pm 2.1	9.1 \pm 2.9
		BL-LLGC	16.0 \pm 7.0	13.3 \pm 5.7	10.5 \pm 3.3	13.3 \pm 5.3
D2V	PUL	PU-LP-LPHN	6.7 \pm 1.5	5.0 \pm 3.5	7.0 \pm 4.6	6.2 \pm 3.2
		PU-LP-GNM	6.7 \pm 1.2	9.0 \pm 5.2	9.8 \pm 5.6	8.5 \pm 4.0
		PU-LP-GFHF	8.0 \pm 4.4	8.3 \pm 4.5	9.7 \pm 5.5	8.7 \pm 4.8
		PU-LP-LLGC	6.7 \pm 2.5	7.7 \pm 3.1	6.8 \pm 3.3	7.1 \pm 3.0
		RC-SVM	14.3 \pm 9.1	12.3 \pm 8.5	13.3 \pm 8.6	13.3 \pm 8.7
	OCL	k-NND	10.7 \pm 9.0	11.3 \pm 9.6	13.0 \pm 10.8	11.7 \pm 9.8
		k-Means	10.7 \pm 8.7	12.0 \pm 8.7	12.0 \pm 8.7	11.6 \pm 8.7
		OCSVM	5.7 \pm 6.4	12.0 \pm 8.5	9.7 \pm 5.8	9.1 \pm 6.9
		DAE	18.3 \pm 2.1	20.0 \pm 1.7	21.0 \pm 1.0	19.8 \pm 1.6
	BIN	BL-GFHF	7.3 \pm 8.4	4.7 \pm 4.6	3.7 \pm 2.9	5.2 \pm 5.3
		BL-LLGC	8.0 \pm 11.3	7.7 \pm 11.5	7.3 \pm 11.0	7.7 \pm 11.3

reference model to analyze the results, which classified news considering an initial set of labeled real and fake news.

The dataset configurations influenced the behavior of the analyzed algorithms. For the unbalanced dataset FakeNewsNet, OCL algorithms based on distance and density provided the best results, achieving higher performance than the binary semi-supervised algorithm with only 10% of labeled news. For Fake.BR, the OCSVM and PU-LP-LPHN algorithms performed better, staying close to or exceeding the results achieved by the binary approach, even in a scenario containing 6 different subjects. For Fact-checking news that has only policy news, in which data are well behaved according to the analyzes performed with t-SNE, PU-LP with the propagation algorithms GFHF and LLGC achieved good results, inferring more pure sets of reliable interest and non-interest news.

Among the methods evaluated in this paper, PU-LP presented, in general, a better average ranking of the F_1 measure, considering LPHN label propagation algorithm. Future directions consist of improving PU-LP. Since PU-LP is a network-based model in which nodes are news, its interpretability can be explored to recover neighboring

Table 6 Average ranking and standard deviation of the OCL, PUL and binary (BIN) reference model algorithms, considering 10%, 20% and 30% of labeled data for the macro F_1 results. Last column presents the mean of the average rankings. The best performances considering PUL and OCL algorithms are highlighted in bold

Macro F_1							
Rep. Model	Type	Algorithm	10%	20%	30%	Avg \pm Std	
BoW	PUL	PU-LP-LPHN	6.3 \pm 0.6	8.0 \pm 1.0	10.0 \pm 6.1	8.1 \pm 2.6	
		PU-LP-GNM	11.0 \pm 3.6	11.3 \pm 5.1	12.3 \pm 5.9	11.6 \pm 4.9	
		PU-LP-GFHF	13.7 \pm 2.9	12.0 \pm 2.0	13.3 \pm 2.3	13.0 \pm 2.4	
		PU-LP-LLGC	12.7 \pm 2.9	10.0 \pm 1.7	10.7 \pm 0.6	11.1 \pm 1.7	
		RC-SVM	22.0 \pm 0.0	21.7 \pm 0.6	15.0 \pm 6.6	19.6 \pm 2.4	
	OCL	k-NND	14.5 \pm 2.3	14.8 \pm 2.4	16.0 \pm 2.6	15.1 \pm 2.4	
		k-Means	13.8 \pm 2.8	14.8 \pm 2.8	16.3 \pm 2.5	15.0 \pm 2.7	
		OCSVM	18.3 \pm 1.2	19.3 \pm 3.1	20.0 \pm 2.0	19.2 \pm 2.1	
	BIN	DAE	13.7 \pm 8.7	9.7 \pm 4.9	12.3.0 \pm 4.6	11.9 \pm 6.1	
		BL-GFHF	9.7 \pm 5.0	7.0 \pm 3.5	7.0 \pm 3.6	7.9 \pm 4.0	
	D2V	PUL	BL-LLGC	13.3 \pm 5.0	9.0 \pm 4.6	9.3 \pm 3.5	10.6 \pm 4.4
			PU-LP-LPHN	3.7 \pm 1.5	4.0 \pm 1.7	4.3 \pm 2.1	4.0 \pm 1.8
PU-LP-GNM			6.3 \pm 1.5	8.3 \pm 5.9	8.0 \pm 6.1	7.6 \pm 4.5	
PU-LP-GFHF			6.7 \pm 3.5	5.8 \pm 1.3	6.7 \pm 2.5	6.4 \pm 2.4	
PU-LP-LLGC			5.3 \pm 3.2	3.5 \pm 2.2	4.0 \pm 1.0	4.3 \pm 2.1	
OCL		RC-SVM	20.7 \pm 0.6	19.3 \pm 1.5	20.7 \pm 0.6	20.2 \pm 0.9	
		k-NND	14.0 \pm 5.6	15.0 \pm 5.0	17.0 \pm 5.0	15.3 \pm 5.2	
		k-Means	14.3 \pm 4.5	15.0 \pm 3.6	17.3 \pm 5.5	15.6 \pm 4.5	
		OCSVM	5.0 \pm 6.9	19.3 \pm 2.1	9.3 \pm 7.2	11.2 \pm 5.4	
BIN		DAE	14.3 \pm 6.4	17.0 \pm 3.5	17.7 \pm 1.5	16.3 \pm 3.8	
		BL-GFHF	6.0 \pm 6.1	2.0 \pm 0.0	2.3 \pm 0.6	3.4 \pm 2.2	
			BL-LLGC	7.7 \pm 10.7	6.0 \pm 8.7	3.3 \pm 4.0	5.7 \pm 7.8

nodes, notifying the user of news that are similar to the previously classified ones. Furthermore, the network-based structure allows incorporating new features, such as linguistic features already covered in the literature or even features related to the sentiment expressed in the news, that allow differentiating news that have similar content but different intentions. Also, the proposed approach can be evaluated using more powerful representation models, such as context-based, and label propagation algorithms.

Acknowledgements This work was supported by Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) [grant number 10662147/D], Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) [grant number 2019/25010-5] and Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) [grant number 433082/2018-6].

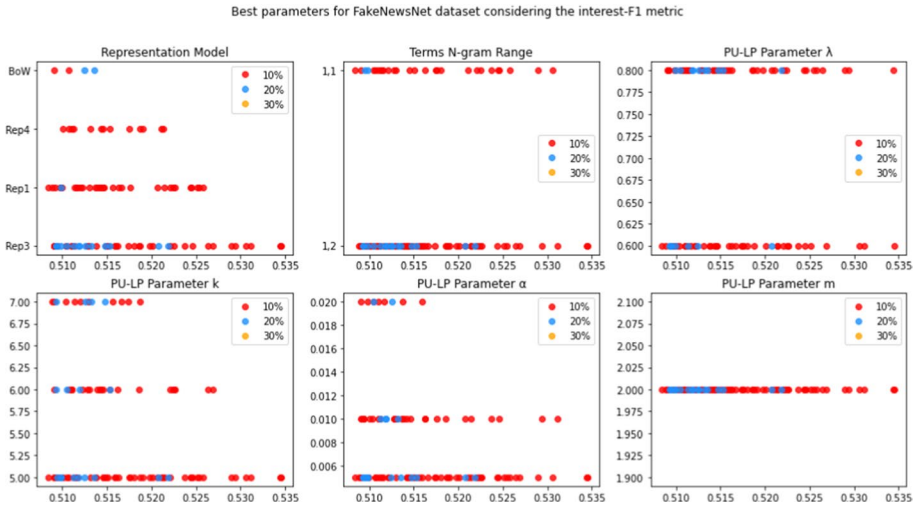


Fig. 11 Parameter evaluation considering the best 100 results achieved by our proposed PU-LP approach in FakeNewsNet, using heterogeneous networks, and LPHN and GNetMine label propagation algorithms. First chart show the bests models, considering BoW and the four D2V representations. Second chart presents the representative terms range, in which (1,1) indicates the inclusion of only unigrams and (1,2) unigrams and bigrams. Third, fourth, fifth and sixth charts are about the PU-LP parameters λ , k , α and m . 10%, 20% and 30% indicate the percentage of fake news labeled

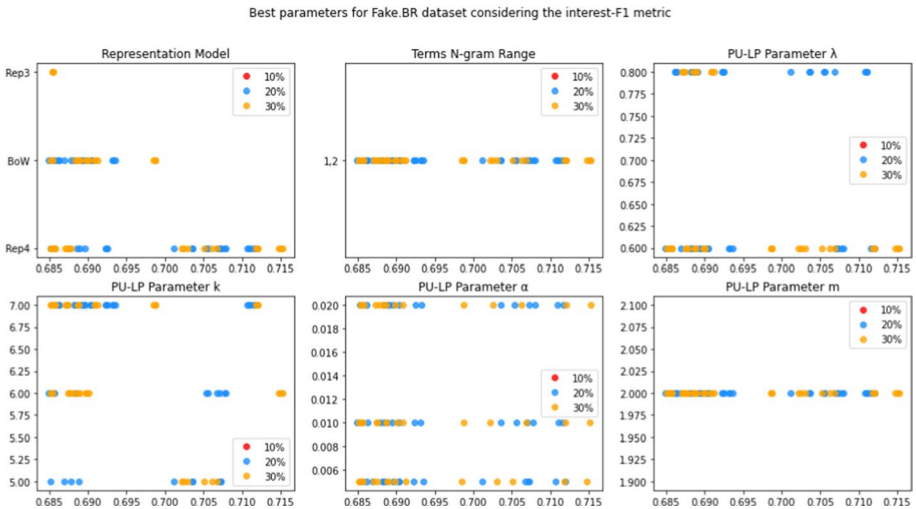


Fig. 12 Parameter evaluation considering the best 100 results achieved by our proposed PU-LP approach in Fake.BR, using heterogeneous networks and LPHN and GNetMine label propagation algorithms. First chart show the bests models, considering BoW and the four D2V representations. Second chart presents the representative terms range, in which (1,1) indicates the inclusion of only unigrams and (1,2) unigrams and bigrams. Third, fourth, fifth and sixth charts are about the PU-LP parameters λ , k , α and m . 10%, 20% and 30% indicate the percentage of fake news labeled

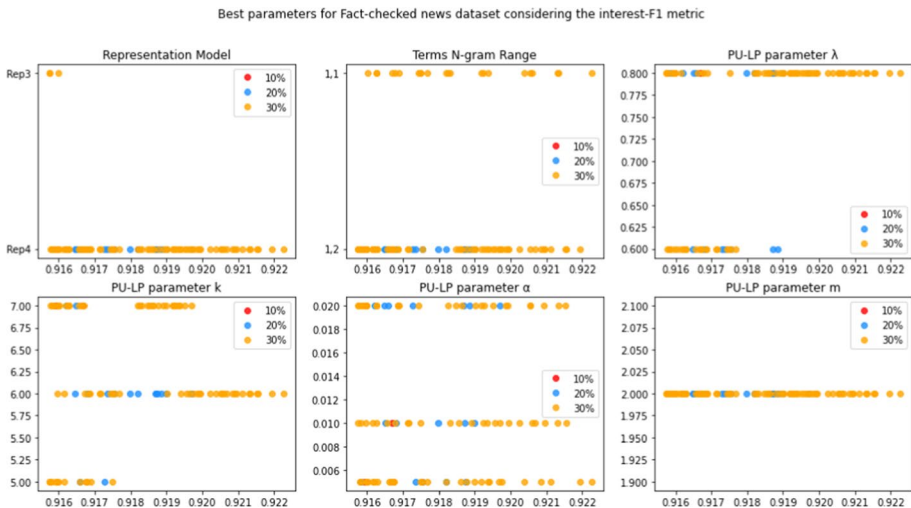


Fig. 13 Parameter evaluation considering the best 100 results achieved by our proposed PU-LP approach in Fact-checked News, using heterogeneous networks and LPHN and GNetMine label propagation algorithms. First chart show the bests models, considering BoW and the four D2V representations. Second chart presents the representative terms range, in which (1,1) indicates the inclusion of only unigrams and (1,2) unigrams and bigrams. Third, fourth, fifth and sixth charts are about the PU-LP parameters λ , k , α and m . 10%, 20% and 30% indicate the percentage of fake news labeled

Author Contributions Mariana Caravanti de Souza and Bruce Neves dos Santos contributed to conceptualizing, designing, and implementing the proposed approach PU-LP for heterogeneous networks. Mariana Caravanti de Souza contributed to the news datasets curation. Rafael Geraldeli Rossi and Bruno Magalhães Nogueira contributed to formal analysis. Mariana Caravanti, Bruno Magalhães Nogueira, Rafael Geraldeli Rossi, Ricardo Marcondes Marcacini, and Solange Oliveira Rezende contributed on results analysis and in the manuscript writing and revision.

Availability of data and code Our PU-LP approach using heterogeneous networks and the three Portuguese and English datasets are available in the online repository on GitHub: <https://github.com/marianacaravanti/PU-LP-for-fake-news-detection>.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Ethics approval Not Applicable.

Consent to participate Not Applicable.

Consent for publication Not Applicable.

References

- Aggarwal, C. C. (2018). *Neural networks and deep learning*. Springer, 10, 978–3.
- Aggarwal, C. C., & Li, N. (2011). On node classification in dynamic content-based networks. In *Proceedings of the 2011 SIAM international conference on data mining* (pp. 355–366). SIAM.

- Ahmed, H., Traore, I., & Saad, S. (2017). Detection of online fake news using n-gram analysis and machine learning techniques. In *ISDDC 2017: International conference on intelligent, secure, and dependable systems in distributed and cloud environments* (pp. 127–138).
- Akoglu, L., Tong, H., & Koutra, D. (2015). Graph based anomaly detection and description: A survey. *Data Mining and Knowledge Discovery*, 29(3), 626–688.
- Alam, S., Sonbhadra, S. K., Agarwal, S., & Nagabhushan, P. (2020). One-class support vector classifiers: A survey. *Knowledge-Based Systems*, 196.
- Barbier, G., Feng, Z., Gundecha, P., & Liu, H. (2013). Provenance data in social media. *Synthesis Lectures on Data Mining and Knowledge Discovery*, 4(1), 1–84.
- Bekker, J., & Davis, J. (2020). Learning from positive and unlabeled data: A survey. *Machine Learning*, 109(4), 719–760.
- Bellinger, C., Sharma, S., Zarane, O. R., & Japkowicz, N. (2017). Sampling a longer life: Binary versus one-class classification revisited. In *International workshop on learning with imbalanced domains: Theory and applications* (pp. 64–78).
- Bondielli, A., & Marcelloni, F. (2019). A survey on fake news and rumour detection techniques. *Information Sciences*, 497, 38–55.
- Breve, F., Zhao, L., Quiles, M., Pedrycz, W., & Liu, J. (2012). Particle competition and cooperation in networks for semi-supervised learning. *TKDE 2012: IEEE Transactions on Knowledge and Data Engineering*, 24(9), 1686.
- Chakravarthy, S., Venkatachalam, A., Telang, A., & Aery, M. (2014). Infosift: A novel, mining-based framework for document classification. *International Journal of Next-Generation Computing*, 5(2), 84–122.
- Chalapathy, R., & Chawla, S. (2019). Deep learning for anomaly detection: A survey. arXiv preprint [arXiv:190103407](https://arxiv.org/abs/190103407).
- Chen, W., Yeo, C. K., Lau, C. T., & Lee, B. S. (2016). Behavior deviation: An anomaly detection view of rumor preemption. In *IEMCON 2016: information technology, electronics and mobile communication conference* (pp. 1–7). IEEE.
- Datta, P. (1998). Characteristic concept representations. PhD thesis, Irvine, California, United States of America.
- De Sales Santos, R. L., & Pardo, T. A. S. (2020). Fact-checking for portuguese: Knowledge graph and google search-based methods. In *PROPOR 2020: International conference on computational processing of the Portuguese language* (pp. 195–205). Springer.
- Deepak, P., Chakraborty, T., Long, C., et al. (2021). *Data science for fake news: Surveys and perspectives* (Vol. 42). Springer Nature.
- Delalleau, O., Bengio, Y., & Le Roux, N. (2005). Efficient non-parametric function induction in semi-supervised learning. In *AISTATS 2005: International conference on artificial intelligence and statistics* (Vol. 27).
- DePaulo, B. M., Charlton, K., Cooper, H., Lindsay, J. J., & Muhlenbruck, L. (1997). The accuracy-confidence correlation in the detection of deception. *Personality and Social Psychology Review*, 1(4), 346–357.
- Eltanbouly, S., Bashendy, M., AlNaimi, N., Chkirkbene, Z., & Erbad, A. (2020). Machine learning techniques for network anomaly detection: A survey. In *ICIOT 2020: International conference on informatics, IoT, and enabling technologies* (pp. 156–162). IEEE.
- Faustini, P., & Covões, T. F. (2019). Fake news detection using one-class classification. In *BRACIS 2019: Brazilian conference on intelligent systems* (pp. 592–597). IEEE.
- Feldman, R., Sanger, J., et al. (2007). *The text mining handbook: Advanced approaches in analyzing unstructured data*. Cambridge University Press.
- Frank, M. G., Feeley, T. H., Paolantonio, N., & Servoss, T. J. (2004). Individual and small group accuracy in judging truthful and deceptive communication. *Group Decision and Negotiation*, 13(1), 45–59.
- Garrette, D., & Klein, E. (2009). An extensible toolkit for computational semantics. In *IWCS 2009: International conference on computational semantics* (pp. 116–127).
- Gôlo, M., Marcacini, R., & Rossi, R. (2020). Uma extensa avaliação empírica de técnicas de pré-processamento e algoritmos de aprendizado supervisionado de uma classe para classificação de texto. In *ENIAC 2019: Encontro Nacional de Inteligência Artificial e Computacional*, SBC (pp. 262–273).
- Greifeneder, R., Jaffe, M., Newman, E., & Schwarz, N. (2021). *The psychology of fake news: Accepting, sharing, and correcting misinformation*. London: Routledge.
- Guacho, G. B., Abdali, S., Shah, N., & Papalexakis, E. E. (2018). Semi-supervised content-based detection of misinformation via tensor embeddings. In *ASONAM 2018: ACM international conference on advances in social networks analysis and mining* (pp. 322–325). IEEE.

- Hassan, N., Gomaa, W., Khoriba, G., & Haggag, M. (2020). Credibility detection in twitter using word n-gram analysis and supervised machine learning techniques. *International Journal of Intelligent Engineering and Systems*, 13(1), 291–300.
- Horne, B. D., & Adali, S. (2017). This just in: Fake news packs a lot in title, uses simpler, repetitive content in text body, more similar to satire than real news. In *ICWSM 2017: International AAAI conference on web and social media*.
- Hua, Z., Yang, Y., & Qiu, H. (2021). Node influence-based label propagation algorithm for semi-supervised learning. *Neural Computing and Applications*, 33(7), 2753–2768.
- Jaskie, K., & Spanias, A. (2019). Positive and unlabeled learning algorithms and applications: A survey. In *IISA 2019: International conference on information, intelligence, systems and applications* (pp. 1–8). IEEE.
- Ji, M., Sun, Y., Danilevsky, M., Han, J., & Gao, J. (2010). Graph regularized transductive classification on heterogeneous information networks. In *PKDD 2010: Joint European conference on machine learning and knowledge discovery in databases* (pp. 570–586). Springer.
- Kang, S., Hwang, J., & Yu, H. (2020). Multi-modal component embedding for fake news detection. In *IMCOM 2020: 14th international conference on ubiquitous information management and communication* (pp. 1–6). IEEE.
- Katz, L. (1953). A new status index derived from sociometric analysis. *Psychometrika*, 18(1), 39–43.
- Khan, S. S., & Madden, M. G. (2014). One-class classification: Taxonomy of study and review of techniques. *The Knowledge Engineering Review*, 29(3), 345–374.
- Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. In Bengio, Y., & LeCun, Y. (Eds.), *ICLR 2019: International Conference on Learning Representations*.
- Lazhar, F. (2019). Fuzzy clustering-based semi-supervised approach for outlier detection in big text data. *Progress in Artificial Intelligence*, 8(1), 123–132.
- Le, Q., & Mikolov, T. (2014). Distributed representations of sentences and documents. In *ICML 2014: International conference on machine learning* (pp. 1188–1196).
- Li, H., Chen, Z., Liu, B., Wei, X., & Shao, J. (2014). Spotting fake reviews via collective positive-unlabeled learning. In *ICDM 2014: International conference on data mining* (pp. 899–904). IEEE.
- Li, Q., Hu, Q., Lu, Y., Yang, Y., & Cheng, J. (2019). Multi-level word features based on CNN for fake news detection in cultural communication. *Personal and Ubiquitous Computing* (pp. 1–14).
- Li, X., & Liu, B. (2003). Learning to classify texts using positive and unlabeled data. In *IJCAI 2003: International joint conferences on artificial intelligence* (Vol. 3, pp. 587–592).
- Liang, D. M., & Li, Y. F. (2018). Lightweight label propagation for large-scale network data. In *International joint conference on artificial intelligence* (pp. 3421–3427).
- Liu, B., Lee, W. S., Yu, P. S., & Li, X. (2002). Partially supervised classification of text documents. In *ICML 2002: International conference on machine learning* (Vol. 2, pp. 387–394).
- Liu, Y., & Wu, Y. F. B. (2020). Fned: A deep network for fake news early detection on social media. *TOIS 2020: ACM Transactions on Information Systems*, 38(3), 1–33.
- Lü, L., Jin, C. H., & Zhou, T. (2009). Similarity index based on local paths for link prediction of complex networks. *Physical Review E*, 80(4), 046122.
- Ma, S., & Zhang, R. (2017). Pu-lp: A novel approach for positive and unlabeled learning by label propagation. In *ICMEW 2017: International conference on multimedia & expo workshops* (pp. 537–542). IEEE.
- Manevitz, L., & Yousef, M. (2007). One-class document classification via neural networks. *Neurocomputing*, 70(7–9), 1466–1481.
- Manevitz, L. M., & Yousef, M. (2001). One-class svms for document classification. *Journal of Machine Learning Research*, 2(Dec), 139–154.
- Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to information retrieval*. Cambridge University Press.
- Martinčić-Ipšić, S., Miličić, T., & Todorovski, L. (2019). The influence of feature representation of text on the performance of document classification. *Applied Sciences*, 9(4), 743.
- Meel, P., & Vishwakarma, D. K. (2019). Fake news, rumor, information pollution in social media and web: A contemporary survey of state-of-the-arts, challenges and opportunities. *Expert Systems with Applications*, 106991.
- Meneses Silva, C. V., Silva Fontes, R., & Colaço Júnior, M. (2021). Intelligent fake news detection: A systematic mapping. *Journal of Applied Security Research*, 16(2), 168–189.
- Mihalcea, R., Strapparava, C., & Pulman, S. (2010). Computational models for incongruity detection in humour. In *CICLING 2010: International conference on intelligent text processing and computational linguistics* (pp. 364–374). Springer.

- Muir, A. (2005). *Lean six sigma statistics: Calculating process efficiencies in transactional project*. McGraw Hill Professional – Six Sigma Operational Methods Series.
- Osatuyi, B. (2013). Information sharing on social media sites. *Computers in Human Behavior*, 29(6), 2622–2631.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, Édouard (2011). Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(85), 2825–2830
- Perera, P., Oza, P., & Patel, V. M. (2021). One-class classification: A survey. arXiv preprint [arXiv:210103064](https://arxiv.org/abs/210103064).
- Pérez-Rosas, V., Kleinberg, B., Lefevre, A., & Mihalcea, R. (2017). Automatic detection of fake news. arXiv preprint [arXiv:170807104](https://arxiv.org/abs/170807104).
- Pimentel, T., Monteiro, M., Viana, J., Veloso, A., & Ziviani, N. (2018). A generalized active learning approach for unsupervised anomaly detection. *Stat*, 1050, 23.
- Pita, M., & Pappa, G. L. (2018). Strategies for short text representation in the word vector space. In *BRACIS 2019: Brazilian conference on intelligent systems* (pp. 266–271). IEEE.
- Rashkin, H., Choi, E., Jang, J. Y., Volkova, S., & Choi, Y. (2017). Truth of varying shades: Analyzing language in fake news and political fact-checking. In *EMNLP 2017: Proceedings of the conference on empirical methods in natural language processing* (pp. 2931–2937).
- Reis, J., Correia, A., Murai, F., Veloso, A., & Benevenuto, F. (2019). Explainable machine learning for fake news detection. In *WebSci 2019: ACM conference on web science* (pp. 17–26). ACM.
- Rossi, R. G. (2016). *Classificação automática de textos por meio de aprendizado de máquina baseado em redes*. PhD thesis. Brazil: Universidade de São Paulo.
- Rossi, R. G., de Andrade, Lopes A., de Paulo, Faleiros T., & Rezende, S. O. (2014). Inductive model generation for text classification using a bipartite heterogeneous network. *Journal of Computer Science and Technology*, 29(3), 361–375.
- Rossi, R. G., Rezende, S. O., & de Andrade Lopes, A. (2015). Term network approach for transductive classification. In *CICLing 2015: International conference on intelligent text processing and computational linguistics* (pp. 497–515). Springer.
- Rossi, R. G., de Andrade, Lopes A., & Rezende, S. O. (2016). Optimization and label propagation in bipartite heterogeneous networks to improve transductive classification of texts. *Information Processing & Management*, 52(2), 217–257.
- Rubin, V. L., Conroy, N., Chen, Y., & Cornwell, S. (2016). Fake news or truth? using satirical cues to detect potentially misleading news. In *Proceedings of the second workshop on computational approaches to deception detection* (pp. 7–17).
- Ruff, L., Zemlyanskiy, Y., Vandermeulen, R., Schnake, T., & Kloft, M. (2019). Self-attentive, multi-context one-class classification for unsupervised anomaly detection on text. In *Meeting of the association for computational linguistics* (pp. 4061–4071).
- Salmazzo, N. (2016). *Classificação one-class para predição de adaptação de espécies em ambientes desconhecidos*. PhD thesis. Brazil: Universidade Federal do ABC.
- Salton, G. (1989). *Automatic text processing: The transformation, analysis, and retrieval of information by computer*. Addison-Wesley series in computer science, Addison-Wesley.
- Salton, G., et al. (1971). *The smart system-experiments in automatic document processing*. Prentice-Hall.
- Santos, B. N. (2018). *Classificação transdutiva de eventos usando redes heterogêneas*. Master's thesis. Universidade Federal de Mato Grosso do Sul.
- Sharma, K., Qian, F., Jiang, H., Ruchansky, N., Zhang, M., & Liu, Y. (2019). Combating fake news: A survey on identification and mitigation techniques. *TIST 2019: ACM Transactions on Intelligent Systems and Technology*, 10(3), 1–42.
- Shi, C., & Philip, S. Y. (2017). *Heterogeneous Information Network Analysis and Applications*. Data Analytics: Springer.
- Shu, K., Sliva, A., Wang, S., Tang, J., & Liu, H. (2017). Fake news detection on social media: A data mining perspective. *SIGKDD 2017: Special Interest Group on Knowledge Discovery in Data Explorations Newsletter*, 19(1), 22–36.
- Shu, K., Bernard, H. R., & Liu, H. (2019). Studying fake news via network analysis: detection and mitigation. In *Emerging research challenges and opportunities in computational social network analysis and mining* (pp. 43–65). Springer.
- Shu, K., Mahudeswaran, D., Wang, S., Lee, D., & Liu, H. (2020). Fakenewsnet: A data repository with news content, social context, and spatiotemporal information for studying fake news on social media. *Big Data*, 8(3), 171–188.

- Silva, R. M., Santos, R. L., Almeida, T. A., & Pardo, T. A. (2020). Towards automatically filtering fake news in Portuguese. *Expert Systems with Applications*, 146, 113199.
- Sonbhadra, S. K., Agarwal, S., & Nagabhusan, P. (2020). Target specific mining of covid-19 scholarly articles using one-class approach. *Chaos, Solitons & Fractals*, 140, 110155.
- Tan, P., Steinbach, M., Karpatne, A., & Kumar, V. (2019). Anomaly detection. In *Introduction to data mining*, Pearson, Oxford, chap 10 (pp. 651–684).
- Tax, D. M. J. (2001). One-class classification: Concept-learning in the absence of counter-examples. PhD thesis, Delft University of Technology.
- Van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(11), 2579–2605.
- Van Engelen, J. E., & Hoos, H. H. (2020). A survey on semi-supervised learning. *Machine Learning*, 109(2), 373–440. <https://doi.org/10.1007/s10994-019-05855-6>
- Volkova, S., & Jang, J. Y. (2018). Misleading or falsification: Inferring deceptive strategies and types in online news and social media. In *WWW 18: Companion proceedings of the the web conference* (pp. 575–583).
- Wang, H., Bah, M. J., & Hammad, M. (2019). Progress in outlier detection techniques: A survey. *IEEE Access*, 7, 107964–108000.
- Wang, Y., Ma, F., Jin, Z., Yuan, Y., Xun, G., Jha, K., Su, L., & Gao, J. (2018). Eann: Event adversarial neural networks for multi-modal fake news detection. In *SIGKDD 2028: ACM international conference on knowledge discovery & data mining* (pp. 849–857).
- Xu, Y., Li, L., Huang, J., Yin, Y., Shao, W., Mai, Z., & Hei, L. (2019). Positive-unlabeled learning for sentiment analysis with adversarial training. In *CollaborateCom 2019: International conference on collaborative computing: networking, applications and worksharing* (pp. 364–379). Springer.
- Yan, D., Li, K., Gu, S., & Yang, L. (2020). Network-based bag-of-words model for text classification. *IEEE Access*, 8, 82641–82652.
- Yang, C., Xiao, Y., Zhang, Y., Sun, Y., & Han, J. (2020). Heterogeneous network representation learning: A unified framework with survey and benchmark. *TKDE 2020: IEEE Transactions on Knowledge and Data Engineering*.
- Yu, S., & Li, C. (2007). PE-PUC: A graph based PU-Learning approach for text classification. In *MLDM 2007: International workshop on machine learning and data mining in pattern recognition* (pp. 574–584). Springer.
- Zhang, C., Ren, D., Liu, T., Yang, J., & Gong, C. (2019). Positive and unlabeled learning with label disambiguation. In *IJCAI 2019: International joint conference on artificial intelligence* (pp. 1–7).
- Zhang, X., & Ghorbani, A. A. (2020). An overview of online fake news: Characterization, detection, and discussion. *Information Processing & Management*, 57(2), 102025.
- Zhao, J., Cao, N., Wen, Z., Song, Y., Lin, Y. R., & Collins, C. (2014). # fluxflow: Visual analysis of anomalous information spreading on social media. *Transactions on Visualization and Computer Graphics*, 20(12), 1773–1782.
- 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.
- Zhou, D., Bousquet, O., Lal, T. N., Weston, J., & Schölkopf, B. (2004a). Learning with local and global consistency. In *Advances in neural information processing systems* (pp. 321–328).
- Zhou, L., Burgoon, J. K., Twitchell, D. P., Qin, T., & Nunamaker, J. F., Jr. (2004b). A comparison of classification methods for predicting deception in computer-mediated communication. *Journal of Management Information Systems*, 20(4), 139–166.
- Zhu, X., & Goldberg, A. B. (2009). Introduction to semi-supervised learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 3(1), 1–130.
- Zhu, X., Ghahramani, Z., & Lafferty, J. D. (2003). Semi-supervised learning using gaussian fields and harmonic functions. In *ICML 2003: International conference on machine learning* (pp. 912–919).

Authors and Affiliations

Mariana Caravanti de Souza¹  · **Bruno Magalhães Nogueira**² ·
Rafael Geraldeli Rossi³ · **Ricardo Marcondes Marcacini**¹ · **Bruce Neves dos Santos**¹ ·
Solange Oliveira Rezende¹

✉ Mariana Caravanti de Souza
mariana.caravanti@usp.br

Bruno Magalhães Nogueira
bruno@facom.ufms.br

Rafael Geraldeli Rossi
rafael.g.rossi@ufms.br

Ricardo Marcondes Marcacini
ricardo.marcacini@icmc.usp.br

Bruce Neves dos Santos
bruce.neves@usp.br

Solange Oliveira Rezende
solange@icmc.usp.br

¹ ICMC-USP, São Carlos 13566-590, Brazil

² FACOM-UFMS, Campo Grande 79070-900, Brazil

³ CPTL-UFMS, Três Lagoas 79613-000, Brazil