A Hybrid Approach for Fake News Detection in Twitter Based on User Features and Graph Embedding

Tarek Hamdi[1](✉), Hamda Slimi[1,2](✉), Ibrahim Bounhas[2](✉), and Yahya Slimani[2]

[1] National School for Computer Science, Manouba, Tunisia
{tarek.hamdi,hamda.slimi}@ensi-uma.tn
[2] INSAT, LISI Research Laboratory, University of Carthage, Tunis, Tunisia
bounhas.ibrahim@gmail.com, yahya.slimani@gmail.com

Abstract. The quest for trustworthy, reliable and efficient sources of information has been a struggle long before the era of internet. However, social media unleashed an abundance of information and neglected the establishment of competent gatekeepers that would ensure information credibility. That's why, great research efforts sought to remedy this shortcoming and propose approaches that would enable the detection of non-credible information as well as the identification of sources of fake news. In this paper, we propose an approach which permits to evaluate information sources in term of credibility in Twitter. Our approach relies on node2vec to extract features from twitter followers/followees graph. We also incorporate user features provided by Twitter. This hybrid approach considers both the characteristics of the user and his social graph. The results show that our approach consistently and significantly outperforms existent approaches limited to user features.

Keywords: Fake news source detection · User credibility · Twitter network analysis · Graph embedding · Machine learning

1 Introduction

According to a recent survey [44], in 2017 there is approximately 224.3 million smartphone users in the United States and more than 2 billion worldwide. Factors such as the affordable price of mobiles markets, the development of wireless communications technology, and the competitive prices of internet providers facilitated the creation of various Online Social Networks (OSNs). Also, microblogging is considered as a bridge for people to share, coordinate and spread information about events instantly. However, the ease of information sharing allowed the spread of malicious and incorrect content. The absence of gatekeepers led to a need for a credibility evaluation of content shared on social media.

For instance, Twitter has more than half a billion users worldwide with more than 300 million active users [22] who produce approximately 170 million tweets

D. V. Hung and M. D'Souza (Eds.): ICDCIT 2020, LNCS 11969, pp. 266–280, 2020.
https://doi.org/10.1007/978-3-030-36987-3_17

per day, about 10,000 tweets per second. This increase of inter-connectivity and the massive data can be source of interest to a wide range of organizations. This can put the users at the mercy of fake news, misinformation and rumors which have critical consequences [30] that would imperil their personal safety and privacy. Businesses, governmental agencies, and political parties [33] can take the valuable information in tweets for different purposes, including policy formation, decision-making, advertising campaigns, and gauging people's awareness of governmental programs [18]. With an analysis of users data collected from OSNs, it is possible to predict users' future behavior [4,20].

Based on their different interests, Source Of Fake News (SOFNs) continue to disturb and fallacy people opinions. In politics, during the 2016 U.S. presidential election, Grinberg and his team came to the conclusion that fake news was most concentrated among conservative voters [14]. In the same way, Allcott and Gentzkow mention that the 2016 election might have been affected by several rumors and fake news that became viral on Facebook and Twitter [5]. Allcott, Gentzkow and Econ mentioned that 27% of people visited a fake news source in the final weeks before the election [5].

In the cryptocurrency domain, Aggarwa with his group underline the relationship between social factors and the way the stock market of Bitcoin and cryptocurrency behaves. They draw attention to the major impact that social media has on cryptocurrency value [2]. Such statement clearly shows that, in some cases, fake news can dictate the outcome of a stock market.

In 2018, crypto-investors lost more than $9.1 million a day on average to cryptocurrency fraud based on a survey done by Bitcoin.com News [34]. Fake news can affect badly the market and lead to heavy losses. In 2013, a fake tweet about explosion in White House claimed that Barack Obama was injured in this explosion caused $130 billion loss on the stock market [29]. Fake news are present during disaster event and terrorism attacks. In the Boston Marathon bombing during the spring of 2013, terrorist and fake sources terrified people with lies and rumors. Some twitter profiles and bots are established entirely with the intent to spread fabricated, unfounded and misleading information [36].

Those faceless and anonymous people play a major role on what becomes viral on a social media platform during a certain event. Therefore as a remedy to such malicious behavior, some researchers proposed various approaches to detect sources of fake news in Twitter.

2 Related Work

Automatic SOFNs detection can be regarded as a binary classification problem, that can be solved using feature extraction and machine learning algorithms. Our main focus is Twitter, where user credibility can be analyzed from three levels: post or tweet level, topic or event level, and user level.

The topological follow relationship in Twitter yields a social graph which is asymmetric. The Twitter follow graph is directed as the user being followed does not necessarily follow his/her follower. Many follows are built on social ties, which means that Twitter is primarily about information consumption based on the “follow” relationship [27]. Twitter follow graph represent an information and a social network based on its structural characteristics [27].

Twitter is considered as a news medium as well as a social network. The directed nature of the links between users distinguishes Twitter from other OSNs such as Facebook which is based on undirected links between users [6]. Nevertheless, symmetric relationships can exist on Twitter; two users can follow each other. Based on these features, we perceive Twitter to be more about the manner in which information propagates in reality [6].

Some of the various approaches that were proposed to tackle the issue of the sources of fake news detection are discussed below. Mainly we can distinguish the following categories: user-based features, graph-based features, NLP features and influence-based features.

Some of the first attempts to detect the source of information spread where focused on the node proprieties, they used various graph-theoretic measures such as degree of centrality, betweenness, closeness and eigenvector centrality. They assume that the node with the highest values is likely to be the source [10].

Gupta and Lamba [16] focused on tweets sharing fake images during Hurricane Sandy. They used two levels: user and tweet features. In the first level, they collected the followers graph between users. Then, they analyzed the role of Twitter social graph in propagating the fake tweet images. In the second level, they focused on the retweet graph and they found that 86% of tweets spreading the fake images were retweets. Hence, very few were original tweets. They also extracted the sources of tweets from the intersection of the followers graph and the tweet graph, then they relied on their classification model to detect if the picture shared by the source was fake or not. Their classification model attained 97% accuracy in detecting fake images. Finally, authors highlight that Tweet-based features are very effective in discovering fake images. Whereas, the performance of user based features is mediocre. They also discovered that the top 30 users (0.3% of the people) resulted in 90% of retweets of the fake images. Therefore, sources of fake news usually represent a minority compared to credible sources. Hence, Aditi and Hemank conclude that a handful of users contributed to the majority of the damage, via the retweeting activity on Twitter.

Caninin’s group [9] focused their research on automatic ranking to measure the credibility of information resources on Twitter. For each given topic, tweet content and network structure acts as a prominent feature for the effective ranking of user credibility. Some research had focused on time snapshot of the infected nodes to construct a maximum likelihood estimator to identify a single source of rumor [35]. Canini et al., showed that by combining a basic text search with an analysis of the social structure of the network, it is possible to generate a ranked list of relevant users for any given topic. They found that the most important thing to identify relevant and credible users is the content-based topic analysis of the social network [9]. In the end, Canini et al. perceived that their algorithm could be more relevant by combining topic-based textual analyses with network-based information in order to identify interesting users to follow. This goal is only attainable if they have knowledge of the follower graph which is the biggest issue [6].

The ease of information sharing on a massive scale renders the detection of SOFNs a fastidious task, since Twitter permits almost anyone to publish their thoughts or share news about events they attended. Moreover, the scarcity and uniqueness of fake news encumbers the ability to learn a trend in data. Based on the aforementioned inconveniences, we believe it is crucial to appropriately deal with the atypical character of sources of fake news.

Other researchers used graph theory to assist in the detection of sources of fake news. In fact, Shu et al. [38] showed that network diffusion models can be applied to trace the provenance nodes and the provenance paths of fake news.

3 Methodology

We are motivated by the great success in applications of attention-based on graph machine learning. We consider extracting some useful features of users from Twitter social graph (followers/following) by using graph embedding which captures the irregular behavior of SOFNs. It takes into consideration the relationship between Twitter users. In this paper, we propose an efficient automated SOFNs detection model by combining user features and social features. As social feature contain important clues, our proposed model is expected to improve the performance of SOFNs detection.

In this section, we discuss our pipeline process and the data collection setup which consists in several different stages: (1) Searching for an appropriate twitter social graph to test our approach based on graph embedding. (2) For each user in the graph, we collect the topics he twitted about. (3) Calculate each user credibility based on topics credibility annotation. (4) We compare each user credibility to a fixed threshold in order to label each user as SOFNs or not. (5) Once the appropriate graph is selected, a graph embedding operation is applied in order to extract nodes feature. (6) We extract each user profile features from Twitter. (7) We apply feature selection to the resulted graph embedding vector. (8) We concatenate user profile features and user graph features. (9) We will try several binary machine learning models to classify correctly SOFNs. All those steps are clearly shown in Fig. 1.

Data Collection and Preprocessing. In general, most of the works related to SOFNs, rumors or fraud, are likely to face the problem of imbalanced data which renders the classification task tedious. Therefore, as a data source for our approach we refereed to published graph dataset, ego-Twitter[1] in SNAP[2] by McAuley and Leskovec [24]. Then, we tried to find a dataset which contains a list of Twitter users labeled as sources of fake news or source of reliable news. Moreover, to classify a user as a SOFNs or not, we need to evaluate their tweets and topic credibility. The collection of users tweets can be achieved by using

[1] https://snap.stanford.edu/data/ego-Twitter.html.

[2] snap-stanford/snap: Stanford Network Analysis Platform (SNAP) is a general purpose network analysis and graph mining library.

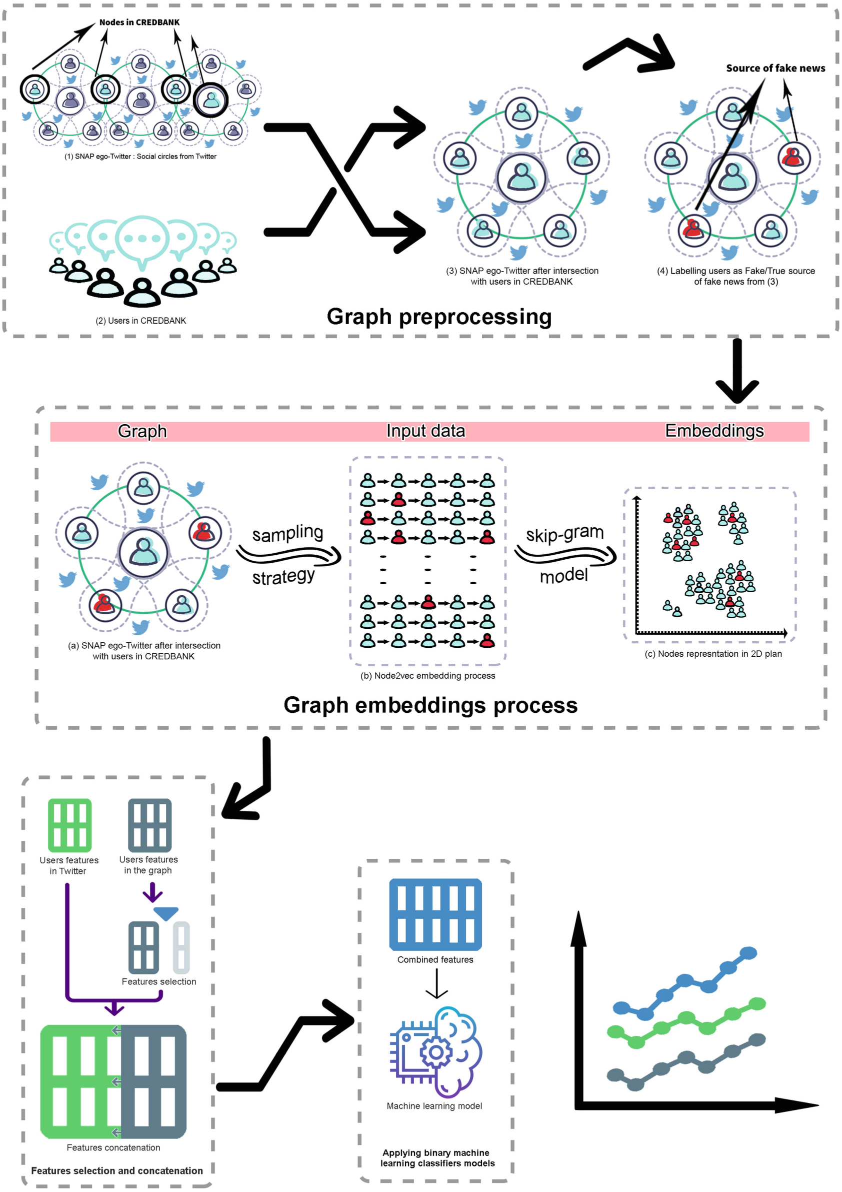

Nodes in CREDBANK
(1) SNAP ego-Twitter : Social circles from Twitter
(2) Users in CREDBANK
(3) SNAP ego-Twitter after intersection with users in CREDBANK
Source of fake news
(4) Labelling users as Fake/True source of fake news from (3)
Graph preprocessing
Graph
Input data
Embeddings
sampling strategy
skip-gram model
(a) SNAP ego-Twitter after intersection with users in CREDBANK
(b) Node2vec embedding process
(c) Nodes represntation in 2D plan
Graph embeddings process
Users features in Twitter
Users features in the graph
Features selection
Features concatenation
Features selection and concatenation
Combined features
Machine learning model
Applying binary machine learning classifiers models

Fig. 1. The pipeline process.

Twitter API. Manually determining the veracity of tweets about news is a challenging task since only experts in the field are eligible for news annotation. These annotators perform careful analysis of claims and additional evidence, context, and reports from authoritative sources [39] to achieve a credible annotation of news content. However, there are no agreed upon benchmark datasets for the fake news detection problem, there is no way of determining whether or not news is fake or not [39], and if there was, it would not be a problem in the first place. Fortunately, in 2015, during the ninth International AAAI Conference on Web and Social Media (ICWSM), a Large-scale Social Media Corpus with Associated Credibility Annotations of approximately 60 million tweets that covers 96 days starting from October 2015 named CREDBANK[3] was announced by Mitra and Gilbert [26], so we used CREDBANK to determinate which user is SOFNs or not in the Labelling Users subsection.

3.1 Graph Selection and Preprocessing

We established an intersection between nodes in ego-Twitter network and the users in CREDBANK to retrieve the graph of users like shown in the graph preprocessing step in the Fig. 1. Also, we selected from ego-Twitter network only the edges which their node source and destination are in CREDBANK. However, an issue was accoutered during the establishment of the intersection, users in CREDBANK are identified with their screen names (string) on Twitter but users in ego-Twitter dataset are identified with their ID (integer). In order to handle this issue, we used Twitter API through Tweepy[4] library to extract each users screen name in ego-Twitter dataset to remedy the issue. The resulted graph contains the relationships between 11592 unique users in CREDBANK and 63108 edges. Then, we created this graph using NetworkX[5], a Python package for the creation, manipulation, and study of the structure, dynamics, and functions of complex networks. Figure 2 presents this graph using Gephi[6]: a software for network analysis and visualization. The zoomed graph is shown in the right part of the Fig. 2. Some nodes and edges can be seen clearly as white circles and white lines, the size of each node increases depending on his degree. This combination yielded a graph that will be used along with the users features to train credibility classification model. In the following, we will refer to the graph as CREDGRAPH.

3.2 Graph Embeddings

The central problem in machine learning on graphs is finding a way to incorporate information about the structure of the graph into the model [17]. Furthermore, graph data is represented in a non-linear manner and it is non-trivial to identify a "context" for any single node in the graph [37].

We start by defining an unweighted graph by synthesizing the two definitions of Wang et al. [43] and Abu-El-Haija et al. [1]. A graph G(V, E) is a collection of

[3] http://compsocial.github.io/CREDBANK-data/.

[4] Tweepy is open-sourced, hosted on GitHub and enables Python to communicate with Twitter platform and use its API.

[5] https://networkx.github.io/.

[6] https://gephi.org/.

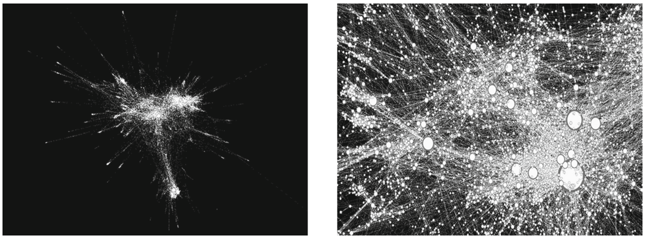


Fig. 2. The graph of credibility 'CREDGRAPH'

$V = \{v_1, \cdots, v_n\}$ vertices (a.k.a. nodes) and $E = \{e_{ij}\}_{i,j=1}^{n}$ edges. The adjacency matrix $\mathbf{A} \in \{0,1\}^{|V|\times|V|}$ of unweighted graph G can be constructed according to $A_{vu} = \mathbb{1}[(v,u) \in E]$, where the indicator function $\mathbb{1}[.]$ evaluates the values in the matrix to 1 if its boolean argument is true.

To solve graph-based problems using a model, there are two choices available. Either, operations may be directly applied on the original adjacency matrix or on a derived vector space. Working directly on the original adjacency matrix is time consuming and requires substantial memory space. Recently, the idea of representing a network in a vector space without losing their properties by automatically learning to encode graph structure into low-dimensional embedding, using techniques based on deep learning, Factorization, Random Walk and nonlinear dimensionality reduction have become widely popular [13,17,37]. In general, graph embedding methods minimize an objective:

$$\min_{\mathbf{Y}} \mathcal{L}(f(\mathbf{A}), g(\mathbf{Y}));$$

where $\mathbf{Y} \in \mathbb{R}^{|V|\times d}$ is a d-dimensional node embedding dictionary; $f : \mathbb{R}^{|V|\times|V|} \to \mathbb{R}^{|V|\times|V|}$ is a transformation of the adjacency matrix; $g : \mathbb{R}^{|V|\times d} \to \mathbb{R}^{|V|\times|V|}$ is a pairwise edge function; and $\mathcal{L} : \mathbb{R}^{|V|\times|V|} \times \mathbb{R}^{|V|\times|V|} \to \mathbb{R}$ is a loss function.

The idea behind these approaches is to learn a mapping that embeds nodes, as points in a low-dimensional vector space of features $\mathbb{R}^d$ [17]. Also, it maximizes the likelihood of preserving network neighborhoods of nodes [15] and translates the relationships between nodes from graph space to embedding space [37]. The goal is to optimize this mapping so that geometric relationships in this learned space reflect the structure of the original graph [17].

Our task is considered as a user classification, where analogy is applied with node embedding. Hence, we are on a node classification problem which is the most common benchmark task.

Some algorithms are able to generate node embeddings, such as DeepWalk [32], Line [41], SDNE [43], HOPE [28], Graph Factorization (GF) [3] and Laplacian Eigenmaps [7]. DeepWalk is a random walk based algorithm with a fixed return parameter and in-out parameter of 1.

The node2vec model extends DeepWalk by providing a more flexible way to train different combinations of these parameters. Grover and Leskovec [15], define node2vec as an semi-supervised algorithmic framework for scalable feature and for representational learning in networks [15,37]. Therefore, the sampling strategy provided by DeepWalk can be considered as a special case of node2vec. Line learn graph embedding with an efficient way by leveraging both first-order and second-order proximities [37]. However, there is a difficulty in learning feature representation with a balance between BFS and DFS in a graph network. According to a graph embedding survey paper by Cai [8], node2vec is the only work that takes into account both BFS and DFS in a biased random walk algorithm and leads to the learning of an enriched graph embedding [37]. Goyal and Ferrara [13] tested in their survey the performance of several graph embedding methods such as node2vec, GF, SDNE, HOPE and LE on the task of node classification with several datasets. Their results in Figs. 3 and 4 show that node2vec outperforms other methods in this task.

Indeed, node2vec is motivated and inspired by prior work on natural language processing. More specifically, the word2vec [25,37] model. There is an analogy between node2vec and word2vec as each node in the graph behaves as a single word in text. Also, a group of neighborhood nodes around a specific node are similar to the context around a word.

In general, as introduced by Shen et al. [37], these methods operate in two discrete steps. First, they prepare input data through a random walk based sampling strategy. Specifically, for graph data, node2vec takes two forms of equivalences into consideration: homophily [42] and structural equivalence [12,45]. For homophily equivalence, node2vec uses a breadth first search (BFS) algorithm to reach neighborhood nodes based on homogeneous clusters. For structural equivalence, node2vec leverages a depth first search (DFS) strategy to identify neighborhood nodes based on their structural roles (e.g., hub node or peripheral node). Node2vec makes a mixture of both equivalences and customizes a random walk algorithm to switch between BFS and DFS in a moderate way, in order to balance graph searching. Second, once the input data is ready, node2vec

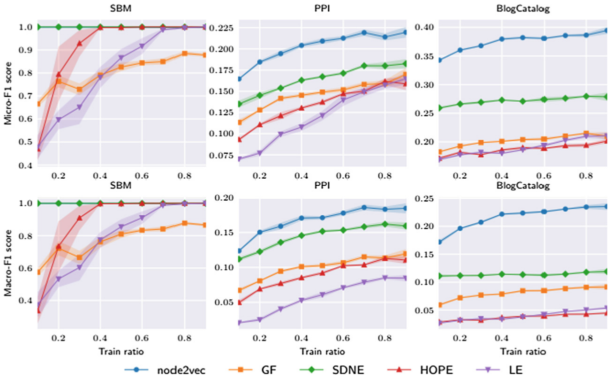
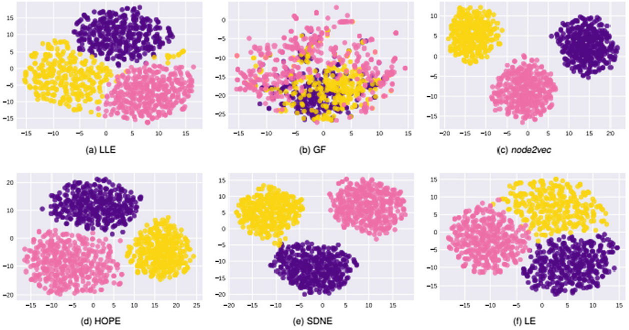


Fig. 3. Micro-F1 and Macro-F1 of node classification for different data sets varying the train-test split ratio (dimension of embedding is 128) [13]



(a) LLE (b) GF (c) *node2vec*

(d) HOPE (e) SDNE (f) LE

Fig. 4. Visualization of SBM using t-SNE (original dimension of embedding is 128). Each point corresponds to a node in the graph. Color of a node denotes its community [13] (Color figure online)

trains an embedding model using Skip-gram to learn representations that encode pairwise node similarities and generate node embeddings [1].

The hyperparameters used by node2vec are: embedding dimensions d, walks per node r, walk length l, context size k, return p, in-out q.

Therefore, we chose node2vec in this study for generating node embedding along with the hyper-parameters: $d = 128$, $r = 200$, $l = 5$, $p = 0.5$ and $q = 3$ applied to our CREDGRAPH. We tried the 256, 64, 32 and 16 but the 128 dimension attained the best results with the classifiers.

3.3 Users Labelling

The steps of this stage are: (1) Collecting from CREDBANK dataset all the tweets posted by the 11592 users. (2) Joining those results with their topics terms (an identifier of 3 words for each topic in CREDBANK) as a key. (3) For each tweet, we calculate the average of the 30 ratings given by the AMT[7] to the topic terms. There is over a 100 thousand tweets posted by the selected users. (4) Retrieving the number of tweets posted by each user. Furthermore, we considered users having reached more than 5% of their tweets with an average value less than one as SOFNs and we attribute them the value 1 as a label. The rest are considered as SORNs with the label 0. We chose 5% as a threshold to decide if the user is SOFNs or not because we discovered that we are in an imbalanced dataset with a very low tweets that contain fake news. In other terms, most of the news posted by the selected users are credible. So, it was hard to find a number of users which have more than 50% or 25% of their tweets less than one in average rating. This can guide us for an other imbalance in the level of user classes. By applying the 5% as threshold, we identified less than 500 users as SOFNs and more than 11000 users as SORNs.

[7] https://www.mturk.com/.

3.4 User Feature Extraction

Analyzing user-based features can provide useful information for SOFNs detection. Such features represent the profile characteristics of users who have interacted with news on social media. We extracted these features for the selected set of users from their corresponding accounts in Twitter. We referred to Twitter API through Tweepy to acquire the user-based features. Unfortunately, we were able to extract only 5536 accounts information from the 11592 CREDBANK users because some accounts were blocked or private. Finally, 468 users were deemed as SOFNs and 5068 users as SORNs after following the instructions in Subsect. 3.3. The extracted features are: followers, friends, created_at, name, default_profile, default_profile_image, favourites_count, statuses_count, description.

Once the user-based features are extracted, a standard feature engineering techniques was applied. It consist of the following steps (1) generate elaborate features based on the basic features extracted from Twitter; (2) normalization; and, (3) feature combination. These steps produced the following features: month and year extracted from created_at, Same_name_scrname which describes if the user handle is the same as user screen name, length_user_id: the number of digits in the user ID, length_description: presents the number of characters in the description, length_description_words: the number of words in the description and number of tweets tweeted by each user in the CREDBANK dataset.

We did not focus on the engineering process for user features since it is irrelevant to our study case. Also, user-based features are only considered as baseline for our approach. Finally, the main interest of our contribution is to show the impact of node embedding in the detection of users who spread fake news by combining it with the user-based features.

3.5 Node2vec Features Selection

One of the biggest challenges in node embedding is to find the optimal dimensions of the vector representation of nodes in the graph. The reconstruction precision of a graph depends on the number of dimensions. However, a high number of dimensions will increase the training time and decrease the precision in node classification.

First, an examination of the vector representation of nodes is performed in order to determine the importance of each dimension or feature in the vector. Then, features that reduce the vector dimensions and increase the performance of node classification are kept. We used Light GBM [21] model to evaluate the importance of each feature. It approaches the problem in two ways. The first one is "split" [21] which represents the number of times a feature is used in a model. The second one is "gain" [21] which the total gains of splits that use the feature.

The vector contains 128 dimensions and each dimension represents a feature. We relied on the gain value to select prominent features since the number of dimensions is fairly small. The process is: (1) Sort the 128 features that resulted from node2vec based on their gain importance. (2) Drop all features with zero gain value. (3) Drop features starting from the lowest gain value until the best dimensions size that yields the highest results in node classification is achieved. (4) Concatenate the selected node embedding features with user-based features. These steps are resulted in the Fig. 1.

4 Results and Discussion

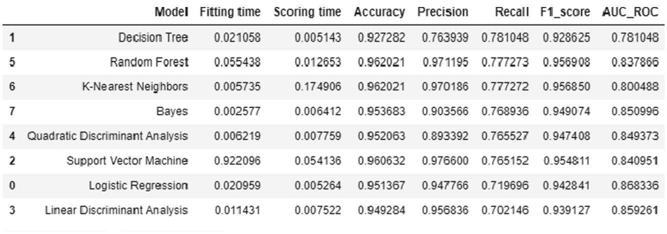
To evaluate the performance of our approach, various metrics were used. In this section, we review the most widely used metrics for SOFNs detection. The choice of the appropriate evaluation metric plays a critical role in achieving an unbiased judgment of classifier performance. Also, a selection of the suitable evaluation metric is an important key for choosing the optimal classifier [19].

The classification problem was defined as follows: can we detect the sources of fake news on Twitter through a combination of users account features and features from the user social graph? Fitting and scoring time are used to show the duration of both the learning and the prediction process respectively. Whereas, the remaining measures showcase the performance and the ability of the proposed model to detect in an efficient and non-biased manner the sources of fake news.

Precision measures the fraction of all detected users as SOFNs that are annotated as SOFNs. It considers the important aspect of identifying which user is SOFNs. However, SOFNs datasets are often skewed, a high precision can be easily achieved by making fewer positive predictions [39]. Thus, recall is used to measure the sensitivity, or the fraction of annotated fake news sources that are predicted to be SOFNs. F1_score is used to combine precision and recall, which can provide an overall prediction performance for SOFNs detection. We note that for Precision, Recall, F1, and Accuracy, the higher the value, the better the performance. Like recall, the specificity metric should be maximized, although recall has a higher impact because falsely predicting that source of real news as SOFNs will not affect the trust placed in the sources predicted to be fake. The maximum value for the AUC is 1.0, which indicates a perfect test since it is 100% sensitive and 100% specific. An AUC value of 0.5 indicates there is no class separation capacity at all [11]. As our concatenated features are represented in a vector space, dimensionality reduction techniques like Principal Component Analysis (PCA) [31], t-distributed stochastic neighbor embedding (t-SNE) [23] and Truncated SVD [40] can be applied to visualize the disparity between different group of users (Fig. 6).

With our approach, SOFNs detection achieved optimal AUC-ROC and Recall equal to 0.99–0.99 with Linear Discriminant Analysis (LDA) algorithm which was equal to 0.79–0.79 respectively.

Based on the metrics shown in the Fig. 7, we can clearly notice that LDA outperforms other classifiers in Recall, f measure and AUC-ROC. By comparing results in Figs. 5 and 7, an improvement of classifier performance can be noticed. Amongst the derived subsets of features, we can clearly notice that the combination of Node2Vec and user features achieved the best results. In fact, LDA



	Model	Fitting time	Scoring time	Accuracy	Precision	Recall	F1_score	AUC_ROC
1	Decision Tree	0.021058	0.005143	0.927282	0.763939	0.781048	0.928625	0.781048
5	Random Forest	0.055438	0.012653	0.962021	0.971195	0.777273	0.956908	0.837866
6	K-Nearest Neighbors	0.005735	0.174906	0.962021	0.970186	0.777272	0.956850	0.800488
7	Bayes	0.002577	0.006412	0.953683	0.903566	0.768936	0.949074	0.850996
4	Quadratic Discriminant Analysis	0.006219	0.007759	0.952063	0.893392	0.765527	0.947408	0.849373
2	Support Vector Machine	0.922096	0.054136	0.960632	0.976600	0.765152	0.954811	0.840951
0	Logistic Regression	0.020959	0.005264	0.951367	0.947766	0.719696	0.942841	0.868336
3	Linear Discriminant Analysis	0.011431	0.007522	0.949284	0.956836	0.702146	0.939127	0.859261

Fig. 5. The classification results with only user-based features

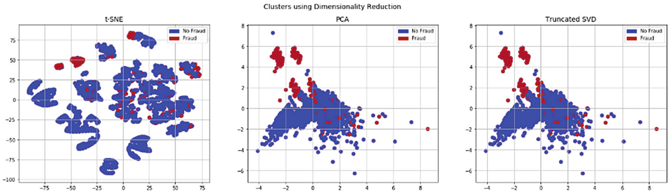
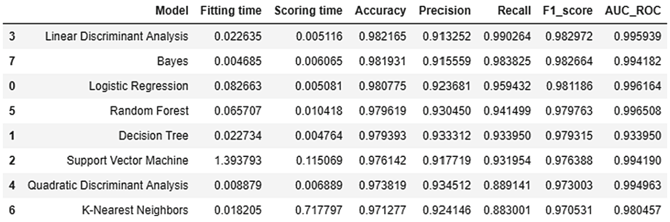


Fig. 6. Visualization of users with only user-based features using t-SNE, PCA and truncated SVD. Each point corresponds to a user. Color of a node denotes its class (Color figure online)



	Model	Fitting time	Scoring time	Accuracy	Precision	Recall	F1_score	AUC_ROC
3	Linear Discriminant Analysis	0.022635	0.005116	0.982165	0.913252	0.990264	0.982972	0.995939
7	Bayes	0.004685	0.006065	0.981931	0.915559	0.983825	0.982664	0.994182
0	Logistic Regression	0.082663	0.005081	0.980775	0.923681	0.959432	0.981186	0.996164
5	Random Forest	0.065707	0.010418	0.979619	0.930450	0.941499	0.979763	0.996508
1	Decision Tree	0.022734	0.004764	0.979393	0.933312	0.933950	0.979315	0.933950
2	Support Vector Machine	1.393793	0.115069	0.976142	0.917719	0.931954	0.976388	0.994190
4	Quadratic Discriminant Analysis	0.008879	0.006889	0.973819	0.934512	0.889141	0.973003	0.994963
6	K-Nearest Neighbors	0.018205	0.717797	0.971277	0.924146	0.883001	0.970531	0.980457

Fig. 7. The classification results using node embeddings with 30 features and user-based features

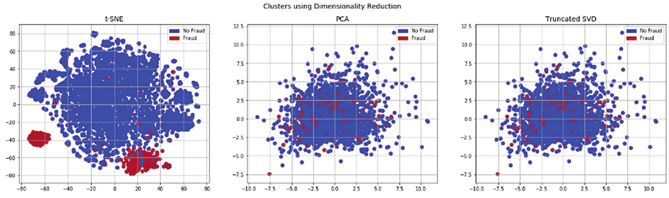


Fig. 8. Visualization of users using t-SNE, PCA and truncated SVD using node embeddings with 30 features and user-based features. Each point corresponds to a user. Color of a node denotes its community (Color figure online)

achieved the best results with a Recall equal to 0.98, an F1_score equal to 0.98 and AUC-ROC equal to 0.98. This can be explained by the fact that node2vec features encompasses information about user credibility that mere Twitter user features are unable to convey (Fig. 8).

5 Conclusion

In this paper, we proposed a hybrid approach which combines node embeddings and user-based features to enrich the detection of sources of fake news in Twitter. Our study showed that the extracted knowledge from the social network graph using node2vec is able to provide a generalized way to assist in detecting SOFNs. After the evaluation of classifiers performance, we conclude that features generated by graph embedding are efficient for the detection of SOFNs. Also, node embeddings features are powerful predictors of SOFNs that convey useful information about the credibility of a user in the social graph. We showed a significant improvements on SOFNs detection over state-of-the-art baseline which mainly relies on user-based features.

In future works, we intend to apply the aforementioned features in other domains related to source evaluation such as fraud, spam, rumor, and anomaly detection. Also, we will try to generate synthetic networks with real-world characteristics in order to study their evolution. Finally, link prediction and recommendation are a promising research tracks in the field of graph embedding.

References

1. Abu-El-Haija, S., Perozzi, B., Al-Rfou, R., Alemi, A.A.: Watch your step: learning node embeddings via graph attention. In: Advances in Neural Information Processing Systems, pp. 9180–9190 (2017). abs/1710.09599
2. Aggarwal, G., Patel, V., Varshney, G., Oostman, K.: Understanding the social factors affecting the cryptocurrency market (2019). arXiv preprint arXiv:1901.06245
3. Ahmed, A., Shervashidze, N., Narayanamurthy, S., Josifovski, V., Smola, A.J.: Distributed large-scale natural graph factorization. In: Proceedings of the 22nd International Conference on World Wide Web, pp. 37–48. ACM (2013)
4. Al-Qurishi, M., Al-Rakhami, M., Alrubaian, M., Alarifi, A., Rahman, S.M.M., Alamri, A.: Selecting the best open source tools for collecting and visualzing social media content. In: 2015 2nd World Symposium on Web Applications and Networking (WSWAN), pp. 1–6. IEEE (2015)
5. Allcott, H., Gentzkow, M.: Social media and fake news in the 2016 election. J. Econ. Perspect. **31**(2), 211–36 (2017)
6. Alrubaian, M., Al-Qurishi, M., Alamri, A., Al-Rakhami, M., Hassan, M.M., Fortino, G.: Credibility in online social networks: a survey. IEEE Access **7**, 2828–2855 (2018)
7. Belkin, M., Niyogi, P.: Laplacian eigenmaps and spectral techniques for embedding and clustering. In: Advances in Neural Information Processing Systems, pp. 585–591 (2002)

8. Cai, H., Zheng, V.W., Chang, K.C.C.: A comprehensive survey of graph embedding: problems, techniques, and applications. IEEE Trans. Knowl. Data Eng. **30**(9), 1616–1637 (2018)
9. Canini, K.R., Suh, B., Pirolli, P.L.: Finding credible information sources in social networks based on content and social structure. In: 2011 IEEE Third International Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third International Conference on Social Computing, pp. 1–8. IEEE (2011)
10. Comin, C.H., da Fontoura Costa, L.: Identifying the starting point of a spreading process in complex networks. Phys. Rev. E **84**(5), 056105 (2011)
11. Fan, J., Upadhye, S., Worster, A.: Understanding receiver operating characteristic (ROC) curves. Can. J. Emerg. Med. **8**(1), 19–20 (2006)
12. Fortunato, S.: Community detection in graphs. Phys. Rep. **486**(3–5), 75–174 (2010)
13. Goyal, P., Ferrara, E.: Graph embedding techniques, applications, and performance: a survey. Knowl. Based Syst. **151**, 78–94 (2018)
14. Grinberg, N., Joseph, K., Friedland, L., Swire-Thompson, B., Lazer, D.: Fake news on twitter during the 2016 US presidential election. Science **363**(6425), 374–378 (2019)
15. Grover, A., Leskovec, J.: Node2vec: scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2016, pp. 855–864. ACM (2016). https://doi.org/10.1145/2939672.2939754
16. Gupta, A., Lamba, H., Kumaraguru, P., Joshi, A.: Faking sandy: characterizing and identifying fake images on twitter during hurricane sandy. In: Proceedings of the 22nd International Conference on World Wide Web, pp. 729–736. ACM (2013)
17. Hamilton, W.L., Ying, R., Leskovec, J.: Representation learning on graphs: methods and applications (2017). arXiv preprint arXiv:1709.05584
18. Hassan, N.Y., Gomaa, W.H., Khoriba, G.A., Haggag, M.H.: Supervised learning approach for twitter credibility detection. In: 2018 13th International Conference on Computer Engineering and Systems (ICCES), pp. 196–201. IEEE (2018)
19. Hossin, M., Sulaiman, M.: A review on evaluation metrics for data classification evaluations. Int. J. Data Min. Knowl. Manag. Process **5**(2), 1 (2015)
20. Jin, L., Chen, Y., Wang, T., Hui, P., Vasilakos, A.V.: Understanding user behavior in online social networks: a survey. IEEE Commun. Mag. **51**(9), 144–150 (2013)
21. Ke, G., et al.: LightGBM: a highly efficient gradient boosting decision tree. In: Advances in Neural Information Processing Systems, pp. 3146–3154 (2017)
22. Kim, J., Hastak, M.: Social network analysis: characteristics of online social networks after a disaster. Int. J. Inf. Manag. **38**(1), 86–96 (2018)
23. Maaten, L.V.D., Hinton, G.: Visualizing data using t-SNE. J. Mach. Learn. Res. **9**(Nov), 2579–2605 (2008)
24. Mcauley, J., Leskovec, J.: Discovering social circles in ego networks. ACM Trans. Knowl. Discov. Data **8**(1), 4:1–4:28 (2014). https://doi.org/10.1145/2556612
25. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space (2013). arXiv preprint arXiv:1301.3781
26. Mitra, T., Gilbert, E.: CREDBANK: a large-scale social media corpus with associated credibility annotations. In: Ninth International AAAI Conference on Web and Social Media (2015)
27. Myers, S.A., Sharma, A., Gupta, P., Lin, J.: Information network or social network? The structure of the twitter follow graph. In: Proceedings of the 23rd International Conference on World Wide Web, pp. 493–498. ACM (2014)

28. Ou, M., Cui, P., Pei, J., Zhang, Z., Zhu, W.: Asymmetric transitivity preserving graph embedding. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1105–1114. ACM (2016)
29. Paluch, R., Lu, X., Suchecki, K., Szymański, B.K., Hołyst, J.A.: Fast and accurate detection of spread source in large complex networks. Sci. Rep. **8**(1), 2508 (2018)
30. Pastor-Satorras, R., Vespignani, A.: Epidemic spreading in scale-free networks. Phys. Rev. Lett. **86**(14), 3200 (2001)
31. Pearson, K.: LIII. On lines and planes of closest fit to systems of points in space. Lond. Edinb. Dublin Philos. Mag. J. Sci. **2**(11), 559–572 (1901)
32. Perozzi, B., Al-Rfou, R., Skiena, S.: Deepwalk: online learning of social representations. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2014, pp. 701–710 (2014). https://doi.org/10.1145/2623330.2623732
33. Sáez-Mateu, F.: Democracy, screens, identity, and social networks: the case of Donald Trump's election. Am. Behav. Sci. **62**(3), 320–334 (2018)
34. Seth, S.: $9 million lost each day in cryptocurrency scams. Investopedia 13 (2018)
35. Shah, D., Zaman, T.: Rumor centrality: a universal source detector. In: Proceedings of the 12th ACM SIGMETRICS/PERFORMANCE Joint International Conference on Measurement and Modeling of Computer Systems, SIGMETRICS 2012, pp. 199–210 (2012). https://doi.org/10.1145/2254756.2254782
36. Shao, C., Ciampaglia, G.L., Varol, O., Flammini, A., Menczer, F.: The spread of fake news by social bots, pp. 96–104 (2017). arXiv preprint arXiv:1707.07592
37. Shen, F., et al.: HPO2Vec+: leveraging heterogeneous knowledge resources to enrich node embeddings for the human phenotype ontology. J. Biomed. Inform. **96**, 103246 (2019). https://doi.org/10.1016/j.jbi.2019.103246
38. Shu, K., Bernard, H.R., Liu, H.: Studying fake news via network analysis: detection and mitigation. In: Agarwal, N., Dokoohaki, N., Tokdemir, S. (eds.) Emerging Research Challenges and Opportunities in Computational Social Network Analysis and Mining. LNSN, pp. 43–65. Springer, Cham (2019). https://doi.org/10.1007/978-3-319-94105-9_3
39. Shu, K., Sliva, A., Wang, S., Tang, J., Liu, H.: Fake news detection on social media: a data mining perspective. ACM SIGKDD Explor. Newslett. **19**(1), 22–36 (2017)
40. Speer, R., Havasi, C., Lieberman, H.: Analogyspace: reducing the dimensionality of common sense knowledge. In: Proceedings of the 23rd National Conference on Artificial Intelligence, AAAI 2008, pp. 548–553 (2008)
41. Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., Mei, Q.: Line: large-scale information network embedding. In: Proceedings of the 24th International Conference on World Wide Web, WWW 2015, pp. 1067–1077 (2015). https://doi.org/10.1145/2736277.2741093
42. Tang, L., Liu, H.: Leveraging social media networks for classification. Data Min. Knowl. Disc. **23**(3), 447–478 (2011)
43. Wang, D., Cui, P., Zhu, W.: Structural deep network embedding. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1225–1234. ACM (2016)
44. Wu, L., Zhang, Y., Xie, Y., Alelaiw, A., Shen, J.: An efficient and secure identity-based authentication and key agreement protocol with user anonymity for mobile devices. Wirel. Pers. Commun. **94**(4), 3371–3387 (2017). https://doi.org/10.1007/s11277-016-3781-z
45. Yang, J., Leskovec, J.: Overlapping communities explain core-periphery organization of networks. Proc. IEEE **102**(12), 1892–1902 (2014)