Detection and Analysis of Fake News Users' Communities in Social Media

Abdelouahab Amira, Abdelouahid Derhab, Samir Hadjar, Mustapha Merazka, Md. Golam Rabiul Alam, *Member, IEEE*, and Mohammad Mehedi Hassan, *Senior Member, IEEE*

***Abstract*— The widespread use of social media platforms has led to an increase in the dissemination of fake news with the intention of manipulating public opinion and causing chaos and panic among the population. To address this issue, we focus on detecting the organized groups that participate together in fake news campaigns without prior knowledge of the news content or the profiles of social accounts. To this end, we propose a *spatial–temporal similarity graph*, a novel graph structure that connects social accounts that participate in the early stage of similar fake news campaigns. A community detection algorithm is applied on the similarity graph to cluster the users into communities. We propose a *community labeling algorithm* to label the communities as benign or malicious based on the output of a fake news classifier. Evaluation results show that the community labeling algorithm can correctly label the communities with an accuracy of 99.61%. In addition, we perform a statistical comparison analysis to identify the structural community features that are statistically significant between benign and malicious communities.**

***Index Terms*— Community detection, community labeling, fake news, similarity graph.**

I. Introduction

SOCIAL media are playing an increasingly significant role in today's society. People use these platforms to report events and share news on a large scale. It has become commonplace for news to reach people through social media before being broadcast by traditional media such as TV, radio, and newspapers.

In general, users do not check the accuracy of news and content they share on social media, resulting in the wide dissemination of fake news. Fake news can take many forms, including clickbait, disinformation, misinformation, hoax, parody, satire, rumor, and deceptive news [1], [2], [3]. Fake news can have negative effects on users and society. For instance, fake news can manipulate public opinion and be exploited by states to negatively influence the political decisions of a population and destabilize society.

Different approaches have been proposed in the literature to deal with fake news [1]. The first approach focuses on analyzing the content of news to determine whether it is fake or not. The second approach aims at detecting social bot accounts that are programmed to launch different attacks, including spreading fake content. The third approach focuses on detecting fake news campaigns that aim to manipulate public opinion in an organized way. To this end, the propagation patterns of news are analyzed, as fake and real news propagate differently.

Although detecting fake news campaigns has received attention in the literature [4], [5], [6], it cannot identify the source of the threat, i.e., the human and social bot accounts that intentionally participated in the campaigns. People could participate in the fake news campaign without bad intentions because they simply disseminated every news they received or wanted to express an opinion regarding the news.

To address the above limitation, we focus in this article on identifying the threat actors behind the news campaigns, i.e., groups of social accounts that spread the fake news in an organized manner. To this end, we rely on the assumption that these organized groups actively participate together in many news campaigns. In addition, the organized groups are considered early birds as they usually start the news campaigns or join them in early stages [7].

In this article, we propose a novel approach for detecting organized social groups that participate in fake news campaigns without prior knowledge of the news content or user profiles. We use the assumption that these groups actively participate together in many news campaigns. We cluster these groups using a community detection algorithm based on their spatial–temporal correlated activities in the fake news campaigns. Our main contributions are as follows.

1) We propose a novel graph structure called the *spatial–temporal similarity graph*, which connects social accounts that participate in the early stage of similar fake news campaigns.
2) Based on this similarity graph, we apply a state-of-the-art weighted community detection algorithm, named the label propagation algorithm (LPA), to identify the

Manuscript received 31 December 2022; revised 1 April 2023; accepted 24 May 2023. Date of publication 15 June 2023; date of current version 2 August 2024. This work was supported by the Deanship of Scientific Research, King Saud University through the Vice Deanship of Scientific Research Chairs, Chair of Pervasive and Mobile Computing. *(Corresponding author: Abdelouahid Derhab.)*

Abdelouahab Amira, Samir Hadjar, and Mustapha Merazka are with the Research Center for Scientific and Technical Information (CERIST), Algiers 16000, Algeria, and also with the Departement Informatique, Faculte des Sciences Exactes, Universite de Bejaia, Bejaia 06000, Algeria (e-mail: amira@cerist.dz; hadjar@cerist.dz; mmerazka@cerist.dz).

Abdelouahid Derhab is with the Center of Excellence in Information Assurance (CoEIA), King Saud University, Riyadh 11451, Saudi Arabia (e-mail: abderhab@ksu.edu.sa).

Md. Golam Rabiul Alam is with the Department of Computer Science and Engineering, BRAC University, Dhaka 1212, Bangladesh (e-mail: rabiul.alam@bracu.ac.bd).

Mohammad Mehedi Hassan is with the Department of Information Systems and Research Chair of Pervasive and Mobile Computing, College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia (e-mail: mmhassan@ksu.edu.sa).

Digital Object Identifier 10.1109/TCSS.2023.3282572

2329-924X © 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: University of Edinburgh. Downloaded on November 04,2024 at 09:49:04 UTC from IEEE Xplore. Restrictions apply.

different communities of users who participate in the early stage of similar fake news campaigns.

3) We propose a community labeling algorithm that uses the class of news campaigns, i.e., real or fake, to label the constructed communities as benign or malicious. The community labeling algorithm can correctly identify the class of communities with an accuracy of 99.61%.
4) Using structural community features, we perform a statistical comparison analysis between the obtained benign and malicious communities. Through this analysis, we identify the features that are statistically significant between benign and malicious communities.

The rest of the article is organized as follows: In Section II, we give a background on community detection algorithms. Section III presents related work. The methodology of the proposed approach is described in Section IV. In Section V, we describe the implementation of the approach. Section VI presents the evaluation results. Finally, Section VII concludes the article.

II. Background

Social network analysis is the process of gathering data from a social network and using mathematical, statistical, and artificial intelligence techniques to analyze them. It is used to understand the connections between people, identify patterns of behavior, and uncover influential users. However, the analysis of such networks is challenging due to their large scale and dynamic nature. Community detection is a valuable tool that can help identifying clusters or communities within a social network, providing insights into the structure and function of the whole network [8], [9], [10], [11], [12]. This can be useful for a variety of applications, including marketing, content recommendation, and prediction of users behaviors. The most popular four common unsupervised algorithms for community detection are the following.

1) *The Louvain Algorithm [13]:* It seeks to optimize a quality function and operates in two steps. In the first step, nodes are moved to a selected community in a way that gives the best increase in the quality function. In the second step, communities are represented as individual nodes, and the process is repeated until the quality of the function can no longer be increased.
2) *Infomap [14]:* It uses the code length of a random walk of the map as the objective to be optimized, in order to detect the network structure. Infomap is based on the map equation, a method that models a flow as random walks through the graphs and a graph partitioning is scored by finding a compressed modular representation of this flow.
3) *LPA [15]:* It is a fast community detection algorithm that uses the network's structure as a guide. Each node is assigned a specific label. Afterward, an iterative process of label propagation is applied, whereby a node's label is set as the most common label among its neighbors. The process stops when no labels are updated for all the nodes. LPA can also apply semi-supervised learning by initially assigning some nodes to communities.
4) *The Girvan–Newman Algorithm [16]:* It is based on the idea that nodes are strongly connected within the same community and loosely connected to nodes from other communities. To identify these groups, the algorithm removes edges with the most number of shortest paths between nodes. This process breaks down the network into distinct communities.

III. Related Work

In this section, we present the different analysis detection tasks related to fake news, including fake news detection, fake account detection, and fake news campaigns detection.

A. Fake News Detection

Fake news detection [17] aims at classifying the news content as real or fake. The detection could be done manually through experts and analysts, or through automatic techniques, such as natural language processing (NLP), machine learning, and information retrieval [18]. The fake news detection systems can be categorized as: content-based, social-context-based, and knowledge-based approaches [19].

1) Content-Based Approaches: As fake news exhibit a distinguishable writing style from the real ones [20], the content-based approaches consider the linguistic features, which can be extracted from the news [21], [22], [23], [24], [25]. Artificial intelligence techniques, which have shown success in NLP, have been leveraged for fake news detection [26].

Although content-based approaches have shown success in detecting fake news, they have some limitations. For instance, as fake news styles and topics keep changing, the models that are trained on a specific dataset and their features cannot generalize well to different contents, writing styles, and languages and will perform poorly on new datasets. It has been shown in [27] and [28] that content-based approaches are not sufficient to detect fake news, as they rely on changing data and high data veracity [19].

2) Knowledge-Based Approaches: They use external sources to check the truthfulness of the news content [29], [30], [31]. To this end, they use manual and automated fact-checking. In manual fact-checking, regular or expert individuals, called fact-checkers, provide their judgments on the news [1]. However, manual fact-checking is a cumbersome task, as it requires manual and continuous update. In automatic fact-checking, the user's claims, which are extracted from different sources along with the reputations of sources, are fed to a classification algorithm, to decide on the truthfulness of the news [32].

3) Social-Context-Based Approaches: They consider the users' social viewpoints and interaction related to social media posts to decide on the truthfulness of the post [17], [19], [33], [34]. Specifically, interaction features, such as following and followers accounts, as well as reaction features related to posting activities, i.e., post, comments, replies, and reviews, represent explicit expression of users' opinions. However, the collection of social media contexts is a challenging task, as the context data could be large, incomplete, and noisy [19].

Authorized licensed use limited to: University of Edinburgh. Downloaded on November 04,2024 at 09:49:04 UTC from IEEE Xplore. Restrictions apply.

B. Fake Account Detection

Fake accounts are social bot accounts that are controlled by software program and operate under a botnet. They can be used in many malicious activities, such as DDoS attack coordination, hijacking human accounts, and contributing in large-scale manipulation campaigns to influence public opinions. Literature review considered different features for fake account detection [35], [36], [37], [38], [39], [40], [41], [42], [43], [44], [45], [46], [47]. For example, verified accounts in Twitter are surely human accounts. Also, the age of the account and the ratio of followers to following are the main features to differentiate between human and bot accounts. In general, bot accounts are short-lived and mass-follow.

C. Fake News Campaigns Detection

Fake news campaigns are characterized by a set of accounts, mostly the fake ones reproduce fake news through different posting activities, such as reposting and replying. They analyze how the news are spread on social network to identify fake news. It has been shown in [7] and [48] that real and fake news have different propagation patterns. Fake news campaigns can also be detected using the graph structure to model the propagation patterns of the news [7].

D. Comparison With Related Work

Our approach differs from related work in the following points.

1) Differently from related work, our approach focuses on identifying users' communities, i.e., the organized threat actors, which participate together in similar news campaigns.
2) Differently from fake news detection approaches, our approach does not require prior knowledge of news contents or the profiles of social accounts.
3) Differently from fake account detection approaches, which focus on identifying fake accounts, our approach aims to identify human and fake accounts that participate in fake news campaigns.
4) The approaches related to fake news campaigns detection focus on labeling the news, whereas our approach uses the news label information to label the users' communities.

IV. Methodology

A. Basic Idea

The fake news campaigns are usually orchestrated by botnets that instruct fake accounts to spread particular news. In addition, human accounts could intentionally participate in fake news campaigns to serve particular agendas. We assume that the organized threat groups, which are behind fake news, are likely to participate together in similar fake news campaigns. In addition, these groups actively participate in the early stage of the campaign. Fig. 1 shows an example of three fake news campaigns in the form of tree, with different initiator nodes. We can observe that three nodes demonstrate spatial–temporal correlation and similarities, i.e., they participate in the same campaigns before time T, which represents the duration of the early stage of the campaign. On the other hand, normal users are unlikely to exhibit such correlations and similarities. We leverage this observation to detect the organized threat groups that perform synchronized similar activities.

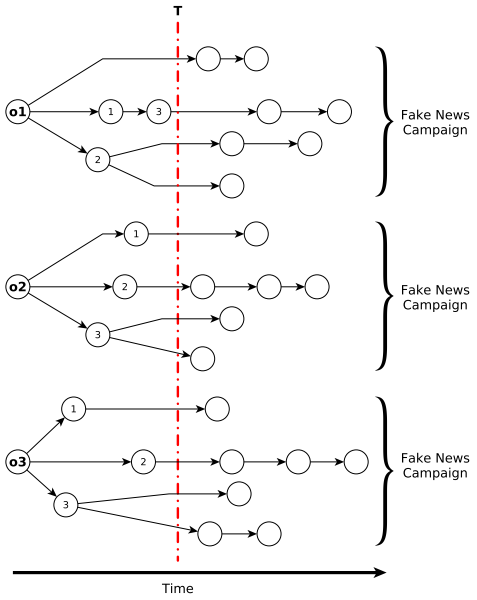

Fig. 1. Illustration of temporal propagation graphs.

B. Approach Overview

The proposed approach is depicted in Fig. 2. It is composed of the following modules.

1) *Propagation Graph Generation:* It generates the directed graph related to each news propagation. In the propagation graph, nodes represent users, and the links from user x to user y if y retweets or replies to x's tweet. These propagation graphs are used for detecting fake news campaigns.
2) *Spatial–Temporal Similarity Graph Generation:* It generates the graph that connects the users who participate in the same news campaigns during the first T time period since the beginning of the campaigns.
3) *Community Generation:* It takes the spatial–temporal similarity graph as the input and applies a community detection to cluster users into communities.
4) *Fake News Classification:* It aims to classify the campaigns as fake or real. We test different classifiers, such as DNN, which are trained using features that are extracted from the propagation graphs.
5) *Community Labeling:* It takes the output of: 1) community generation and 2) fake news classification modules to label the communities as malicious or benign.

C. Propagation Graph Generation

The propagation graph/network of an original tweet i posted by user o can be modeled as a directed graph $\mathrm{PN}_i = (U_i, L_i)$,

Authorized licensed use limited to: University of Edinburgh. Downloaded on November 04,2024 at 09:49:04 UTC from IEEE Xplore. Restrictions apply.

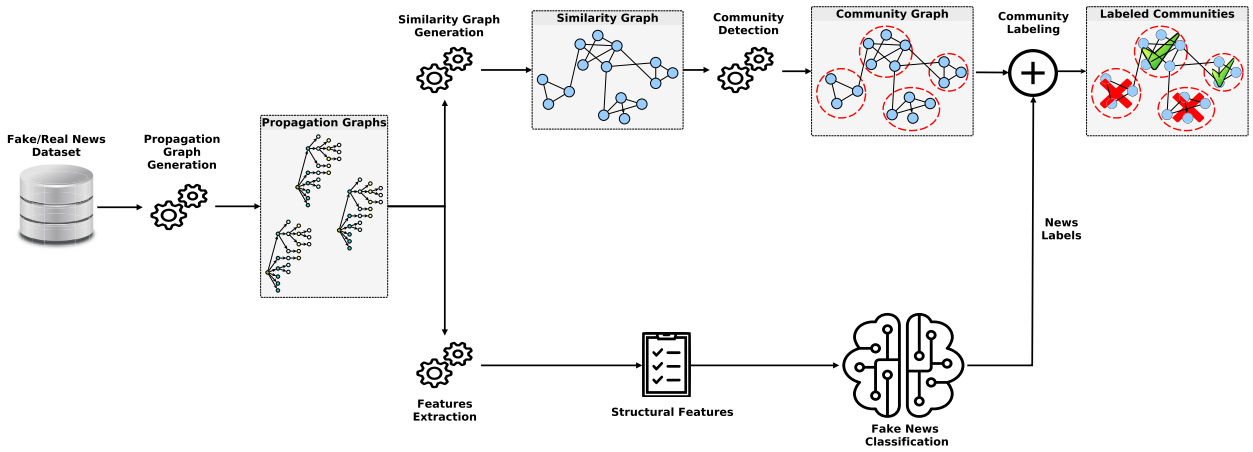

Fig. 2. Approach overview.

where U_i is the set of users who retweet or reply to the original tweet i. A directed link (x, y) is formed when user y retweets or replies to user x's tweet. We define the propagation time of user x with respect to PN_i, denoted by $\text{PT}(x, i)$ as follows:

$$\text{PT}(x, i) = \text{Timestamp}(x, i) - \text{Timestamp}(o, i)$$

where $\text{Timestamp}(x, i)$ is the time user x posts a tweet with respect to propagation graph PN_i. From each PN_i, we derive a subgraph $\text{PN}_i^T = (U_i^T, L_i^T)$, where T is the duration of news propagation, and

$$U_i^T = \{x \in U_i : \text{PT}(x, i) \leq T\}$$
$$L_i^T = \left\{(x, y) \in L_i : x \in U_i^T \wedge y \in U_i^T\right\}.$$

D. Spatial–Temporal Similarity Graph Generation

For a set of N propagation graphs $\mathbb{PN} = \{\text{PN}_1^T, \ldots, \text{PN}_i^T, \ldots, \text{PN}_N^T\}$, we define the spatial–temporal similarity graph as $\text{ST} = (V, E, \omega)$, where

1) V is the set of users in $\mathbb{PN}$.
2) An edge $(x, y) \in E$ exists if $\exists\ \text{PN}_i^T$ such that both users x and y belong to PN_i^T

$$\omega: \quad E \rightarrow \mathbb{N}^+$$
$$(x, y) \rightarrow \text{Number of shared news campaigns between } x \text{ and } y.$$

E. Community Generation

A community is defined as a partition of a network, which has local structure and clustering properties. The nodes inside a community are densely connected, i.e., a node that belongs to a community must satisfy the condition that its internal node degree inside that community exceeds its external degree. We consider nonoverlapping communities if there are no shared nodes between the communities.

The community generation module aims to extract groups of users who have the same propagation pattern with respect to the news campaigns. It takes the spatial–temporal similarity graph as the input and applies a community detection algorithm to group users into the nonoverlapping communities. We choose LPA, which is a popular nonoverlapping community detection algorithm known for its scalability, simplicity, and its ability to produce meaningful results with minimal user interaction. Compared with other algorithms, label propagation is much faster and requires less data or parameters to produce results. It is also capable of handling large networks with multiple communities [49].

F. Fake News Classification

Fake news classification uses the structural features, which are extracted from the propagation graphs to label campaigns as fake or real. Fake news classification operates in two steps: training and testing. First, structural features are extracted from the propagation graph. We use the following structural features [7] for fast classification of fake news campaigns.

- $F1$ *Tree Depth:* It gives the depth of the propagation graph and indicates how far the news is spread/retweeted by users.
- $F2$ *Number of Nodes:* It indicates the number of users participating in the propagation graph.
- $F3$ *Maximum Outdegree:* It is the degree of the node with the greatest number of links directed out of that node in the propagation graph. It reveals the node with the most influence in the graph.
- $F4$ *Number of Cascades:* It is the number of new tweets that repost the original news article.
- $F5$ *Depth of Node With Maximum Outdegree:* It is related to influential nodes (or nodes with maximum outdegree). It indicates the number of propagation steps the news takes to be spread by the influential nodes.
- $F6$ *Number of Cascades With Retweets:* It gives the number of tweets that were retweeted at least once.
- $F7$ *Fraction of Cascades With Retweets:* It indicates the fraction of tweets with retweets among all the retweets.
- $F8$ *Number of Bot Users Retweeting:* It gives the number of bot users who have retweeted the news.
- $F9$ *Fraction of Bot Users Retweeting:* It is the ratio of bot users among all the users. It shows whether the information is more likely to be spread by bots or real accounts.

Authorized licensed use limited to: University of Edinburgh. Downloaded on November 04,2024 at 09:49:04 UTC from IEEE Xplore. Restrictions apply.

In the next step, each propagation graph is represented by a vector of structural features ($F1, \ldots, F9$) and labeled according to its campaign class: fake or real. We split the dataset of propagation graphs into training and testing. The training process uses the structural features to train a classifier on the training dataset to generate the classifier. Afterward, the testing process performs predictions on unseen propagation graphs and labels their corresponding news campaigns as real or fake.

G. Community Labeling

The community labeling algorithm is illustrated in Algorithm 1. It takes as input the community graph and the labeled news campaigns, which are obtained from the fake news classification module (Algorithm 1: line 1). It then computes a maliciousness score for each community. The maliciousness of a community is based on its users' reputation scores. The reputation score of a node/user is computed as follows.

1) Botscore[1]: is a score between 0 and 1. Higher scores mean the account is more likely to be a bot (Algorithm 1: line 5).
2) Fake news participation: is the number of fake news campaigns, in which the user participated (Algorithm 1: line 6).
3) Real news participation: is the number of real news campaigns, in which the user participated (Algorithm 1: line 7).

Botscore is obtained through Botometer [50], a public tool that uses supervised machine learning to differentiate between bot-like and human-like accounts. Bot accounts or social bots are automated programs that use social media accounts to post content and interact with other accounts. These bots have the potential to be both beneficial and harmful, and malicious bots can mimic humans to spread misinformation and distort online discussions. Botometer uses a variety of features to determine whether the accounts are bots or not, such as user profile, friends, followers, language, and sentiment of contents.

The algorithm computes the maliciousness score of each community, which is obtained by averaging the scores of all the nodes that belong to that community (Algorithm 1: line 11). If the community score is higher than a predefined threshold, the community is considered as malicious; otherwise, it is labeled as benign (Algorithm 1: lines 12–16).

Algorithm 1 Community Labeling Algorithm

```
1: INPUT:
   S = {C_1, ..., C_i}: Set of communities
   G = (U, V): community graph
   L: labeled news campaigns
2: for C in S do
3:    nodes_core ← {}
4:    for Node in C do
5:       bot_score ← Get_Bot_score(Node)
6:       v1 ← Nb_fake_news(G, L, Node)
7:       v2 ← Nb_real_news(G, L, Node)
8:       reputation ← (v1 + bot_score) / (v1 + v2 + 1)
9:       nodes_score ← nodes_score U
   {reputation}
10:   end for
11:   maliciousness_score ←
   AVG(nodes_score)
12:   if maliciousness_score > Threshold then
13:      community_Label(c) ← Malicious
14:   else
15:      community_Label(c) ← Benign
16:   end if
17: end for
```

TABLE I
STATISTICS OF POLITIFACT DATASET

Variable	Value	Dataset
# Real news	624	[51]
# Fake news	432	[51]
# Propagation graphs (real news)	277	[7]
# Propagation graphs (fake news)	351	[7]
# Users	384,813	[7]
# Tweets	275,058	[7]
# Retweets	293,438	[7]
# Replies	125,654	[7]

V. IMPLEMENTATION

A. Dataset Description

Our approach is evaluated using the hierarchical propagation networks' dataset [7], which was constructed using news samples from the FakeNewsNet dataset [51]. The original dataset consists of news content from fact-checking websites that are labeled as fake or real by professional journalists. We test our approach on the PolitiFact dataset, which is described in Table I.

[1]https://botometer.osome.iu.edu/

B. Software Implementation

To evaluate our approach, we use Python3 to implement the different modules. Different classification algorithms are used to evaluate the fake news classification module. We use Scikit-learn [52] to implement the different machine learning algorithms and Keras [53] to implement DNN. In addition, the Networkx library [54] is used to visualize the different graphs, which are generated by our approach. As for the community labeling algorithm, we use LPA implementation from the Networkx library to apply community detection on the weighted spatial–temporal similarity graph. In addition, cdlib [55] is used to compute the different structural community features.

In our implementation, T parameter is set to three hours. In the community labeling algorithm, the threshold that determines the class of the community, i.e., benign or malicious, is set to 0.5. The training and testing are performed on 80% and 20% splits, respectively.

VI. EVALUATION

A. Performance Metrics

In this section, we present the metrics used to evaluate the performance of both the fake news detection and community labeling modules.

Authorized licensed use limited to: University of Edinburgh. Downloaded on November 04,2024 at 09:49:04 UTC from IEEE Xplore. Restrictions apply.

1) Fake News Classification Metrics: We use the following metrics, which are commonly used to evaluate the performance of classifiers: true/false positives, true/false negatives, accuracy, precision, recall, and F1 score.

1) True positives (TP_f) and false negatives (FN_f) are the number of correct/incorrect fake news predictions, respectively. True negatives (TN_f) and false positives (FN_f) are the number of correct/incorrect real news predictions, respectively.
2) *Accuracy (Acc_f):* is defined as the ratio of the number of correct fake news predictions to the total number of news campaigns
$$\text{Acc}_f = \frac{\text{TP}_f + \text{TN}_f}{\text{TP}_f + \text{TN}_f + \text{FP}_f + \text{FN}_f}.$$
3) *Precision ($Prec_f$):* is the ratio of the number of correctly predicted fake news campaigns to the total number of fake news campaigns
$$\text{Prec}_f = \frac{\text{TP}_f}{\text{TP}_f + \text{FP}_f}.$$
4) *Recall (Rec_f):* is the ratio of the number of correct positive predictions to the total number of positive observations
$$\text{Rec}_f = \frac{\text{TP}_f}{\text{TP}_f + \text{FN}_f}.$$
5) *F1 Score ($F1_f$):* is the weighted average of precision and recall. It takes into account both false positives and false negatives
$$\text{F1}_f = \frac{2 \times \text{Prec}_f \times \text{Rec}_f}{\text{Prec}_f + \text{Rec}_f}.$$

2) Community Labeling Metrics: The community labeling algorithm is fed with the outputs of the fake news classification module and compares them with a ground truth, which are obtained by assuming a hypothetical perfect classifier, i.e., a classifier with perfect accuracy, which is fed to the community labeling algorithm. To this end, we use the following evaluation metrics.

1) *True positives (TP_c):* is the number correctly classified malicious communities. False negatives (FN_c) give the number of incorrectly classified malicious communities. True negatives (TN_c) represent the number of correctly predicted benign communities, while false positives (FN_c) give the number of benign communities incorrectly labeled as malicious
$$\text{Prec}_c = \frac{\text{TP}_c}{\text{TP}_c + \text{FP}_c}.$$
2) *Accuracy (Acc_c):* is defined as the ratio of the number of correctly predicted malicious communities to the total number of communities
$$\text{Acc}_c = \frac{\text{TP}_c + \text{TN}_c}{\text{TP}_c + \text{TN}_c + \text{FP}_c + \text{FN}_c}.$$
3) *Precision ($Prec_c$):* is the ratio of the number of correctly predicted malicious communities to the total number of malicious communities.

TABLE II
FAKE NEWS CLASSIFICATION RESULTS

Classifier	$Acc_c(\%)$	$Prec_c(\%)$	$Rec_c(\%)$	$F1_c(\%)$
DNN	57.90	52.94	100.00	69.23
Gaussian NB	57.89	52.94	100.00	69.23
Logistic Regression	57.89	52.94	100.00	69.23
Decision Tree	68.42	61.54	88.89	72.73
Random Forest	62.11	56.38	88.89	68.99
SVM	57.89	52.94	100.00	69.23

4) *Recall (Rec_c):* is the ratio of the number of correct predictions of malicious communities to the total number of predicted malicious communities.
$$\text{Prec}_c = \frac{\text{TP}_c}{\text{TP}_c + \text{FP}_c}.$$
5) *F1 Score ($F1_c$):* is the weighted average of precision and recall
$$\text{F1}_c = \frac{2 \times \text{Prec}_c \times \text{Rec}_c}{\text{Prec}_c + \text{Rec}_c}.$$

B. Fake News Classification Results

Table II presents the performance results of fake news classification module. In our experiments, we use different machine learning classifiers, as well as a deep neural network model. The obtained results show an accuracy that ranges from 57.89% to 68.42%. The precision results vary between 52.94% and 61.54%. The recall reaches 100% for certain classifiers. F1 scores have values ranging between 69.23% and 72.73%. The obtained classification results are not very good, due to the limited number of used features. We use a small set of features, i.e., structural features, for fast detection of fake news. In addition, these values are obtained using only the first three hours of the news campaigns, which helps in ensuring early fake news detection.

C. Community Labeling Results

Fig. 3 visualizes the output of our approach, i.e., the set of communities that are obtained from applying our approach on the Politifcat dataset. The nodes in the graph are colored according to their community classes. Benign communities are colored in green, whereas malicious communities are colored in red. In Table III, we summarize Fig. 3 in numbers. It shows that there are more benign communities than malicious communities. Also, the same observation can be made on the number of nodes that are member of benign communities, which is higher than nodes belonging to malicious communities. Community labeling is also evaluated by considering the results of fake news classification module, and the results of hypothetical perfect classifier that ensures 100% accuracy. This experiment uses two different variants: one using the original equation to compute the reputation and the other one excluding the botscore from the equation.

1) With Botscore: In the first variant, the results, as depicted in the first line of Table IV, show that community labeling gives excellent results with an accuracy of 99.61%. As depicted in the confusion matrix of the community labeling algorithm (Fig. 4(a)), all the malicious communities are

Authorized licensed use limited to: University of Edinburgh. Downloaded on November 04,2024 at 09:49:04 UTC from IEEE Xplore. Restrictions apply.

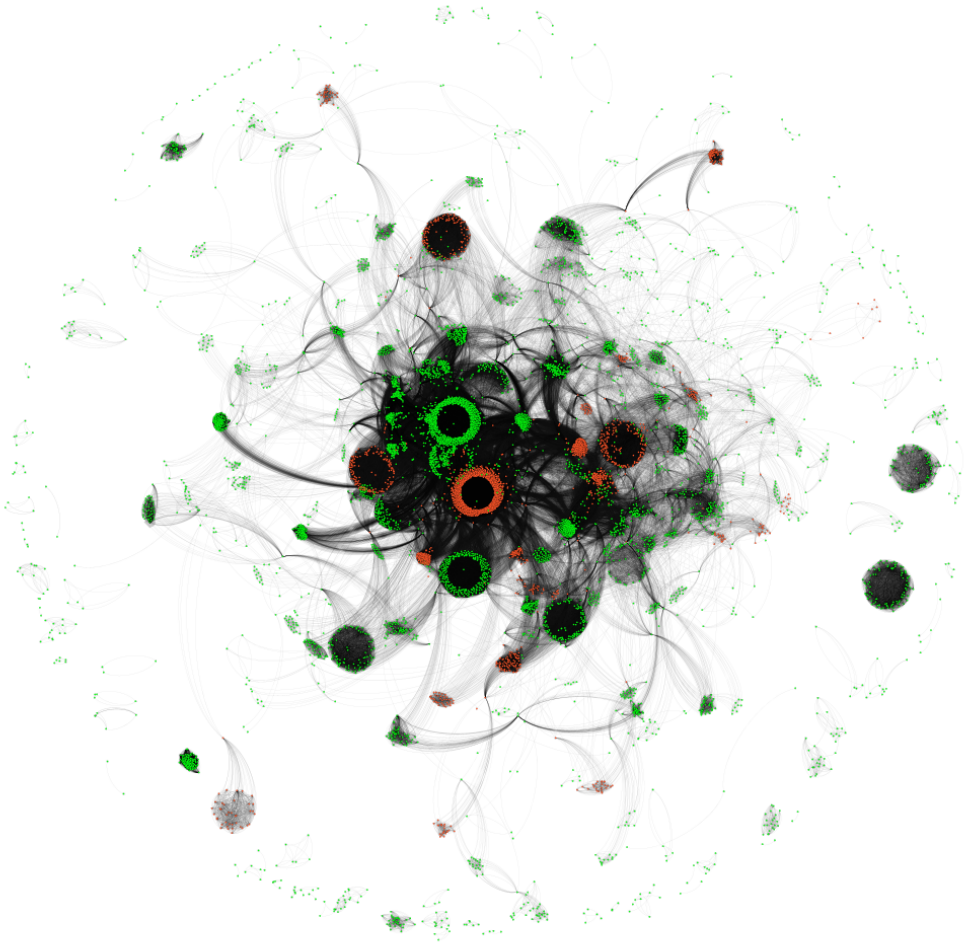

Fig. 3. Visualization output of our approach tested under the Politifcat dataset, where nodes represent users and two users are connected if they participate in the same news campaign. Malicious communities are colored in red, and benign communities are colored in green.

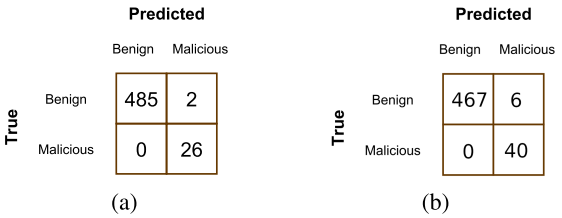

Fig. 4. Confusion matrix of the community labeling algorithm. (a) With botscore. (b) Without botscore.

correctly detected, and only two benign communities are predicted as malicious.

2) Without Botscore: The results of the second variant are presented in the second line of Table IV and in Fig. 4(b). They show that the community labeling algorithm still maintains a high accuracy, i.e., 98.83%, and is very close to the first variant. This means that the botscore slightly contributes to the performance of the algorithm. Furthermore, the confusion matrix reveals that there are more false positives when the botscore is excluded.

TABLE III
SUMMARY OF FIG. 3

Variable	Value
Number of nodes	10,487
Number of edges	1,682,607
Number of malicious communities	26
Number of benign communities	487
Number of nodes assigned to malicious communities	870
Number of nodes assigned to benign communities	9617

TABLE IV
COMMUNITY LABELING RESULTS

	$Acc_l(\%)$	$Prec_l(\%)$	$Rec_l(\%)$	$F1_l(\%)$
with Botscore	99.61	92.86	100.00	96.30
without Botscore	98.83	86.96	100.00	93.02

D. Analysis of Benign and Malicious Communities

In this section, we characterize, analyze, and compare the structure of benign and malicious communities by considering

Authorized licensed use limited to: University of Edinburgh. Downloaded on November 04,2024 at 09:49:04 UTC from IEEE Xplore. Restrictions apply.

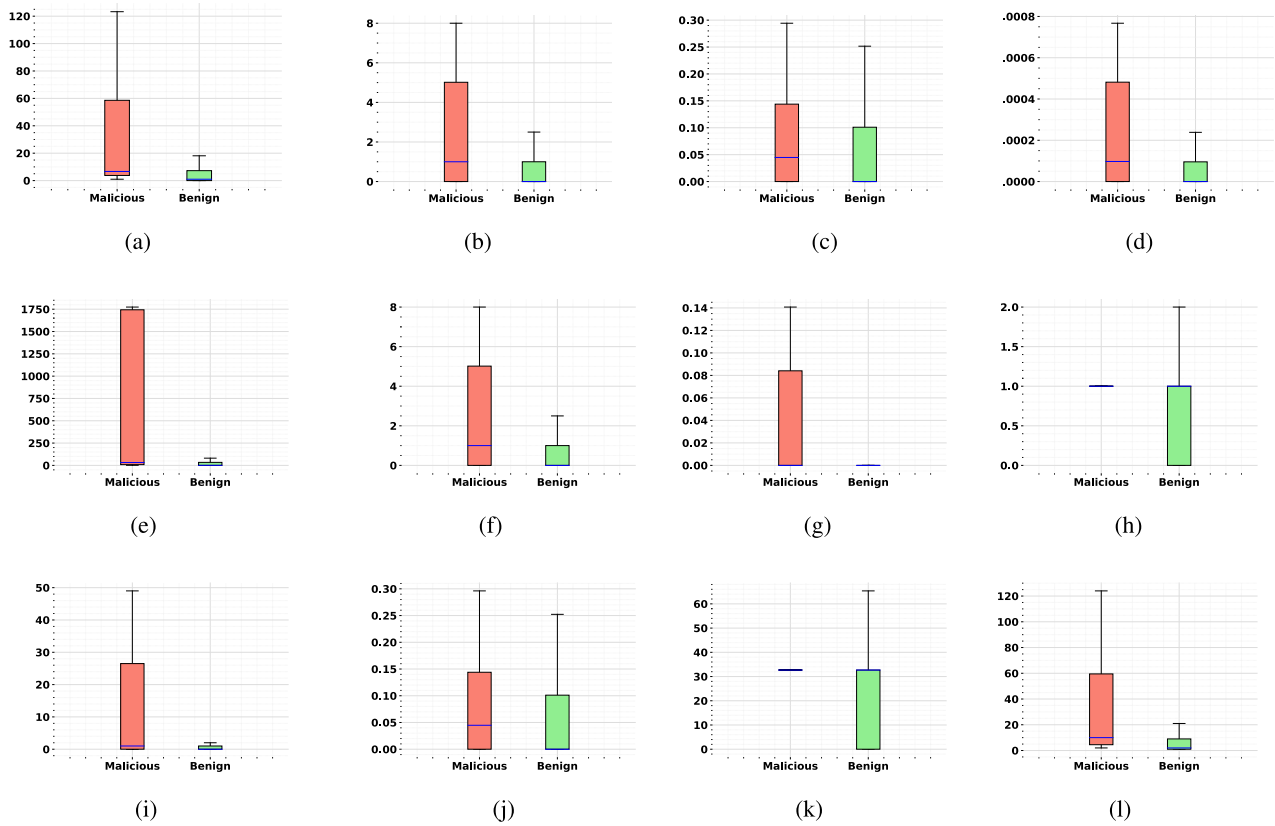

Fig. 5. Box plots demonstrating the differences in the distribution of structural community features (minimum, first quartile, median, third quartile, and maximum) of malicious and benign communities from the PolitiFact dataset. Statistically significant features are represented by an asterisk in the subfigure title. The median is shown in blue. (a) $C1$ (*). (b) $C2$ (*). (c) $C3$. (d) $C4$ (*). (e) $C5$ (*). (f) $C6$ (*). (g) $C7$. (h) $C8$. (i) $C9$ (*). (j) $C10$. (k) $C11$ (*). (l) $C12$ (*).

various structural community features. These features can be classified into three distinct groups: 1) features based on internal connectivity of the communities; 2) features based on external connectivity of the communities; and 3) features based on both internal and external connectivities [56]. We consider 12 scoring functions $f_i(S), 1 \leq i \leq 12$, which compute the values of these structural features for a network community S. In the following, m_s is defined as the number of community internal edges, n_s represents the number of community nodes, E is the set of graph edges, $d(u)$ is the degree of a node u, and c_S represents the number of nodes of community S. The structural features are the following.

1) *Average Internal Degree (C1):* It is the average internal degree of the set of communities. $f_1(S) = (2m_S/n_S)$.
2) *Average-ODF (C2):* It represents the average fraction of edges of nodes in a community that point outside the community itself. $f_2(S) = (1/n_s)\sum_{u\in S}(|(u,v) \in E : v \notin S|/d(u))$.
3) *Conductance (C3):* It is the fraction of total edge volume that points outside the community. $f_3(S) = (c_S/(2m_S + c_S))$.
4) *Cut Ratio (C4):* It is defined as the fraction of all possible existing edges quitting the community. $f_4(S) = (c_S/(n_S(n - n_S)))$.
5) *Edges Inside (C5):* It is the number of edges internal to the community.
6) *Expansion (C6):* It represents the number of edges per community node that point outside the cluster. $f_6(S) = (c_S/n_S)$.
7) *Fraction Over Median Degree (C7):* It is defined as the fraction of community nodes having internal degree higher than the median degree value. $f_7(S) = (|u : u \in S, |(u,v) : v \in S| > d_m|/n_S)$.
8) *Internal Edge Density (C8):* It is the internal density of the community set. $f_8(S) = \frac{m_S}{(n_S(n_S-1)/2)}$.
9) *Max-ODF (C9):* It represents the maximum fraction of edges of a node of a community that point outside the community itself. $f_9(S) = \max_{u\in S}(|(u,v) \in E : v \notin S|/d(u))$.
10) *Normalized Cut (C10):* It is the normalized variant of the cut ratio. $f_{10}(S) = (c_S/(2m_S + c_S)) + (c_S/(2(m - m_S) + c_S))$.
11) *Scaled Density (C11):* It is defined as the ratio of the community density with respect to the complete graph density.
12) *Size (C12):* It represents the number of nodes of a community.

We apply t-test, a statistical test that is used to compare the means of two groups, to select the features that are statistically significant between two groups (Table V). The larger the value, the more significant the difference between the groups [57]. The two-sample t-test is as

Authorized licensed use limited to: University of Edinburgh. Downloaded on November 04,2024 at 09:49:04 UTC from IEEE Xplore. Restrictions apply.

TABLE V
ANALYSIS OF COMMUNITY LABELING FEATURES

	Malicious Communities			Benign Communities		
Features	Min	Max	Avg	Min	Max	Avg
$C1$*	1.000	1149.198	89.321	0.000	839.304	14.544
$C2$*	0	51.395	5.949	0.000	67.305	2.020
$C3$	0.000	0.400	0.087	0.000	0.824	0.079
$C4$*	0.000	0.005	0.001	0.000	0.007	0.000
$C5$*	1.000	663k	291k	0.000	432k	1664.316
$C6$*	0.000	51.395	5.949	0.000	67.305	2.020
$C7$	0.000	0.250	0.050	0.000	0.500	0.028
$C8$	0.711	3.000	1.055	0.000	2.000	0.570
$C9$*	0.000	2065.000	181.963	0.000	1726.000	17.649
$C10$	0.000	0.400	0.088	0.000	0.824	0.080
$C11$*	23.237	98.032	34.466	0.000	65.355	18.616
$C12$*	2.000	1154.000	93.593	1.000	1029.000	16.438

follows:

$$t = \frac{x1 - x2}{\sqrt{s^2\left(\frac{1}{n1} + \frac{1}{n2}\right)}}$$

where $x1$ and $x2$ are the means of the two groups being compared, and s^2 is the pooled standard error of these two groups. $n1$ and $n2$ are the number of observations in the two groups.

Table V and Fig. 5 depict the distribution of community structural features. The statistically significant features are denoted by an asterisk in the feature name.

Based on the results in Table V and Fig. 5, we make the following observations.

1) $C1$, $C2$, $C4$–$C6$, $C9$, $C11$, and $C12$ features are different—under t-test—from the malicious and benign communities.
2) Malicious communities have in average a higher internal average degree ($C1$) than benign communities. This means that the groups of malicious users usually participate in the same campaigns more than the groups of benign users.
3) By observing some features that take into account the external connections of a community ($C4$ and $C6$), we conclude that users from malicious communities participate in more real campaigns than those users from benign communities that participate in fake campaigns.
4) Features that take into account both the internal and external connections of communities have higher values for malicious campaigns.

VII. CONCLUSION

In this article, we have proposed a community-based approach to cluster the organized groups that participate together in fake news campaigns. Our approach is low-cost, as it does not require prior knowledge of the news content or the profiles of social accounts. It first builds the spatial–temporal similarity graph, which connects social accounts that participate in the early stage of similar fake news campaigns. Then, it applies a community detection algorithm on the spatial–temporal similarity graph to cluster the users into communities. Finally, we propose a community labeling algorithm to label the communities as benign or malicious, by taking the output of a fake news classifier, i.e., real or fake. Although the fake news classifiers are not effective in distinguishing between fake and real campaigns, i.e., around 57.9% accuracy is achieved, but the evaluation results show that the community labeling algorithm can correctly label the communities with an accuracy of 99.61%. Furthermore, we perform a statistical comparison analysis to identify the structural community features, such as average internal degree, edges inside, scaled density, and community size, which are statistically significant between benign and malicious communities. As future work, we plan to study the effect of different community structural features related on the performance of our approach. In addition, we plan to test our approach on other fake news datasets. Finally, we can investigate the impact of parameter T on the effectiveness of detecting malicious communities.

REFERENCES

[1] X. Zhou and R. Zafarani, "A survey of fake news: Fundamental theories, detection methods, and opportunities," *ACM Comput. Surv.*, vol. 53, no. 5, pp. 1–40, Sep. 2021.
[2] P. Biyani, K. Tsioutsiouliklis, and J. Blackmer, "'8 Amazing secrets for getting more clicks': Detecting clickbaits in news streams using article informality," in *Proc. AAAI Conf. Artif. Intell.*, 2016, pp. 94–100.
[3] Y. Chen, N. J. Conroy, and V. L. Rubin, "Misleading online content: Recognizing clickbait as 'false news,'" in *Proc. ACM Workshop Multimodal Deception Detection*, Nov. 2015, pp. 15–19.
[4] D. Michail, N. Kanakaris, and I. Varlamis, "Detection of fake news campaigns using graph convolutional networks," *Int. J. Inf. Manage. Data Insights*, vol. 2, no. 2, Nov. 2022, Art. no. 100104.
[5] M. Meyers, G. Weiss, and G. Spanakis, "Fake news detection on Twitter using propagation structures," in *Proc. Multidisciplinary Int. Symp. Disinformation Open Online Media*. Cham, Switzerland: Springer, 2020, pp. 138–158.
[6] T. Murayama, S. Wakamiya, E. Aramaki, and R. Kobayashi, "Modeling the spread of fake news on Twitter," *PLoS ONE*, vol. 16, no. 4, Apr. 2021, Art. no. e0250419.
[7] K. Shu, D. Mahudeswaran, S. Wang, and H. Liu, "Hierarchical propagation networks for fake news detection: Investigation and exploitation," 2019, *arXiv:1903.09196*.
[8] P. Bedi and C. Sharma, "Community detection in social networks," *Wiley Interdiscipl. Rev., Data Mining Knowl. Discovery*, vol. 6, no. 3, pp. 115–135, May 2016.
[9] Z. Liu and Y. Ma, "A divide and agglomerate algorithm for community detection in social networks," *Inf. Sci.*, vol. 482, pp. 321–333, May 2019.

Authorized licensed use limited to: University of Edinburgh. Downloaded on November 04,2024 at 09:49:04 UTC from IEEE Xplore. Restrictions apply.

[10] X. You, Y. Ma, and Z. Liu, "A three-stage algorithm on community detection in social networks," *Knowl.-Based Syst.*, vol. 187, Jan. 2020, Art. no. 104822.
[11] R. Hosseini and A. Rezvanian, "AntLP: Ant-based label propagation algorithm for community detection in social networks," *CAAI Trans. Intell. Technol.*, vol. 5, no. 1, pp. 34–41, Mar. 2020.
[12] E. D. Raj, G. Manogaran, G. Srivastava, and Y. Wu, "Information granulation-based community detection for social networks," *IEEE Trans. Computat. Social Syst.*, vol. 8, no. 1, pp. 122–133, Feb. 2021.
[13] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *J. Stat. Mech., Theory Exp.*, vol. 2008, no. 10, Oct. 2008, Art. no. P10008.
[14] M. Rosvall and C. T. Bergstrom, "Maps of random walks on complex networks reveal community structure," *Proc. Nat. Acad. Sci. USA*, vol. 105, no. 4, pp. 1118–1123, Jan. 2008.
[15] U. N. Raghavan, R. Albert, and S. Kumara, "Near linear time algorithm to detect community structures in large-scale networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 76, no. 3, Sep. 2007, Art. no. 036106.
[16] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proc. Nat. Acad. Sci. USA*, vol. 99, no. 12, pp. 7821–7826, Jun. 2002.
[17] Y. Liu and Y.-F.-B. Wu, "FNED: A deep network for fake news early detection on social media," *ACM Trans. Inf. Syst.*, vol. 38, no. 3, pp. 1–33, Jul. 2020.
[18] N. R. de Oliveira, P. S. Pisa, M. A. Lopez, D. S. V. de Medeiros, and D. M. F. Mattos, "Identifying fake news on social networks based on natural language processing: Trends and challenges," *Information*, vol. 12, no. 1, p. 38, Jan. 2021.
[19] S. Raza and C. Ding, "Fake news detection based on news content and social contexts: A transformer-based approach," *Int. J. Data Sci. Anal.*, vol. 13, pp. 1–28, May 2022.
[20] X. Zhou, A. Jain, V. V. Phoha, and R. Zafarani, "Fake news early detection: A theory-driven model," *Digit. Threats, Res. Pract.*, vol. 1, no. 2, pp. 1–25, Jun. 2020.
[21] N. Seddari, A. Derhab, M. Belaoued, W. Halboob, J. Al-Muhtadi, and A. Bouras, "A hybrid linguistic and knowledge-based analysis approach for fake news detection on social media," *IEEE Access*, vol. 10, pp. 62097–62109, 2022.
[22] H. Ahmed, I. Traoré, and S. Saad, "Detection of online fake news using N-gram analysis and machine learning techniques," in *Proc. Int. Conf. Intell., Secure, Dependable Syst. Distrib. Cloud Environ.*, 2017, pp. 127–138.
[23] J. Fairbanks, N. Fitch, N. Knauf, and E. Briscoe, "Credibility assessment in the news: Do we need to read?" in *Proc. MIS2 Workshop Held Conjuction With 11th Int. Conf. Web Search Data Mining*, 2018, pp. 799–800.
[24] M. Granik and V. Mesyura, "Fake news detection using naive Bayes classifier," in *Proc. IEEE 1st Ukraine Conf. Electr. Comput. Eng. (UKRCON)*, May 2017, pp. 900–903.
[25] B. Horne and S. Adali, "This just in: Fake news packs a lot in title, uses simpler, repetitive content in text body, more similar to satire than real news," in *Proc. Int. AAAI Conf. Web Social Media*, vol. 11, Mar. 2017, pp. 759–766.
[26] H. E. Wynne and Z. Z. Wint, "Content based fake news detection using N-gram models," in *Proc. 21st Int. Conf. Inf. Integr. Web-based Appl. Services*, Dec. 2019, pp. 669–673.
[27] N. K. Conroy, V. L. Rubin, and Y. Chen, "Automatic deception detection: Methods for finding fake news," *Proc. Assoc. Inf. Sci. Technol.*, vol. 52, no. 1, pp. 1–4, Jan. 2015.
[28] Y. Lahlou, S. El Fkihi, and R. Faizi, "Automatic detection of fake news on online platforms: A survey," in *Proc. 1st Int. Conf. Smart Syst. Data Sci. (ICSSD)*, Oct. 2019, pp. 1–4.
[29] D. Shakeel and N. Jain, "Fake news detection and fact verification using knowledge graphs and machine learning," 2021, doi: 10.13140/RG.2.2.18349.41448.
[30] Y. Liu and Y.-F. Wu, "Early detection of fake news on social media through propagation path classification with recurrent and convolutional networks," in *Proc. AAAI Conf. Artif. Intell.*, vol. 32, no. 1, 2018, pp. 354–361.
[31] K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu, "Fake news detection on social media: A data mining perspective," *ACM SIGKDD Explor. Newslett.*, vol. 19, no. 1, pp. 22–36, Sep. 2017.
[32] S. Cohen, C. Li, J. Yang, and C. Yu, "Computational journalism: A call to arms to database researchers," in *Proc. 5th Biennial Conf. Innov. Data Syst. Res. (CIDR)*, 2011, pp. 148–151.
[33] C. Castillo, M. Mendoza, and B. Poblete, "Information credibility on Twitter," in *Proc. 20th Int. Conf. World Wide Web*, Jan. 2011, pp. 675–684.
[34] K. Shu, S. Wang, and H. Liu, "Exploiting tri-relationship for fake news detection," 2017, *arXiv:1712.07709*.
[35] A. Derhab, R. Alawwad, K. Dehwah, N. Tariq, F. A. Khan, and J. Al-Muhtadi, "Tweet-based bot detection using big data analytics," *IEEE Access*, vol. 9, pp. 65988–66005, 2021.
[36] S. Kudugunta and E. Ferrara, "Deep neural networks for bot detection," *Inf. Sci.*, vol. 467, pp. 312–322, Oct. 2018.
[37] F. Wei and U. T. Nguyen, "Twitter bot detection using bidirectional long short-term memory neural networks and word embeddings," in *Proc. 1st IEEE Int. Conf. Trust, Privacy Secur. Intell. Syst. Appl. (TPS-ISA)*, Dec. 2019, pp. 101–109.
[38] C. Cai, L. Li, and D. Zengi, "Behavior enhanced deep bot detection in social media," in *Proc. IEEE Int. Conf. Intell. Secur. Informat. (ISI)*, Jul. 2017, pp. 128–130.
[39] L. Luo, X. Zhang, X. Yang, and W. Yang, "Deepbot: A deep neural network based approach for detecting Twitter bots," *IOP Conf. Ser., Mater. Sci. Eng.*, vol. 719, no. 1, 2020, Art. no. 012063.
[40] J. Knauth, "Language-agnostic Twitter-bot detection," in *Proc. Int. Conf. Recent Adv. Natural Lang. Process. (RANLP)*, 2019, pp. 550–558.
[41] J. Lundberg, J. Nordqvist, and M. Laitinen, "Towards a language independent Twitter bot detector," in *Proc. DHN*, 2019, pp. 308–319.
[42] K. Kiran, C. Manjunatha, T. S. Harini, P. D. Shenoy, and K. R. Venugopal, "Identification of anomalous users in Twitter based on user behaviour using artificial neural networks," in *Proc. IEEE 5th Int. Conf. Converg. Technol. (ICT)*, Mar. 2019, pp. 1–5.
[43] M. Haidermota, A. Pansare, and D. Mitra, "Classifying Twitter user as a bot or not and comparing different classification algorithms," *Int. J. Adv. Res. Comput. Sci.*, vol. 9, no. 3, p. 29, 2018.
[44] B. Erşahin, O. Aktaş, D. Kılınç, and C. Akyol, "Twitter fake account detection," in *Proc. Int. Conf. Comput. Sci. Eng. (UBMK)*, Oct. 2017, pp. 388–392.
[45] J. Rodríguez-Ruiz, J. I. Mata-Sánchez, R. Monroy, O. Loyola-González, and A. López-Cuevas, "A one-class classification approach for bot detection on Twitter," *Comput. Secur.*, vol. 91, Apr. 2020, Art. no. 101715.
[46] F. Morstatter, L. Wu, T. H. Nazer, K. M. Carley, and H. Liu, "A new approach to bot detection: Striking the balance between precision and recall," in *Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining (ASONAM)*, Aug. 2016, pp. 533–540.
[47] A. Alarifi, M. Alsaleh, and A. Al-Salman, "Twitter Turing test: Identifying social machines," *Inf. Sci.*, vol. 372, pp. 332–346, Dec. 2016.
[48] Z. Zhao et al., "Fake news propagates differently from real news even at early stages of spreading," *EPJ Data Sci.*, vol. 9, no. 1, p. 7, 2020.
[49] S. E. Garza and S. E. Schaeffer, "Community detection with the label propagation algorithm: A survey," *Phys. A, Stat. Mech. Appl.*, vol. 534, Nov. 2019, Art. no. 122058.
[50] K.-C. Yang, E. Ferrara, and F. Menczer, "Botometer 101: Social bot practicum for computational social scientists," *J. Comput. Social Sci.*, vol. 5, pp. 1–18, Aug. 2022.
[51] K. Shu, D. Mahudeswaran, S. Wang, D. Lee, and H. Liu, "FakeNewsNet: A data repository with news content, social context and spatialtemporal information for studying fake news on social media," 2018, *arXiv:1809.01286*.
[52] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Oct. 2011.
[53] A. Gulli and S. Pal, *Deep Learning With Keras*. Birmingham, U.K.: Packt, 2017.
[54] A. Hagberg and D. Conway. (2020). *NetworkX: Network Analysis With Python*. [Online]. Available: https://networkx.github.io
[55] G. Rossetti, L. Milli, and R. Cazabet, "CDLIB: A Python library to extract, compare and evaluate communities from complex networks," *Appl. Netw. Sci.*, vol. 4, no. 1, pp. 1–26, Dec. 2019.
[56] J. Yang and J. Leskovec, "Defining and evaluating network communities based on ground-truth," in *Proc. ACM SIGKDD Workshop Mining Data Semantics*, 2012, pp. 1–8.
[57] H.-Y. Kim, "Statistical notes for clinical researchers: The independent samples t-test," *Restorative Dentistry Endodontics*, vol. 44, no. 3, 2019, pp. 1–6.

Authorized licensed use limited to: University of Edinburgh. Downloaded on November 04,2024 at 09:49:04 UTC from IEEE Xplore. Restrictions apply.