

Complex Network and Source Inspired COVID-19 Fake News Classification on Twitter

KHUBAIB AHMED QURESHI¹, RAUF AHMED SHAMS MALICK², MUHAMMAD SABIH³, AND HOCINE CHERIFI⁴

¹Department of Computer Science, DHA Suffa University, Karachi 75500, Pakistan

²Department of Computer Science, National University of Computer and Emerging Sciences, Karachi 75300, Pakistan

³Department of Electrical Engineering, DHA Suffa University, Karachi 75500, Pakistan

⁴Laboratoire Informatique de Bourgogne, LIB, University of Burgundy, 21078 Dijon, France

Corresponding author: Khubaib Ahmed Qureshi (k.ahmed@dsu.edu.pk)

ABSTRACT In COVID-19 related infodemic, social media becomes a medium for wrongdoers to spread rumors, fake news, hoaxes, conspiracies, astroturf memes, clickbait, satire, smear campaigns, and other forms of deception. It puts a tremendous strain on society by damaging reputation, public trust, freedom of expression, journalism, justice, truth, and democracy. Therefore, it is of paramount importance to detect and contain unreliable information. Multiple techniques have been proposed to detect fake news propagation in tweets based on tweets content, propagation on the network of users, and the profile of the news generators. Generating human-like content allows deceiving content-based methods. Network-based methods rely on the complete graph to detect fake news, resulting in late detection. User profile-based techniques are effective for bots or fake accounts detection. However, they are not suited to detect fake news from original accounts. To deal with the shortcomings in existing methods, we introduce a source-based method focusing on the news propagators' community, including posters and re-tweeters to detect such contents. Propagators are connected using follower-following relations. A feature set combining the connectivity patterns of news propagators with their profile features is used in a machine learning framework to perform binary classification of tweets. Complex network measures and user profile features are also examined separately. We perform an extensive comparative analysis of the proposed methodology on a real-world COVID-19 dataset, exploiting various machine learning and deep learning models at the community and node levels. Results show that hybrid features perform better than network features and user features alone. Further optimization demonstrates that Ensemble's boosting model CATBoost and deep learning model RNN are the most effective, with an AUC score of 98%. Furthermore, preliminary results show that the proposed solution can also handle fake news in the political and entertainment domain using a small training set.

INDEX TERMS Context based model, COVID-19, complex network measures, dis/misinformation, hybrid model, machine learning, pandemics, source based, social media, social network analysis.

I. INTRODUCTION

Social media platforms (e.g., WhatsApp, Sina Weibo, Twitter, Facebook, etc.) have completely changed information dissemination. They are free and easy to access, support fast dissemination, and are highly engaging. Therefore, these digital platforms have gained great popularity attracting large numbers of users [1]. They allow following a variety of events and breaking news across the world [2]. As the Variety, Volume, and Velocity of information transmission have substantially increased, it has led to increased impact and

visibility of real and misleading information as well [3]. It is stated in a survey that 65% of news on social media is fake news [4]. In social media, fake information spreads wider, deeper, and faster [5]. False information has severe and wide consequences to society [6], economy [7], freedom of expression [5], journalism [8], democracy [9], reputation, public trust [10], and peace [11], [12]. Unfortunately, people get news from social media more than traditional news media [13], [14]. Consuming news from such social media platforms has its associated pros and cons. Its use to spread misleading information is straightforward, and it can influence large audiences. It can be done for political, financial, or other gains by an organization or individuals. The 2016

The associate editor coordinating the review of this manuscript and approving it for publication was Mingbo Zhao¹.

US presidential election campaign is a notable instance where the audience has been manipulated using fake news by almost 19 million malicious bot profiles [9], [15]. It could be done due to the influence of fake news that someone who had no chance according to the polls get elected [15]. Fake news is becoming a severe problem because it negatively impacts individuals and society as a whole. Economies worldwide are vulnerable to fake news, which seriously affects the stock market, massive trades, and business-related activities. For example, fake news claiming the injury of Barack Obama in an explosion wiped out 130 billion dollars in the stock market [16]. Fake information may invoke a false sense of feelings related to fear and surprise [5] which ultimately creates social panic [17], [18]. It is pretty difficult to correct someone's cognition after fake information has gained trust [10], [19]. Therefore earning money by influencing opinions [20], [21] is also easy through false information.

It is essential to understand few related terms of misleading information. Indeed, false information or deceptive information has various flavors: Fake/ False News, Misinformation, Disinformation, Hoaxes, Propaganda, Satire, Rumors, Click-Bait, and Junk News, etc., [22]–[26]. Although standardized definitions are missing, it's generally agreed that misinformation is inaccurate and misleading information. It could spread unintentionally in contrast to disinformation which is false information spreading deliberately to deceive people. Developing a mechanism to defeat different forms of misinformation and disinformation is required. However, detecting misleading information in its different forms on social networks and social media microblogs presents unique challenges. First, it is written (intentionally or unintentionally) to mislead readers, making it difficult to detect simply based on content. Second, social media data is highly large-scale, multi-lingual, multi-modal, mostly user-generated, free from grammar and spellings, sometimes anonymous and noisy. Finally, the spread of information on social media is fast, and hence detection mechanisms need to predict misleading information fast enough to stop its dissemination and severe repercussions. Therefore, detecting such information on social media is an extremely important and also technically challenging problem [25].

One can distinguish four main approaches to deal with this issue: Content-based methods, Network/propagation-based methods, Hybrid Based methods, and source-based methods.

Content-based methods try to identify deception directly through the content [27]. They present many limitations, such as contents are multi-lingual therefore, requiring separate models. Additionally, contents are intentionally written to mislead users by mimicking the truth. Therefore even style-based techniques are ineffective.

Propagation/Network-based methods exploit the news propagation network properties (tweet's and retweet network) [6] to feed supervised machine learning classifiers for the required deception detection task. As they need complete or enough propagation, information detection is very late and ultimately suffering with associated grave repercussions.

Hybrid-based methods [28] combine both news content and news propagation network-related information to predict deception through supervised classifiers. Finally, source-based methods [29] focus on the news producers to identify common patterns of unreliable users.

Considering the above few challenges, neither content-based methods nor network/ propagation-based methods are the right choices. Therefore, we propose a novel source-based solution where the information propagators are identified through their context, indirectly for deception detection.

To the best of our knowledge, it is the first attempt where complex network measures and user profile-based features are extracted from the propagators of each news article. Each news article that is either categorized as legitimate or misleading is posted/re-posted by many users. These users are from a respective community that is legitimate or misleading. The above features are applied to the whole community and every news propagator participating in the community. Supervised machine learning and deep learning models are used for the prediction task.

Figure 1 illustrates the major phases of our approach. A brief overview follows.

First, a Real-world COVID-19 Dataset is constructed. Second, four approaches exploiting user profile features and complex network measures are evaluated at the node and the community level. Third, various models are investigated, and the top four models among the best two levels are selected for best features selection using the model's feature importance. Fourth, the best model is further improved using a final set of features.

The main contributions of this work are as follows.

1. We Build a twitter-based real-world COVID-19 dataset for fake news detection. This dataset includes distinct features. It is a dataset with communities, which have additional links between re-tweeters and posters in addition to their context. It provides a crucial resource for COVID-19 fake news detection research studies. It could also be of prime interest for influence estimations and epidemic spreading analysis based on various propagation models.

2. We discuss many fundamental theories that could help the research community to develop theory-guided stronger solutions. These theories also enlightened the study, and a novel theory-driven model is developed. The proposed model is source-based and enriched with hybrid features.

3. We explore complex network measures to identify the news labels through their community's interconnection patterns. To the best of our knowledge, it is the first attempt to use these features in COVID-19 related fake news detection.

4. We also examine User profile features for the same purpose.

5. In this exploratory study, we investigate the effectiveness of both user profile features and complex network measures. Therefore, we explore different combinations of both types of features at the whole community level and the individual's level.

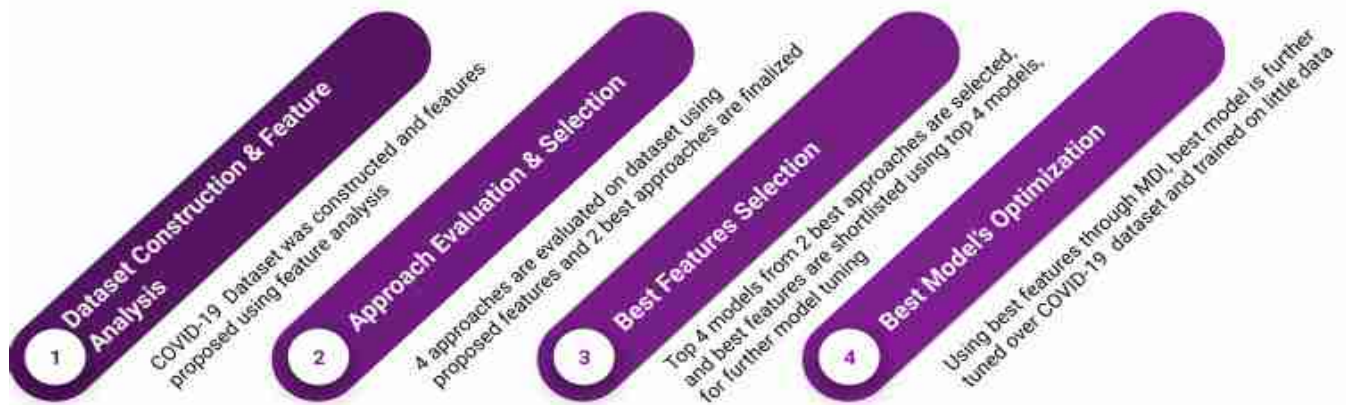


FIGURE 1. Flow of Work: The following four major phases are conducted in the study. 1. (a): Real-world Twitter-based COVID-19 Dataset for fake news detection is constructed. 1.(b): Important features are selected through feature analysis from two groups A) Network-based Features B) User-based Profile Features. These two groups of features are explored, and their effectiveness is examined. Only short-listed features are further used for the next phase. 2: Two groups of features are examined on two levels: node level and community level. Four approaches are investigated. In these approaches, user profile features and complex network measures are evaluated at two different levels. 3: Eleven ML and two deep learning models are examined for each approach. The two best approaches are selected for further experiments. The top four models among the best two approaches are also selected for best features selection using the model's feature importance through mean decrease impurity (MDI). 4: Using a final set of features, the best models are further improved and evaluated.

6. We identify the dominant complex network measures and user profile-based features of such community's users.

7. We give recommendations about the best features and the best approaches. There has never been a similar exploratory study conducted using source-based methods.

8. We perform a comprehensive comparative study with alternative methods.

The remaining of the paper is organized as follows. Section II presents preliminaries on automatic fake news detection. Section III discuss briefly related works. Section IV reports the composition of the COVID-19 dataset. Detail methodology of the research is discussed in section V. Feature analysis is conducted in section VI. Section VII discusses the adequacy of various classification approaches to the problem. Section VIII presents fake classification modeling including model selection and features selection. Section IX starts digging the best two approaches further in detail. The comparative study with the baselines follows in section X. Finally the conclusion is presented in section XI.

II. LITERATURE REVIEW

Methods are briefly discussed for a basic understanding of major methods used in deception detection to highlight their gaps/shortcomings and the importance of the appropriate method selected for the solution. There are many surveys on automatic fake news detection where authors discuss different approaches (e.g., ML features, latent features, graph, IR, NLP, etc.) with their strengths and weaknesses [19], [25], [26], [30], [31]. Perspectives and new challenges on who, how, and why fake news are created, how they propagate, and how they could be detected. are also presented [32].

One can consider four major categories in fake news detection studies. Following are some of their main characteristics.

1. Content-based methods: They focus on contents only, and no auxiliary information is needed. They don't wait for

content to be propagated. These methods are much explored. Exploiting limited information, they are suited to early detection but suffer from limitations of language and domain (e.g., politics, entertainment, etc.). They can be further classified as a. Knowledge-based, b. Style-based/linguistic, and c. Stance-based.

a. Knowledge-based is further classified as (Expert-based: [33], Crowd-sourced: [34], Crowded-sourced Site: [35], IR-based: [36], Semantic Web/SPO: [37]) and b. Style based/linguistic methods are classified as (NLP: [38], [39], ML: [40], [41], DL: [42]) and c. Stance-based methods [43]

a. Knowledge-based: There are different methods to check the authenticity of news from external sources called Knowledge-based methods. These are ineffective when news items correspond to new events. Knowledge-based methods are either Manual or Automatic. Automatic methods are scalable, in contrast to manual.

b. Style-based [44]/linguistic [27] methods: They are best for identifying the intention but fail when the domain changes because each domain differs in style. They are also weak in bot/malicious profiles detection because the bot also masquerades style. It is also very challenging for these methods to adopt evolutionary styles when language is not formal (e.g., social media microblog). It becomes tough to detect malicious information only through content because it mimics real news to mislead users.

c. Stance-based: content-based methods use a very dominant mechanism named stance classification. Fake news is also detected using stance classification. The news content propagates in tweets. One uses their respective reply for stance identification. Once the news is posted through a tweet, then each reply has a different stance. For news-related stances, there are four commonly used labels, i.e., Support, Deny, Query, Commenting (SDQC respectively). A tweet is classified as fake if a majority of its replies are classified as

Deny (D) or Deny (D) with Query (Q) stances. There are many successful studies for fake news detection based on stance classification [43].

2. Propagation/Network-based methods: They use information exposed through news cascades [6], [45]. They may also use some custom networks constructed through different system entities. They can cover different variations of misleading information, i.e., rumor, spam, etc. Combining style-based methods with propagation-based methods makes malicious profiles detection much easier. Propagation-based solutions are domain-specific (e.g., politics, entertainment, health, etc.) because each domain has different propagation patterns. These methods could hardly work before misleading information has been fully propagated on social media [19], which means late detection.

3. Hybrid methods: In addition to content-based and propagation-based methods, hybrid methods are very popular. They combine functionalities of other methods to overcome the limitations. Examples of hybrid methods include (e.g.: Propagation based + Content-based: [28], Content-based + Social Context-based: [46], [47]).

4. Source-based methods: Although they are not very much studied, they have a high potential. Quiet open area for exploration, they indirectly focus on contents through the source identification of misleading information [29]. They can assess suspicious behavior and identify a variety of deceptive information forms. The common idea behind such methods lies in the fact that one can identify many common patterns of unreliable users, helping to assess the reliability of unknown users in advance. Generally, misleading information mostly comes from malicious sources [48] therefore, identifying such sources is malicious information detection. Theories such as social validation theory [49] and the most popular homophily [50] indicate that users in a network/community always share similar behaviors or attributes. For example, they tend to post similar news articles. Therefore, on social media, users mostly form connections or groups with like-minded people resulting in an echo chamber effect [51]. Fake news spreaders/malicious users have quite distinct connectivity patterns, e.g., they mostly form denser networks [52] and perform collective actions [53] to create more polarized groups and fully exploit the echo-chamber effect. There are many theories [54] presenting distinct attributes at the user level and network/community level as a whole which further classify source-based methods in 4.2. Social media user-based methods, which are further classified as 4.2.a. profile/user-based, 4.2.b. network-based, or 4.2.c. combinations/fusion. Source-based methods are initially classified into the following two types: 4.1. Author/Publisher based [55], [56] and 4.2. Social Media User-based [57] which are further classified as; 4.2.a. profile/user, 4.2.b. network, and 4.2.c. fusion/hybrid-based methods.

A. MODEL'S APPROACH AND MOTIVATION

Let's now focus on essential aspects of the model's motivations. Some recent studies inspired us, leading us towards the

proposed directions and enriching the model with their exciting findings. The proposed source-based method combines user profile-based features with complex network measures, exploiting social media users, posting legitimate or misleading content.

Some studies highlight the importance of profile-based features to distinguish between legitimate and misleading. The Exploratory study conducted in [58] shows that some implicit features like IsVerified, age of the account, status count, favorite count are good clues. Some explicit features like personality, gender, age, and the ratio of followers to following (TFF) are highly discriminating. In [59], The authors develop an efficient machine learning system and recommend implicit features (Status Count, Account age, Verified) and explicit features (Location, Political bias, Profile image). The study conducted in [27] concludes that the age of the account, location, bot score, list count, and other post-related context features are discriminating. Related research studies highly recommend some user-based features are in (Verified account, no of followers and following, ratios of followers and followings named Reciprocity and Reputation) [60], [61].

In [62], the authors claim that no attention has been directed towards modeling the properties of online communities interacting with the posts. They show that it is essential, and they got exciting results as well. It is an important motivation to focus our model over such communities modeling [63]. In [64] authors found that social network structure and propagation network have critical features allowing highly accurate fake news detection. Therefore poster's social network context and their propagation network/community are also exploited in our investigations.

Many social psychology-related theories [49]–[54] are much supporting to our model foundations. Additionally, some important profile-based features highlighted in research studies highly motivate us to develop the proposed model.

III. RELATED WORK

In disseminating fake news, internet media has done more harm than mass media. YouTube, WhatsApp, Twitter, Sina Weibo, and Facebook are the five social media sites that generate the highest share of fake news [65]. It would be helpful to check the source of messages/posts and isolate phony messages from valid ones. The data that individuals share in web-based media is to a great extent affected by groups of friends, and the structure of connections on the web [66]. Therefore one can use different post-level and poster profile features successfully be used for COVID-19 related deception detection [67]. In [68], the authors show that users of social media share more counterfeit news on COVID-19 compared to substantial and fact-based news [69]. The authors also dissect data sourced from different social networking platforms. They analyze forged and groundless publications about COVID-19 and reveal that compared to Sina Weibo, Twitter has more unsubstantiated and false news [68]. There are seven prevailing subjects relevant to COVID-19 unverified news on media platforms: Politics,

terrorism, faith, health or fitness, religion, and miscellaneous cultures. Among all the mentioned subjects, the highest rank in the list is related to health counterfeit news revolving around medical prescriptions and clinical services [70]. The Study in [71] concerns the effect of COVID-19 related news shared over social media. It shows peaks of negative reactions and fear. Deep neural networks have been found to exhibit high accuracy, and precision in classification for fake news [72]. Deep diffusive neural networks, developed for fake news detection, which rely on Recurrent Neural Network (RNN), have gained popularity as the model exhibits good performance by investigating associations between news stories, their subjects/topics, and their writers/spreaders [73]. In [66], the author implements a framework using a combination of Convolution Neural Networks (CNN) and Long-Short-Term Memory (LSTM), a Recurrent Neural Network model, to identify and classify fake news messages from Twitter posts.

IV. COMPOSITION OF COVID-19 DATASET

We took 75 news articles related to COVID-19 (35 False and 40 True) from the Poynter [74], Politifact, and Lead Stories websites for this study. These news pieces are categorized into various types of disinformation such as false news, fake news, incomplete, false news, mislead, and so on, by specific websites. All these fake news about COVID-19 labeled as false, mostly false, pants on fire, misleading, and inaccurate by the fact-checking websites are considered fake news in our study. However, all the items of news labeled true or mostly true by fact-checking websites are labeled as True news in our study.

A. GETTING TWEET INFORMATION USING HOAXY

The labeled items of news collected from fact-checking websites are provided to Hoaxy [75]. It is an API to help users extracting fact-checking-based Twitter data. Out of two options, i.e., tweet search and article search, article search is utilized in the data extraction process. All the articles which matched according to keyword searching are then used to extract data for specific news. The most relevant articles are then selected, and their spread is visualized on Twitter. Hoaxy shares little details of the tweet and retweets concerning Twitter data sharing policy. It includes Tweet Ids, Retweet Ids, publishing DateTime, Retweet DateTime, Author user id, Retweeter user id, and bot scores of each user. The graph extracted by Hoaxy provides a tree-like network structure with a level one hierarchy.

B. CREATING FOLLOWERS AND FRIENDS RELATIONSHIPS USING TWEETPY

Tweepy [76] provides access to extract the Twitter data. For further experimentation and exploratory analysis, Tweepy is used for the remaining data extraction. We extract profile features and a list of followers and followings of all tweet and retweet users. The extracted data is not only helpful in our experiments, but it also facilitates analyzing the rate of

propagation, influence ratio, and the epidemic spread analysis based on the SIR [77]–[81] model. It is essential to mention that Hoaxy extracted data does not provide the users' relations. Therefore, Tweepy is used to build user-related relations as directed edges using followers and the following information.

For each extracted news from Hoaxy, we get three user relationships (edges) between each other, exploiting followers and the following information.

- 1) A—A Relationship (edge) between two tweet authors
- 2) A—R Relationship (edge) between authors & retweeters
- 3) R—R Relationship (edge) between two retweeters

The above news-related community network with posters and re-posters of news is created with the help of user relationships (edges). We build a directed graph and carry out an exploratory social network analysis based on the network attributes. Many features have been explored to shortlist the initial set of complex network measures (see section V-C). We calculated node (propagators) and network/community level features for our further model construction.

C. CONSTRUCTION OF TRUE AND FAKE NEWS COMMUNITIES

The true news networks and fake news networks have been constructed using the above-extracted dataset. The relationship between Tweeter and Re-tweeter provides the edge list for network construction. The directed networks for both true news and fake news articles are loaded using Python's NetworkX library. Each of the 75 news items contains a CSV file with all the relevant features/attributes. The same number of graph files are generated using Cytoscape, showing the links between the users who tweeted/retweeted particular news. Sample snapshots of two such graphs showing real and fake news item community structure from our dataset are presented in Appendix A.

V. METHODOLOGY

The section discuss the methodology of the research in detail.

A. PROPOSED FRAMEWORK

The framework shown in Figure 2 is implemented at two levels of each news propagator's community: at each poster/re-tweeter level (node level) and a whole community level (network level). For both levels of implementation, we implement user profile-based features and news community/network-related features. These features are then used to train a variety of machine learning and deep learning models. Few complex network parameters have been selected and proposed for COVID-19 related real and fake news article's community networks. These features extract distinct structural patterns found in both communities of COVID-19. Each news item has a network of all users involved in news either through posting or re-posting. News community is constructed by connecting each such user through followers and friends directed connections as well. Our approach to this problem is exploring and analyzing the dominant complex

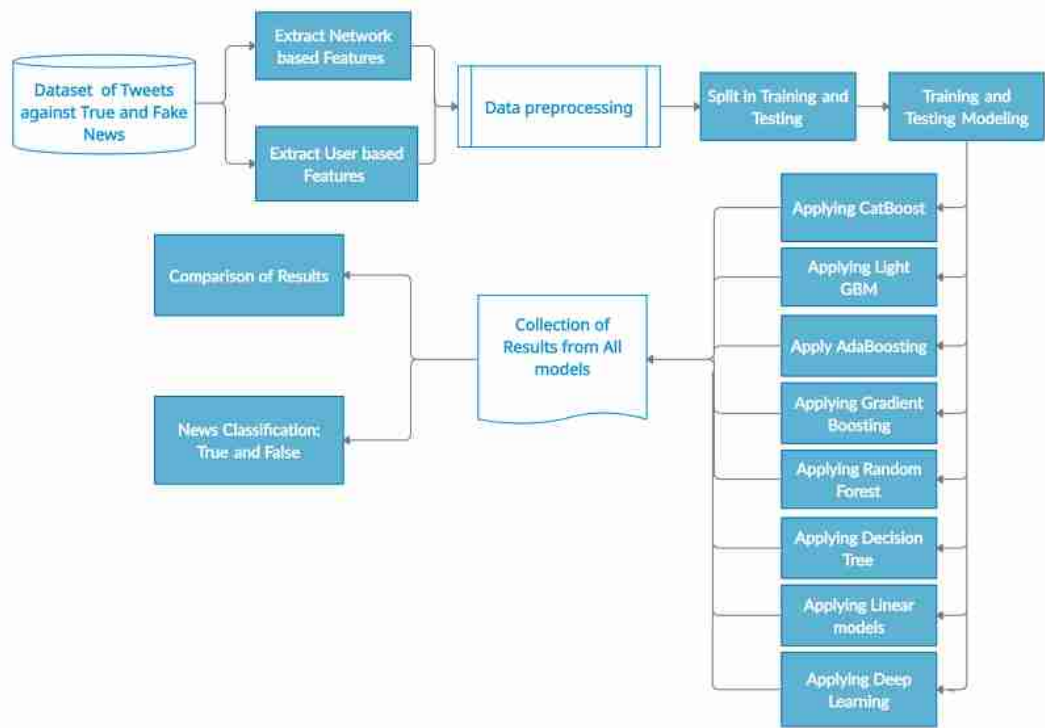


FIGURE 2. High-level COVID-19 Fake News Classification System’s Framework implemented at both levels of news propagators community (fake news and true news propagators’ community). These two levels are node level and whole community level. Two different groups of features, i.e., user profile-based and network-based features, are used to train various machine learning and deep learning models. Four approaches composed of these two groups of features and two levels are examined. The System has different processing stages. These stages are dataset construction, feature extraction, data pre-processing, constructing training and testing data splits, feature analysis and model selection, training ml and deep learning models, testing and evaluation, predicting news labels.

network/social network-related features and user profile features and then combine them to train different machine learning and deep learning models.

B. DATASET

A total of 75 news (35 Fake and 40 True News) about COVID-19 have been extracted. We managed to extract 33,248 tweets which were tweeted by 18,940 users. Table 1 shows the breakdown of news and tweets. It is worth mentioning that the community (only authors are considered) of fake news users and real news users only marginally overlap with only 1% users. All the other users are either fully involved in fake information dissemination or real information dissemination. Almost half of the users are engaged in retweeting/re-posting both types of news. These users are not authors/posters.

C. PROPOSED FEATURES

The proposed set of features includes complex network measures/Network-based features and user profile-based features. We investigate various machine learning and deep learning models to classify news articles as fake and true based on network and user profile-based features.

1) NETWORK BASED FEATURE

A complex network is a graph with non-trivial topological features that do not occur in simple networks. We compute

TABLE 1. Statistics for data involved in machine learning and deep learning. It presents three aspects of COVID-19’s dataset. 1. Total news articles considered. 2. Total tweets corresponding to news articles. 3. How many users involved.

Total News	Fake News	True News	
75	35	40	
Total Tweets	Fake Tweets	True Tweets	
33,248	13,990	19,258	
Total Users	Users in Fake	Users in True	In Both
18,940	10,106	8,834	83

important complex network properties related to each news article of the COVID-19 dataset. These features belong to a variety of complex network measures categories. They are based on neighborhood, path, community, centrality, degree, bridge, etc. They capture distinct structural connectivity patterns. The proposed list of features follows:

1. Number of nodes,
2. Number of edges,
3. Avg degree,
4. Avg shortest path length,
5. Max degree centrality,
6. Max betweenness centrality,
7. Average clustering coefficient,
8. Number of communities,
9. Max closeness centrality,

TABLE 2. Initial feature selection through correlation and value distribution. Only features not selected in columns A (mean value distribution) and B (correlation analysis) are dropped. Three out of twenty features are dropped: Max Degree Centrality, Density, and User Verified. Note: * please note that no's from 1 to 20 written in two columns of correlation analysis section are S# of correlated features.

S#	Features	Mean Fake	Mean True	Selected (A)	Correlation analysis*	Selected (B)*	Decision (A)OR(B)
1	number of nodes	321	243	Yes	2	No (Removed by 2)	Yes
2	number of edges	4,117	1,978	Yes	1,3	Yes	Yes
3	avg degree	18.03	10.39	Yes	5,2,1	No (Removed by 5)	Yes
4	max degree centrality	0.62	0.51	No	11,10	No (Removed by 11,10)	No
5	max betweenness centrality	0.34	0.32	No	2,10,11	Yes	Yes
6	avg shortest path length	2.00	1.81	Yes		Yes(No Corr)	Yes
7	avg clustering coefficient	0.42	0.28	Yes		Yes(No Corr)	Yes
8	density	0.07	0.08	No	2	No (Removed by 2)	No
9	max eigenvector centrality	0.32	0.37	Yes	11	No(Removed by 11)	Yes
10	max closeness centrality	0.68	0.59	Yes	3,4	Yes	Yes
11	max page rank	0.06	0.08	Yes	4,9	Yes	Yes
12	no of Communities	7.86	9.18	Yes		Yes(No Corr)	Yes
13	user botscore	0.34	0.25	Yes		Yes(No Corr)	Yes
14	user followers count	1,916,147	3,160,935	Yes	16,19	Yes	Yes
15	user friends count	16,324	16,709	No		Yes	Yes
16	user listed count	4,066	7,314	Yes	14,19	Yes	Yes
17	user acc age	16.11	16.80	No		Yes(No Corr)	Yes
18	user favourites count	80,311	75,487	Yes	20	No(Removed by 20)	Yes
19	user verified	0.20	0.23	No	14,16	No (Removed by 14,16)	No
20	user statuses count	134,526	131,029	Yes	18	Yes	Yes

10. Max eigenvector centrality,
11. Density,
12. Max page rank.

2) USER BASED PROFILE FEATURES

We also propose many user profile-based features to categorize the news articles as real or fake. Therefore, we extracted the following user profile-based features of each user present in a specific news community. Users either posting or re-posting the news article were present in the graph related to that particular news community. The features of such users are:

1. User Bot Score,
2. Account Age,
3. Is verified,
4. Followers count,
5. Friends count,
6. Favorite count,
7. Status count,
8. Listed count.

D. DATA PREPARATION

Each news is posted and re-posted by many users, forming news article's propagators' community as a network with many nodes. The relationships between these Tweeters and Re-tweeters are provided in the form of the edge list for network construction (see section IV). The directed networks are constructed for true and fake news articles and loaded using Python's NetworkX library. All the proposed complex network measures are also computed using the same library. Features are calculated at two levels, 1.) Each node level. 2.) Each article's network/community level. A complete sample record

of the feature values for the community level is shown in table 2 (see first four columns of the table). A single row contains all feature values against each news item when features are computed at the community level, whereas multiple rows when features are computed at the individual node level. Each of the 75 news items contains a CSV file that has all the relevant features/attributes proposed. Finally, a single CSV is generated for each level through their individual CSVs. The Pandas' DataFrame [82] for Python is constructed through a CSV file, and then machine learning (ML) and deep learning (DL) models are trained and tested. These ML and DL models are also available through Python's respective libraries.

VI. FEATURE ANALYSIS

We present the feature analysis through; 1. Mean value distribution. 2. Correlations. and 3. Empirical Cumulative Distribution Function (ECDF) plots. Details of these feature analysis basis are discussed below:

1. Mean Value Distribution: Real and fake news have different characteristics that can be important while analyzing a tweet. Figure 6 shows the difference in the distribution of network features of both sets of news through ECDF plots. Similarly, the analysis of the mean values for all features in both items of news labels is presented in table 2. For example, the number of nodes, edges, and avg degree values in fake is very high compared to real news. The number of communities is higher in true news. The profile-based features analysis shows that some undeniable trends are highly dominating. Bot scores (bots are considered malicious profiles only) are high in fake news, whereas listed count values are much higher in real news. In column A of table 2 the decision based on mean value distribution is presented.

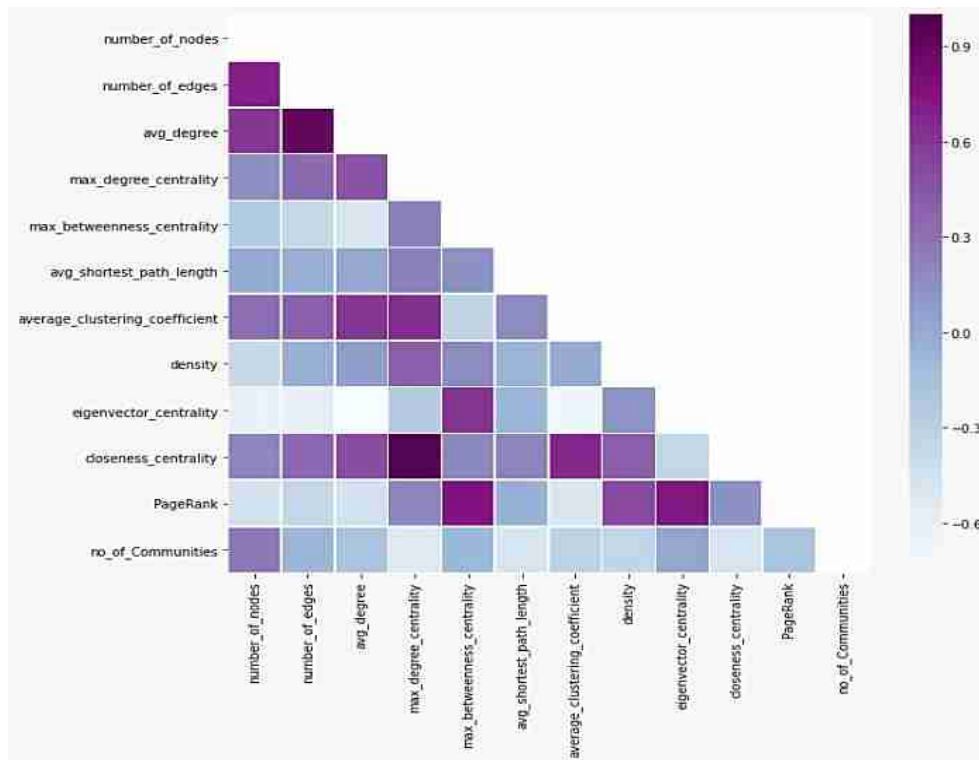


FIGURE 3. Network-based Features Correlation within Real and Fake News Articles Data: The correlation matrix is computed that determines the degree of association between features. The higher correlation value is shown by darker shades, whereas a lighter color means a weaker correlation. Therefore correlations between different features are closely analyzed for performing feature analysis. If one or more features are correlated with some feature then only that feature is selected instead of other correlated features. It is observed that following few features are highly correlated i.e., Avg. Degree vs. Number of Edges and Max Closeness Centrality vs. Max Degree Centrality. The following few features are moderately correlated i.e., Number of Edges vs. Number of Nodes, Max Page Rank vs. Max Eigenvector Centrality, Max Page Rank vs. Max Betweenness Centrality and, Avg Degree vs. Number of Nodes [83], [84].

Finally, it appears that only 17 features allow discriminating both types of news. Two network-based features (Max. degree centrality and Density) are dropped, and one user profile-based feature (Is Verified) is eliminated from the selected list (see table 2 for analysis summary).

2. Correlation Analysis: We measure the Pearson correlations between features to determine the degree of association between features for both categories of features. We also use the Seaborn library in python to visualize the correlations. Figure 3 presents network-based features correlations, whereas figure 4, presents the correlation matrix between the user-based profile features. A high correlation value is in darker shades, whereas a lighter color means a weaker correlation. Based on the correlation we selected our best features [83], [84] (see column B in table 2) from both network-based features and user-based profile features. In correlation analysis, features at S.No: 1,3,4, 8, and 19 in table 2 are highly correlated to other features, therefore, recommended to be replaced with their correlated ones.

Initial feature selection decision through correlation and value distribution is shown in decision column (see the last column) of Table 2. All selected features are selected in column A (for mean value distribution) or column B (for correlation analysis). Therefore, three features are dropped

because they are not selected in the mean value distribution and correlation analysis. A total of 17 features are selected to train the models for better performance.

3. Exploratory Data Analysis- ECDF Plot: The Python library offers the ECDF class to fit an empirical cumulative distribution function and compute the cumulative probabilities for data attribute selection. A method for efficiently plotting and applying percentile thresholds for data exploration to an empirical cumulative distribution function (ECDF). Figure 5 presents ECDF plots for our user-based profile features, and figure 6 shows ECDF plots for network-based features for both True and False news labels.

The majority of previously shortlisted, through mean value distribution and correlation (total 17) features, are also discriminating in ECDF plots. It also appears that the network features are slightly more dominant than the user profile-based features.

VII. CLASSIFICATION APPROACHES

We conduct a complete exploratory study by implementing both categories of features. First, we explore the community level, where each news article is associated with its community comprising all its posters and re-tweeters with

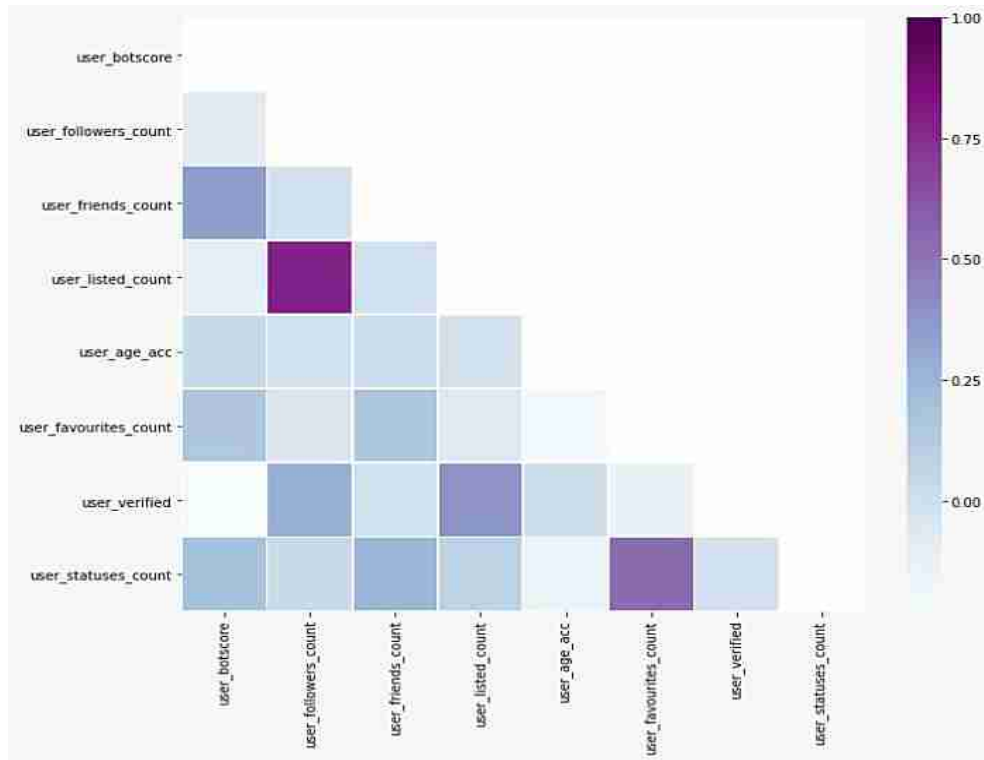


FIGURE 4. User-based Features Correlation within Real and Fake News Articles Data: The correlation matrix that determines the degree of association between features is computed. The higher correlation value is shown by darker shades, whereas a lighter color means a weaker correlation. Therefore correlations between different features are closely analyzed for performing feature analysis. If one or more features are correlated with some feature then only that feature is selected instead of other correlated features. It appears that the following features are highly correlated, i.e., Listed Count vs. Followers Count. The following features are moderately correlated i.e., Status Count vs. Favorite Count [83], [84].

their interconnections through follower-followings. Second, we consider the individual's/node level, where each node represents a poster or a re-tweeter. We investigate the following four approaches through different combinations of levels and categories of features. As two of them are competing and, therefore, further exploration is conducted. We use the features from the feature analysis phase (a total of 17 from both feature categories).

Machine Learning (ML) algorithms from different classes are applied: SVM, Linear regression, Random Forest, AdaBoost classifier, Decision tree, Ensemble learning, Catboost, Light GBM, Gradient boosting classifier, KNN, and Naive Bayes. All these sets of models are evaluated under the following four approaches:

A. APPROACH I: NODE LEVEL-USER FEATURES

In this approach, only user profile-based features are implemented at each participating node within the community of news articles, which are either posters or re-posters. The same set of ML models are trained and tested. The results of experiments related to Approach I are shown in table 3. It appears that this approach performs poorly with an AUC score of 0.70 by EVC and KNN.

TABLE 3. Approach I: Result of User based features at node level.

	Accuracy	ROC-AUC Score
<i>CATBoost</i>	74%	0.63
<i>LightGbm</i>	74%	0.63
<i>AdaBoosting</i>	74%	0.63
<i>GradientBoosting</i>	74%	0.64
<i>EVC</i>	69%	0.70
<i>DecisionTree</i>	72%	0.60
<i>RandomForest</i>	73%	0.63
<i>NaveBayes</i>	69%	0.61
<i>Logistic</i>	70%	0.59
<i>SVM</i>	69%	0.58
<i>KNN</i>	77%	0.70

B. APPROACH II: NODE LEVEL-USER FEATURES & NETWORK FEATURES

In this approach, we apply all network features and user-based features at each participating node within the community of news articles. Which are either posters or re-posters. The same set of ML models are trained and tested. Results of the experiments related to Approach II are shown in table 4. It shows that this approach performs better with an AUC score of 0.98 by CATBoost and LightGbm.

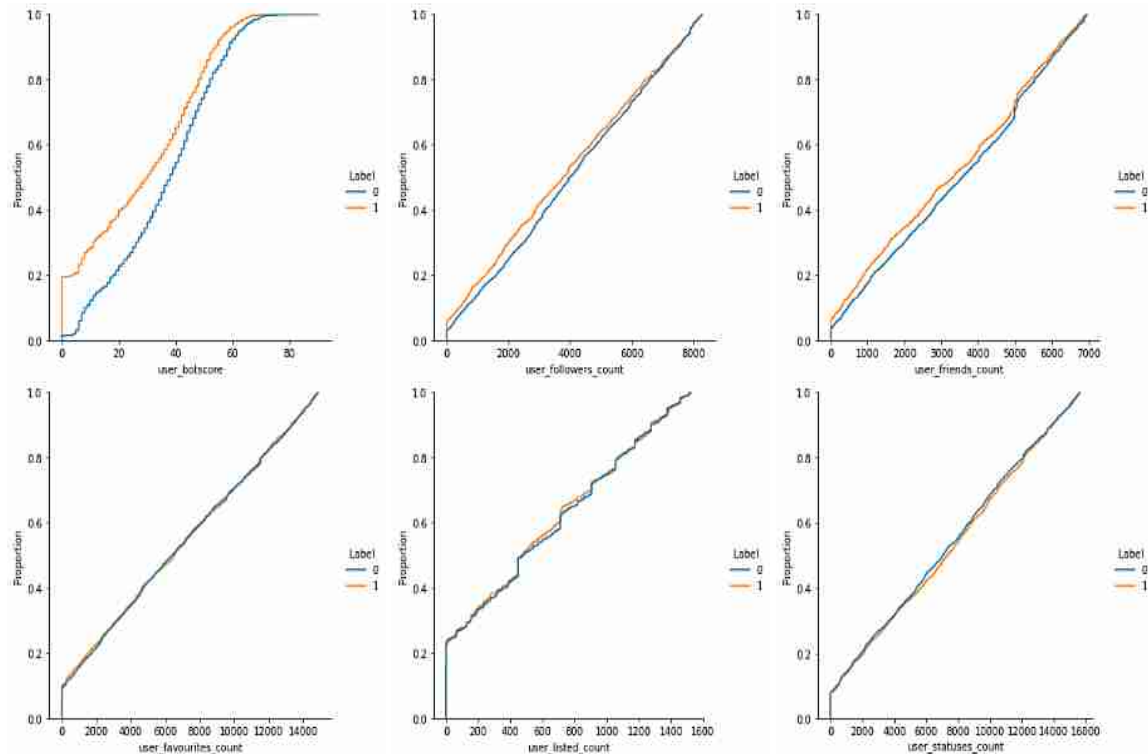


FIGURE 5. ECDF-based Feature Analysis of User Profile Features. Plots of Empirical Cumulative Distribution Function (ECDF) for a few user profile-based features against news labels. There are six plots presented in two rows. The first row of features is more discriminating: user bot score, user followers count, and user-friend count. Features in the second row are less discriminatory: user favorite count, user listed count, and user status count. It shows that these features could be strong predictors for the deception detection problem. Comparison of both groups of features named user profile-based features and network-based features show that network-based features are more discriminating than user profile-based features.

TABLE 4. Approach II: Results of combining user and network features at node level.

	Accuracy	ROC-AUC Score
<i>CATBoost</i>	98%	0.98
<i>LightGbm</i>	97%	0.98
<i>AdaBoosting</i>	91%	0.92
<i>GradientBoosting</i>	97%	0.97
<i>EVC</i>	80%	0.82
<i>DecisionTree</i>	80%	0.82
<i>RandomForest</i>	86%	0.81
<i>NaveBayes</i>	79%	0.78
<i>Logistic</i>	81%	0.78
<i>SVM</i>	72%	0.64
<i>KNN</i>	76%	0.70

C. APPROACH III: COMMUNITY LEVEL- MEDIAN OF USER FEATURES & AGGREGATED NETWORK FEATURES

In this approach, we apply all aggregated network features and aggregated (median) user-based features at the community of each news article. The same set of ML models are trained and tested. Results of the experiments related to Approach III are reported in table 5. It appears that this approach ranks second with an AUC score of 0.90 by CATBoost, AdaBoost, Grad. Boosting, Random Forest, NB, EVC, and KNN.

TABLE 5. Approach III: Result of median users features combined with aggregated network features at community level.

	Accuracy	ROC-AUC Score
<i>CATBoost</i>	87%	0.90
<i>LightGbm</i>	75%	0.83
<i>AdaBoosting</i>	87%	0.90
<i>GradientBoosting</i>	86%	0.90
<i>EVC</i>	87%	0.90
<i>DecisionTree</i>	75%	0.83
<i>RandomForest</i>	87%	0.90
<i>NaveBayes</i>	87%	0.90
<i>Logistic</i>	87%	0.75
<i>SVM</i>	62%	0.70
<i>KNN</i>	87%	0.90

D. APPROACH IV: COMMUNITY LEVEL- AGGREGATED NETWORK FEATURES

In this approach, we apply all aggregated network features to the community of each news article. The same set of ML models are trained and tested. Results of experiments related to Approach IV are shown in table 6. One can conclude that the results of this approach outperform Approach I and III with an AUC score of 0.92 by Light GBM, Grad. Boosting, and KNN.

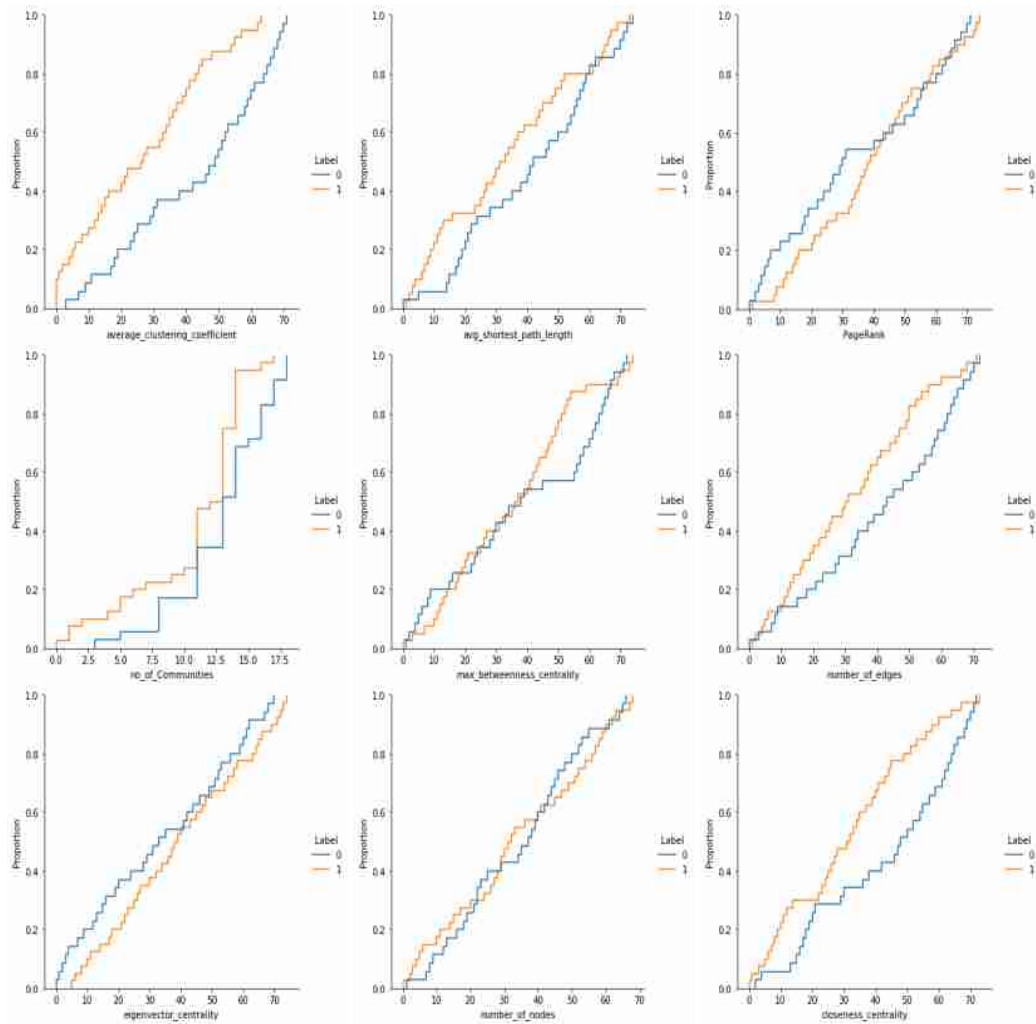


FIGURE 6. ECDF-based Feature Analysis of Network Features: Plots of Empirical Cumulative Distribution Function (ECDF), demonstrating completely distinct distributions of few discriminating Network features (There are a total of nine plots, three in each row, plotted in this figure. In Row 1: Avg. Clustering Coefficient, Avg. Shortest Path Length, Max. Page Rank, in Row 2: Number of Communities, Max. Betweenness Centrality, Number of Edges, in Row 3: Max. Eigenvector Centrality, Number of Nodes, Max. Closeness Centrality is plotted) against news labels. It shows that these features could be strong predictors for the deception detection problem. Comparison of both groups of features named user profile-based features and network-based features show that network-based features are more discriminating than user profile-based features.

E. SUMMARY OF APPROACHES COMPARISONS

The node-based combined feature approach called Approach II performs the best. The community-based approach using only aggregated network features named Approach IV is the second-best performer. In the following, we only consider these two approaches for our further considerations, e.g., best model feature selection, etc.

The following essential findings are helpful to interpret the outcome of the two successful approaches. First of all, network-based features are more discriminating than user profile-based features (see figure 6). Second, considering the particular case of COVID-19, which is rarely understood and authenticity of a message is known to the author/producer only but challenging to identify for the receiver and re-tweeter. Therefore, half of the users are involved in retweeting/forwarding both types of news, fake

news and true news. It creates an overlap between both communities. The remaining half are either pure, true news propagators or fake news propagators. They are the authors/posters of the specific news items, and their feature values are different. In a community-based approach where these two separate groups of users are combined, their user profile-based features are aggregated, reducing their discriminating ability. These aggregated features produced many similar values for both communities affecting the performance of ML models. Therefore Approach IV, based on network-based features, gives better results (see table 6) than Approach III, which combines user profile-based features with network-based features over community level (see table 5). Approach II produces the best results with features at node level (see table 4).

We also trained Deep Learning models named RNN and CNN with Approach II and IV configurations, but results

TABLE 6. Approach IV: Showing the result of Aggregated Network based features at community level.

	Accuracy	ROC-AUC Score
<i>CATBoost</i>	75%	0.80
<i>LightGbm</i>	87%	0.92
<i>AdaBoosting</i>	87%	0.90
<i>GradientBoosting</i>	87%	0.92
<i>EVC</i>	75%	0.80
<i>DecisionTree</i>	75%	0.80
<i>RandomForest</i>	75%	0.80
<i>NaveBayes</i>	75%	0.85
<i>Logistic</i>	75%	0.80
<i>SVM</i>	75%	0.81
<i>KNN</i>	87%	0.92

TABLE 7. Deep Learning Models Results for Approach II & IV Configurations.

Models	Approach II: Acc	Approach IV: Acc
<i>RNN</i>	98%	90%
<i>CNN</i>	97%	89%

produced (see table 7, for results of deep learning models) are almost similar to our best performing machine learning models discussed and presented earlier (tables 3,4,5,6).

VIII. FAKE DETECTION MODELING

This section discusses candidate Machine Learning and Deep Learning models used for solving the required supervised classification problem.

A. MACHINE LEARNING CLASSIFICATION MODELS

We test the proposed set of 17 features classified in two categories under four approaches. The same learning algorithms are used in each approach, and the best approach with its best performing features is chosen. This set of machine learning models used in all experiments are discussed in this section.

We apply algorithms from different classes, including linear, non-linear, tree-based, non-tree based, non-parametric, probabilistic, large margin classifiers, and Ensemble (boosting, election mechanism) models. For machine learning, the algorithms include SVM, Linear Regression, Random forest, AdaBoostClassifier, Decision tree, Ensemble learning, Catboost, Light GBM, Gradient Boosting Classifier, KNN, and Naive Bayes. All these set of models are evaluated under four approaches and their results are reported in tables: 3, 4, 5, and 6, respectively.

The dependent variable is found to be binary when we are dealing with a Logistic Regression classifier. The main application of Logistic Regression is to predict and calculate success probability. Using Logistic Regression classifier, the random state we use is one and set Broyden–Fletcher–Goldfarb–Shanno (bfgs) algorithm as a solver. A hyperplane is created by a Support Vector Machine that distinguishes the data points into two distinct classes. The training data is mapped into kernel space by SVM [85]. Out of many possibilities, we use the linear kernel for the SVM classifier. A requirement for Naïve Bayes classifiers, which are quite scalable, is that the relationship between numbers of

parameters to the number of features is linear. The purpose of a k-Nearest Neighbors' algorithm is to store all accessible cases and classifying new cases on the criteria of a distance function. Random Forest is a meta estimator that fits several decision tree classifiers on various sub-samples of the dataset. Furthermore, the predictive accuracy is improved, and overfitting is controlled using averaging. The maximum depth of the tree is one, and reinforced measures are "Gini." The trees that conduct the classification of instances on the criteria of feature values are Decision Trees. A feature, which can be classified in an instance, is represented by each node of the Decision Tree, and the value assumed by the node is represented by a branch. The EVC classifier is used along with three estimators (Logistic Regression, Decision Tree, and SVM) to predict the class. Many weak learning models are combined, forming a strong predictive model. This is achieved by a group of machine learning algorithms called Gradient Boosting classifiers. Fifty estimators are utilized in Gradient Boosting classifiers using a depth of 2. LightGBM can handle huge amounts of data with ease. This algorithm uses the Gradient Boosting framework, which in turn uses a tree-based algorithm allowing the tree to grow vertically by having a default learning rate of 0.1. CATBoost is a classifier that is rated first and does not require tuning of the parameter. It is an extensible GPU version. This classifier helps in the reduction of overfitting and increase the accuracy of the model.

As per the framework, we divide the dataset into two phases, the training phase: which learns how to classify data (constructs a model) based on the training set. Testing phase: in testing, we check those unseen data examples that do not come under the dataset we have considered for training and now the trained model predicts. After the prediction, we get the class labels used for performance evaluation. Different measure are used for evaluations.

B. EVALUATION METRICS

Since the dataset is slightly imbalanced, we use the AUC curve to validate the performance of machine learning algorithms. AUC is statistically more consistent and more discriminating than accuracy [86]. Other evaluation metrics like accuracy, precision, recall, and F1 scores are also calculated but only accuracy and AUC are used for the results [87]. Following are the formulae for these evaluation measures.

$$\begin{aligned}
 \text{Accuracy} &= \frac{TP + TN}{TP + FP + TN + FN} \\
 \text{Precision} &= \frac{TP}{TP + FP} \\
 \text{Recall} &= \frac{TP}{TP + FN} \\
 \text{F1 - Measure} &= \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}
 \end{aligned}$$

Receiver Operating Characteristic (ROC) curve is a plot that shows the performance of a classification model with two parameters TPR vs. FPR at all classification thresholds.

TPR and FPR are provided as following. Area under the ROC Curve (AUC) measures the entire two-dimensional area underneath the entire ROC curve.

$$\text{TruePositiveRate}(TPR) = \frac{TP}{TP + FN}$$

$$TPR = \text{Sensitivity} = \text{Recall}$$

$$\text{FalsePositiveRate}(FPR) = \frac{FP}{FP + TN}$$

where TP: True Positive, TN: True Negative, FP: False Positive, and FN: False Negative. These all are defined in the confusion matrix.

C. DEEP LEARNING CLASSIFICATION MODELS

After testing the data with several conventional supervised methods, we test the deep learning model as it always gives an efficient output. The two most famous deep learning methods Recurrent Neural Network (RNN) and Convolution Neural Network (CNN) are implemented to see how well our data fit into the model. Such algorithms are ideal for various classifications based on datasets. Keras provides easy and reliable high-level APIs and follows the best methods to reduce the cognitive burden for users, using Keras as a neural network library. Unlike normal neural networks, RNNs rely on prior output knowledge to predict upcoming information. This role becomes extremely helpful when combined with sequential knowledge. An LSTM network is a recurrent neural network with LSTM cell blocks instead of the usual neural network layers. Regarding the CNN model, the architectural structure is available in figure 7 with a given number of parameters. Most of the parameters come from the fully connected component, comparing the number of parameters in the function learning part of the network and the fully connected part of the network. In terms of computing costs, CNN is very effective. Parameters are selected to train both models RNN and CNN. Grid search approach is used to optimize the hyper-parameters and optimal values are derived (see figures 8, 10). Parameters are set as follows for both CNN and RNN: batch size=45, epochs=10, learning rate=0.1, loss = 'mse', activation function = 'ReLU', optimizer='adam' and dropout regularization rate which is set at 20%. In CNN, we include the dropout layer after the embedding layer. Using the dropout layer we can generalize the performance of our model, then the 1D CNN layer is added. We further add a max-pooling layer to reduce the dimensionality of the input layer while preserving the depth and avoid over-fitting of the training data. Then converts the pooled feature map to a single column that is passed to the fully connected layer. In the end, the dense layer add the fully connected layer to the neural network.

In the RNN model, we add an embedding layer after the input layer. ReLU function is used as an activation function on hidden layers to activate the neurons, dropout layer is added to generalize the model's performance. Batch normalization is used for re-centering and re-scaling the input layer through normalization. It makes the neural network faster

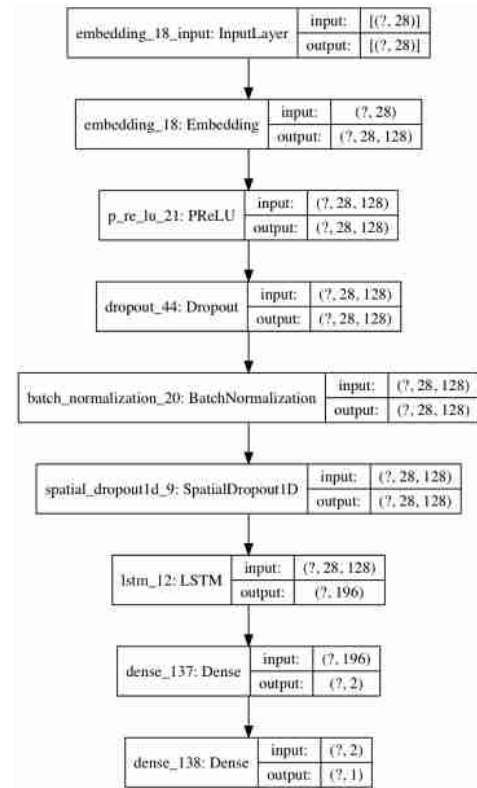


FIGURE 7. Architecture diagram of RNN: In this RNN, an embedding layer is added after the input layer. Embeddings are used for purposes such as finding nearest neighbors, and low-dimensional representations are also learned. ReLU layer is also added, where ReLU function is used as an activation function on hidden layers to activate the neurons. Dropout layer is added to generalize the performance of the model. Batch normalization is used to train very deep neural networks, making the neural network faster and more stable. It reduces the number of training epochs. The long short-term memory (LSTM) layer is added. Unlike standard feedforward neural networks, LSTM has feedback connections. It is best for sequenced data processing. Regular dropout is unable to regularize the activations and decrease the learning rate. Therefore, the spatial dropout layer is used before LSTM. It drops entire 1D feature maps instead of individual elements. Finally, the dense layers are added for the fully connected layer and label predictions.

and more stable and reduces the number of epoch to train the deep neural network. In this model, we use the LSTM layer. Regular dropout cannot regularize the activations and decrease the learning rate so we use the spatial dropout layer. In the end, the dense layer add the fully connected layer to the neural network. The main objective is to significantly increase the accuracy of the model prediction. The architectural diagram for RNN is shown in figure 7 and CNN is shown in figure 8. The summary/description of layers configuration of the RNN and CNN models are shown in figure 9 and figure 10 respectively.

D. MODEL SELECTION

We consider the best two shortlisted approaches for further explorations. These are Approach II (network-based and user profile-based features) and IV (network features). One needs to find the appropriate best-performing features for

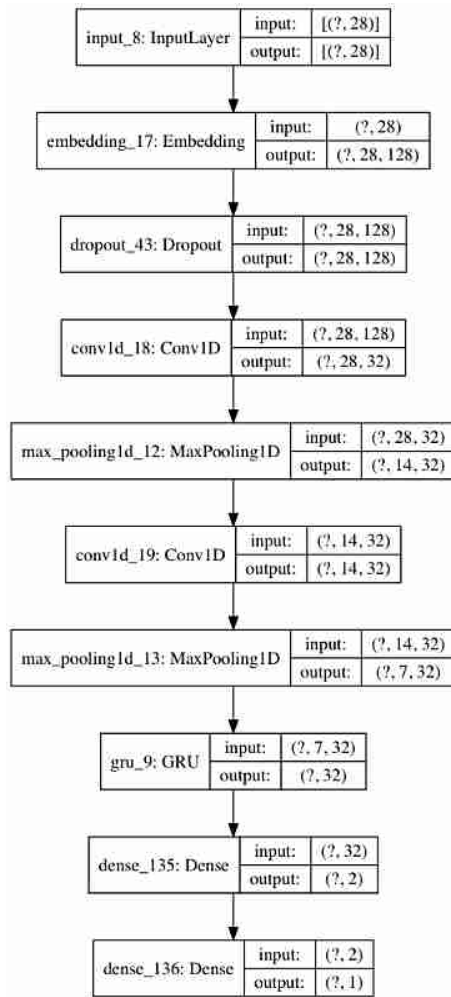


FIGURE 8. Architecture diagram of CNN: The normal dropout layer is used after the input and embedding layer in this CNN. Embeddings are used for purposes such as finding nearest neighbors, and low-dimensional representations are also learned. The dropout layer generalizes the performance of our model then the 1D convolution layer is added to extract the high-level features. To highlight the most present features, a max-pooling layer is added in CNN to reduce the dimensionality of the input layer while preserving the depth and avoid over-fitting of the training data. The same 1D Conv and max-pooling is added again. Then, the pooled feature map is converted to a single column passed to the GRU (gated recurrent unit) layer. The GRU networks replace CNN fully connected layers to transform the classification task into a sequence task. Finally, the dense layers are added for fully connected layer and label predictions.

each approach. These features are then used to train different machine learning and deep learning models for better-optimized performance. Identification of top-performing models is the precursor to best feature selection. The best model selection is made in this section.

We determine the optimal set of hyper-parameters by testing the performance of our models on the training set for different parameter combinations. After tuning our models using an optimal set of hyper-parameters, the four best models are selected as best performing among the two selected approaches. These best models are CATBoost, LightGBM, AdaBoost, and Gradient Boosting.

Layer (type)	Output Shape	Param #
embedding_18 (Embedding)	(None, 28, 128)	6665472
p_re_lu_21 (PRELU)	(None, 28, 128)	3584
dropout_44 (Dropout)	(None, 28, 128)	0
batch_normalization_20 (Batch Normalization)	(None, 28, 128)	512
spatial_dropout1d_9 (Spatial Dropout)	(None, 28, 128)	0
lstm_12 (LSTM)	(None, 196)	254800
dense_137 (Dense)	(None, 2)	394
dense_138 (Dense)	(None, 1)	3
Total params: 6,924,765		
Trainable params: 6,924,509		
Non-trainable params: 256		

FIGURE 9. Summary of RNN model: In addition to the summary of the model's layer provided in this diagram following are the model's configurations. Grid search is used to optimize the hyper-parameters, and optimal values are derived. The hyper-parameters values are as follows: batch size=45, epochs=10, learning rate=0.1, loss = 'mse', activation function = 'ReLU', optimizer='adam' and dropout regularization rate which is set at 20%.

Layer (type)	Output Shape	Param #
input_9 (InputLayer)	[(None, 28)]	0
embedding_19 (Embedding)	(None, 28, 128)	6665472
dropout_45 (Dropout)	(None, 28, 128)	0
conv1d_20 (Conv1D)	(None, 28, 32)	8224
max_pooling1d_14 (MaxPooling1D)	(None, 14, 32)	0
conv1d_21 (Conv1D)	(None, 14, 32)	2080
max_pooling1d_15 (MaxPooling1D)	(None, 7, 32)	0
gru_10 (GRU)	(None, 32)	6336
dense_139 (Dense)	(None, 2)	66
dense_140 (Dense)	(None, 1)	3
Total params: 6,682,181		
Trainable params: 6,682,181		
Non-trainable params: 0		

FIGURE 10. Summary of CNN model: In addition to the summary of the model's layer provided in this diagram following are the model's configurations. Grid search is used to optimize the hyper-parameters, and optimal values are derived. The hyper-parameters values are as follows: batch size=45, epochs=10, learning rate=0.1, loss = 'mse', activation function = 'ReLU', optimizer='adam' and dropout regularization rate which is set at 20%.

E. FEATURE SELECTION

In this section, we analyze the importance of features using the best-performing machine learning models. It is important to know what is the final set of features that are highly discriminating. Indeed, using the only best set of features, models can be tuned for much better performance. We use Mean Decrease Impurity (MDI) for feature importance and feature engineering. To evaluate, we use the feature importance module in the sci-kit learn library. This module calculates the relative weight of each feature in the model, and it sums up to 1. We had 20 features which were reduced to 17 after feature analysis. These 17 features comprising

both network and user-based features are trained on machine learning models, accordingly under specified approaches. Among all ML models, only the top four are selected with Approach II and IV for best feature selection using Mean Decrease Impurity. Table 8 shows the top features of the best four performing models with respect to approaches. Remember that Approach II includes both types of features, whereas Approach IV only uses network-based features. Therefore, best features are with respect to their approach.

Following are few features that can be eliminated from the best features list because they are not found in the top essential feature list among any of the best models (favorite count, No of nodes, max page rank centrality). User profile-based feature named user account age is considered among selected feature list, as a particular case because it is selected at no 11 by two best models and therefore not shown in top 10 important feature list of table 8. The other two models selected max closeness centrality at no 11, which LightGBM already selects in the top 10 list. The remaining 14 features are the best features for model tuning and performance increase.

IX. BEST PERFORMING APPROACHES II AND IV

Following the best features selection reported in the previous subsection VIII-E, 14 features are retained (8 network-based and six user-based). These selected features are used for Approach II, while the eight network-based features are used for Approach IV to optimize the model's performance further. This section discusses the results of the best-performing approaches after a final tuning with these features.

A. MODEL RESULTS

We determine the optimal set of hyper-parameters by testing the models' performance on the training and test sets for different parameter combinations. Table 9 reports the values of the parameters associated with the best results for different algorithms using Approach IV. Table 10 show similar results for Approach II. These results are obtained through the implementation of the best features selected and discussed earlier in section VIII-E. One can observe that Gradient Boosting has the highest test accuracy score: 88%, with the best AUC-ROC score: 0.92 among all other competing models in Approach IV (see table 9). CAT Boost performs best in Approach II with a test accuracy score of 98.4% and a ROC-AUC score of 0.98 (see table 10). Figure 11 shows the testing accuracies of all machine learning and deep learning models following Approach II (the best approach).

We also explore Deep Learning models using best approaches II and IV. Results reported in table 7 are in the same vein. Approach IV relies on network-based features, and approach II uses user and network-based features to optimize the ML models. As Approach II outperforms its alternatives, we discuss in detail only the models in the Approach II context. Table 10 reports the results of the investigations using the 14 best features mentioned in section VIII-E. KNN and Naïve Bayes models exhibit respectively an accuracy of 76% and 79%. SVM and Logistic Regression models have 72.4%

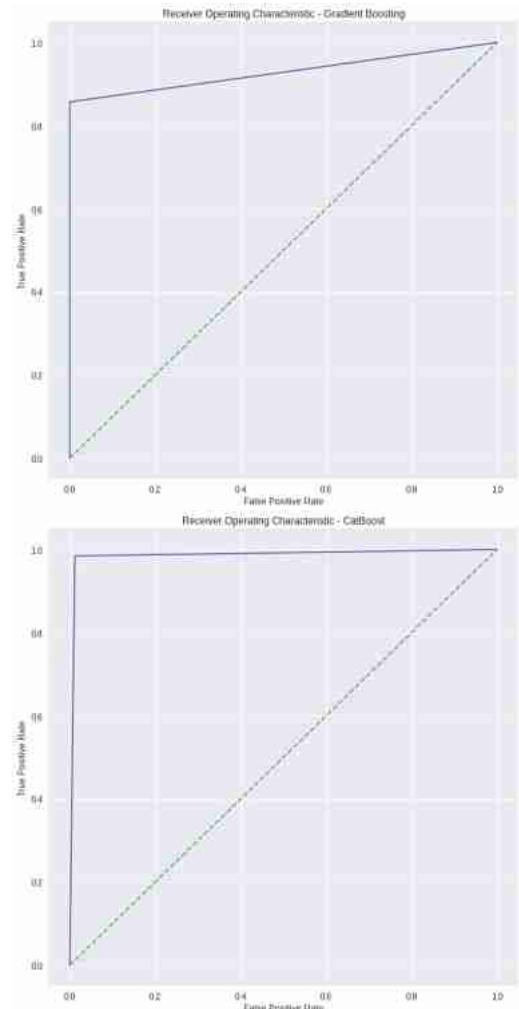


FIGURE 11. AUC-ROC Curves: There are four approaches, and the same set of eleven ML models are under test in each approach. The best performing ML model is named Gradient Boosting in Approach IV (which is second-best among all four strategies) with 92% accuracy. The top-performing ML model with 98% accuracy is CAT Boost in Approach II. It is the best score among the four approaches.

and 81% accuracy, respectively. Tests on Decision Tree and EVC models show that the performance is slightly optimized with an accuracy of 80% for both algorithms. A performance enhancement of 6% over the previous results is observed for Random Forest. was tested for observing any further enhancement. Ada boosting performs much better with an accuracy score of 91%. Tests with the most recent machine learning algorithms named (Gradient Boosting, LightGBM, and CATboost) have been conducted. These algorithms perform well, and one observes further enhancements. Indeed, the testing accuracy reaches 97.92%, 98.1%, and 98.4%, respectively for Gradient Boosting, LightGBM, and CATboost. Table 10 shows the overall comparison of machine learning algorithms for Approach II. CATboost presents the highest accuracy and ROC-AUC score. State-of-the-art deep learning techniques are also used. The accuracy achieved by deep learning algorithms like RNN and CNN is 98% and 97%, respectively. It is shown in figure 12 with the comparison of

TABLE 8. The best model features: Referring node level combined feature Approach II: Top 10 features across best four performing models. Likewise, community-based Network feature Approach IV: Top 7 essential features across same best performing models. Mean Decrease Impurity (MDI) is used for feature importance: Avg Degree, Avg Clustering coefficient, and max eigenvector centrality have the highest importance among all models. Max closeness centrality only appeared in LightGBM model list. Dominating user features are user listed count and user bot score, found in the best three models. Features like: favorite count, No of nodes, and max page rank centrality, could be eliminated because they are not in the top important feature list among any of best models.

Approach II (Top 10 Combined Features)				
S#	CatBoost	LightGBM	AdaBoost	Grad.Boosting
1	avg_degree	max_betweenness_centrality	avg_degree	avg_degree
2	avg_clustering_coefficient	avg_culustering_coefficient	avg_shortest_path_length	avg_culustering_coefficient
3	no_of_communities	max_eigenvector_centrality	max_eigenvector_centrality	max_eigenvector_centrality
4	max_eigenvector_centrality	avg_degree	no_of_communities	max_betweenness_centrality
5	max_betweenness_centrality	avg_shortest_path_length	user_bot_score	no_of_communities
6	no_of_edges	no_of_communities	user_followers_count	avg_shortest_path_length
7	avg_shortest_path_length	user_bot_score	user_listed_count	no_of_edges
8	user_friends_count	no_of_edges	user_status_count	user_bot_score
9	user_status_count	user_listed_count	max_betweenness_centrality	user_followers_count
10	user_listed_count	max_closeness_centrality	no_of_edges	user_friends_count
Approach IV (Top 7 Network Features)				
S#	CatBoost	LightGBM	AdaBoost	Grad.Boosting
1	avg_degree	max_betweenness_centrality	avg_degree	avg_degree
2	avg_clustering_coefficient	avg_culustering_coefficient	avg_shortest_path_length	avg_culustering_coefficient
3	no_of_communities	max_eigenvector_centrality	max_eigenvector_centrality	max_eigenvector_centrality
4	max_eigenvector_centrality	avg_degree	no_of_communities	max_betweenness_centrality
5	max_betweenness_centrality	avg_shortest_path_length	max_betweenness_centrality	no_of_communities
6	no_of_edges	no_of_communities	no_of_edges	avg_shortest_path_length
7	avg_shortest_path_length	no_of_edges	avg_culustering_coefficient	no_of_edges

TABLE 9. Detailed results of Approach IV models using the 8 best features set. Results with hyper-parameters and others. Performance change after implementing the best features. Gradient Boosting has the highest accuracy (88%) and AUC-ROC score (0.92).

Models	Parameters	Accuracy	ROC-AUC	Performance Status
CATBoost	random_seed=30, border_count=110, l2_leaf_reg=7, iterations=20, learning_rate=.8, depth=1.	75	0.8	No Change
Light Gbm	boosting_type='gbdt', learning_rate=0.05, max_depth=1, n_estimators=100, n_jobs=-1.	87	0.92	No Change
AdaBoosting	algorithm='SAMME.R', base_estimator=None, learning_rate=0.7, n_estimators=10, random_state=1.	87	0.90	No Change
GradientBoosting	lr=0.1, max_depth=2, random state=0, n_estimators=50,	88	0.92	Both Scores Improved
EVC	estimators=3.	87	0.92	Both Scores Improved
Decision Tree	max_depth=2, min_Sample_Split=2. n_estimators=50, random_state=0,	75	0.80	No Change
Random Forest	max_depth=2	75	0.80	No Change
Naïve Bayes	alpha=1	75	0.85	No Change
Logistic	Random state = 1, solver= bfgs, intercept_scaling=1	75	0.80	No Change
SVM	kernal = linear svc()	75	0.81	No Change
KNN	k=2	87	0.92	No Change

all machine learning algorithms. One can observe that CAT Boost and RNN are the most effective models in the combined features approach.

X. COMPARISONS OF FINAL PROPOSED MODEL WITH BASELINES

The model implemented in Approach II is the best among all approaches. One obtains further gains through best feature selection. We consider it as the final proposed model. In this

section, its performance is compared with baseline deception detection solutions. Baseline algorithms cover the various categories presented in the fake news taxonomy presented in section II.

These categories include 1. Propagation-based/ network-based model utilizing information exposed through news cascades [6]. 2. Hybrid-based model [28] exploiting both news content and news propagation network information to predict deception. 3. Content-based model [27] fully focused

TABLE 10. Detailed results of Approach II models using the 14 best features. Results with hyper-parameters and others. Performance change after implementing best features. CAT Boost outperforms the alternative with best accuracy(98.4%) and ROC-AUC score (0.98).

Models	Parameters	Accuracy	ROC-AUC	Performance Status
CATBoost	random_seed=42, l2_leaf_reg=7, iterations=20, learning_rate=0.15, depth=2	98.4	0.98	Accuracy Improved
Light Gbm	boosting_type='gbdt', learning_rate=0.1, max_depth=1, n_estimators=100, n_jobs=-1.	98.1	0.97	Accuracy Improved
AdaBoosting	algorithm='SAMME.R', learning_rate=1.0, n_estimators=5, random_state=0, lr=0.1, max_depth=2,	91	0.92	Accuracy Improved
GradientBoosting	random_state=0, n_estimators=50, estimators=3	97.92	0.97	Accuracy Improved
EVC		80	0.82	No Change
Decision Tree	max_depth=2, min_Sample_Split=2	80	0.82	No Change
Random Forest	max_depth=1, measure= gini, random_state =0, min_split=2, estimator=50	86	0.81	No Change
Naïve Bayes	alpha =1	79	0.78	No Change
Logistic	Random state = 1, solver= bfgs	81	0.78	No Change
SVM	kernal = linear svc()	72.4	0.64	Accuracy Improved
KNN	k=4	76	0.70	No Change

TABLE 11. Comparison of the proposed source-based model with the baselines. Baselines cover all types of models presented in fake news taxonomy (see section II). Results show that the proposed deception detection model not only outperforms all baselines but it can perform equally well in different domains i.e. politics, entertainment. It also work well on little training data i.e. 75 news items, and it is well-suited for COVID-19 related deception detection.

S.No.	Best Model	Type	PolitiFact		GossipCop	
			Acc	F1	Acc	F1
1	Vosoughi et al. [5]	Propagation/Network	65	70	80	79
2	Shu et al. [28]	Hybrid	84	84	86	86
3	Shu et al. [27]	Content	64	63	72	73
4	Ji et al. [44]	Style	80	79	60	61
5	Proposed Model	Source Based - Fusion (Network + User Profile)	92	90	91	90
Covid-19			Acc		AUC	
6	Shu et al. [58]	Source Based (Implicit Profile Features)	77		70	
7	Proposed Model	Source Based - Fusion (Network + User Profile)	98.4		98	

on news contents based linguistic information for early fake news detection. 4. Style-based model [44] that depends on news contents through the extraction of writing style and ability to detect fake news in the early phase. Latent representation methods are not considered e.g., [88] and [89] because their performance are not much competitive. 5. Source-based models [58] that indirectly focuses on propagators to detect deceptive contents.

The performance of the best-proposed model is compared with the baselines using three datasets (Politifact, GossipCop, Covid-19). Table 11 reports the results of the comparative analysis. it outperforms all its alternatives whatever their type: content-based (linguistic), Propagation/ Network-based, Hybrid, Style-based, and Source-based models.

One can see that Network/Propagation-based model and Content-based (linguistic) model are far behind, with accuracy scores ranging from 64 % to 80%. Style-based and Source-based models perform better. The hybrid model is the

best alternative. Although content-based models are better for early detection.

Remember that in the final phase of experimentation, conducted on the publicly available datasets of PolitiFact, GossipCop [27] and our COVID-19 dataset, only the best set of features are used. A supervised classifier is trained for deception detection. It is very encouraging that the proposed model outperforms all baselines. It is trained on different datasets covering three domains (politics, entertainment, and COVID-19 pandemics). The proposed model is implemented with minor modifications to better suit the political and entertainment domains. Table 11 shows that the accuracy scores for the proposed model are much higher than all the others. The proposed model's accuracy score for the COVID-19 dataset is 98.4, and for PolitiFact, it is 92, whereas it is 91 for the GossipCop. Results show that the proposed deception detection model outperforms all baselines and performs well in different domains. It also works on little training



FIGURE 12. Best Approach's Accuracy Scores: We investigate eleven ML and two deep learning models for the fake news classification problem. The Accuracy score of each model for Approach II (best among all four approaches) is shown in the figure. One can see that CAT Boost, Light GBM and RNN are the top three performing models, with Accuracy scores of 98.4%, 98.1% and 98%, respectively.

data (i.e., 75 news items), and it is well suited for COVID-19 related deception detection.

XI. CONCLUSION

This paper tackles the novel and challenging problem of fake news detection in COVID-19 epidemics over social media microblogs. While many studies deal with fake news detection in politics and entertainment, there are few reports on COVID-19 epidemics. Four main approaches allow coping with this issue. The Content-based methods focus on the structure of the information itself. The propagation/Network methods exploit the propagation network. Hybrid methods combine various features of both approaches. Finally, recently introduced source-based methods rely on user information to identify fake news. This type of method can easily overcome many limitations of earlier solutions. Therefore, we propose a source-based method that uses information about the source and the propagators of the information. The model detects fake news by analyzing the context of their spreader's community on social networks or social media microblogs. The solution analyses connectivity patterns of such communities together with their user's profile features. We investigate different approaches combining features of the propagation network with user profile-based features, two at the node level (user features alone, user and network features), and two at the community level (aggregated user and network features, aggregated network features). Results of the experiments show that two approaches are very effective in fake news detection. The first one combines network and user profile features at the user level. It is called Approach II. The second one uses network-based features at the community level. It is called Approach IV. In this exploratory study, we perform a comparative analysis of different machine learning and deep learning models. The best results are produced by Approach II when these features are applied at the node level. CATBoost and RNN are the

most effective algorithms with an AUC score of 98%. All non-linear models perform better than the others. The performance of deep learning models is similar to modern machine learning models. The best set of features is also explored and used for further model optimization. A comprehensive comparative study with baselines covering all spectrum of alternative approaches (content-based, Propagation/Network, Hybrid, Source-based) is also conducted using three datasets from various domains (COVID-19, political, entertainment). Results are very encouraging. Indeed, the proposed approach always produces the best results. Additionally, the proposed model works with little training data. This work paves the way for further development of stronger solutions, including network and user features.

APPENDIX

SAMPLE SNAPSHOTS OF REAL AND FAKE NEWS ITEM COMMUNITY'S STRUCTURE

Sample snapshots of two graphs showing real and fake news item community's structure, from our dataset, are presented in figure 13 and figure 14. Figure 13 shows a single fake news article's propagators community and figure 14 is for the true news. The difference between both community's structure is

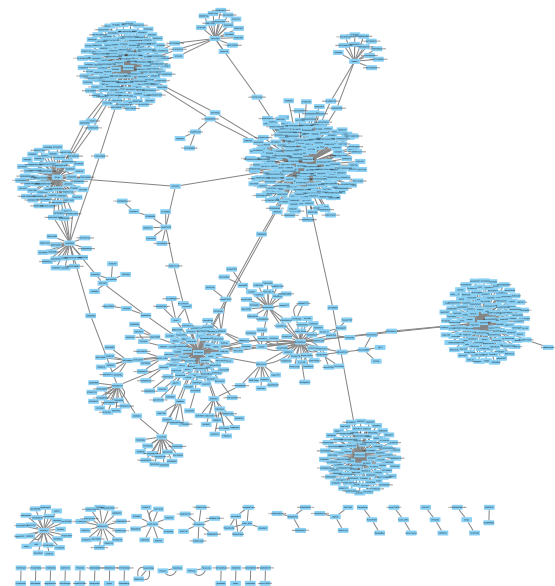


FIGURE 13. COVID-19 Fake News Propagation Community Graph: There are some obvious trends observed in fake community, such as; it has more number of communities which are smaller as well as almost equal in size and all except few are also connected.

completely visible which fully support our research hypothesis. Real community graph has less no of communities with large sizes and the majority of them are not connected whereas graph of fake community has more no of communities which are smaller in size and most of them are also connected. Fake community structure is far more spreading than real.

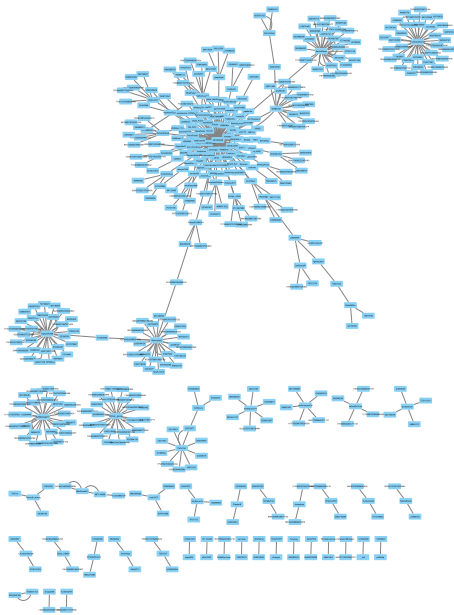


FIGURE 14. COVID-19 True News Propagation Community Graph: There are some obvious trends observed in a real community like real community graph has less number of communities with large and un-equal sizes and almost half of them are disconnected components.

ACKNOWLEDGMENT

The authors would like to thank Sumaiyah Zahid (National University of Computer and Emerging Sciences, Karachi, Pakistan) for data fetching and compilation.

REFERENCES

- [1] S. Martinčić-Ipšić, E. Močibob, and A. Perc, "Link prediction on Twitter," *PLoS ONE*, vol. 12, no. 7, 2017, Art. no. 0181079.
- [2] M. Hu, S. Liu, F. Wei, Y. Wu, J. Stasko, and K.-L. Ma, "Breaking news on Twitter," in *Proc. SIGCHI Conf. Hum. Factors Comput. Syst.*, May 2012, pp. 2751–2754.
- [3] M. Mendoza, B. Poblete, and C. Castillo, "Twitter under crisis: Can we trust what we RT?" in *Proc. 1st Workshop Social Media Anal.*, 2010, pp. 71–79.
- [4] T. Blatchford, *Americans Believe Two-Thirds of News on Social Media is Misinformation*. Accessed: Aug. 17, 2021. [Online]. Available: <https://www.poynter.org/ethics-trust/2018/americans-believe-two-thirds-of-news-on-social-media-is-misinformation/>
- [5] S. Vosoughi, D. Roy, and S. Aral, "The spread of true and false news online," *Science*, vol. 359, pp. 1146–1151, May 2018.
- [6] L. Howell, "Digital wildfires in a hyperconnected world," *WEF Rep.*, vol. 3, pp. 15–94, Dec. 2013.
- [7] J. Bollen, H. Mao, and X. Zeng, "Twitter mood predicts the stock market," *J. Comput. Sci.*, vol. 2, no. 1, pp. 1–8, Mar. 2011.
- [8] A. Hermida, "Twittering the news: The emergence of ambient journalism," *Journalism Pract.*, vol. 4, no. 3, pp. 297–308, Aug. 2010.
- [9] Z. Jin, J. Cao, H. Guo, Y. Zhang, Y. Wang, and J. Luo, "Detection and analysis of 2016 us presidential election related rumors on Twitter," in *Proc. Int. Conf. Social Comput., Behav.-Cultural Modelig Predict. Behav. Represent.* Washington, DC, USA: Springer 2017, pp. 14–24.
- [10] J. De Keersmaecker and A. Roets, "'Fake news': Incorrect, but hard to correct. The role of cognitive ability on the impact of false information on social impressions," *Intelligence*, vol. 65, pp. 107–110, Nov. 2017.
- [11] M. Fisher, J. W. Cox, and P. Hermann. *Pizzagate: From Rumor, to Hashtag, to Gunfire in DC the Washington Post*, 2016. Accessed: Aug. 17, 2021. [Online]. Available: <https://goo.gl/UW43pg>
- [12] K. Starbird, J. Maddock, M. Orand, P. Achterman, and R. M. Mason, "Rumors, false flags, and digital vigilantes: Misinformation on Twitter after the 2013 Boston Marathon bombing," in *Proc. IConference*, 2014, pp. 1–4.
- [13] A. Perrin. *Social Media Usage: 2005-2015*. Accessed: Aug. 17, 2021. [Online]. Available: <https://www.pewresearch.org/internet/2015/10/08/social-networking-usage-2005-2015/>
- [14] Katerina Eva Matsa Elisa Shearer. *Social Media Usage: 2018*. Accessed: Aug. 17, 2021. [Online]. Available: <https://www.journalism.org/2018/09/10/news-use-across-social-media-platforms-2018/>
- [15] H. Allcott and M. Gentzkow, "Social media and fake news in the 2016 election," *J. Econ. Perspect.*, vol. 31, no. 2, pp. 211–236, 2017.
- [16] K. Rapoza. *Can 'Fake News' Impact the Stock Market*. Accessed: Aug. 17, 2021. [Online]. Available: <https://www.forbes.com/sites/kenrapoza/2017/02/26/can-fake-news-impact-the-stock-market/>
- [17] *Boston Attack 2013 BBC*. Accessed: Aug. 17, 2021. [Online]. Available: <http://www.bbc.co.uk/news/world-us-canada-22297568/>
- [18] K. A. Qureshi and M. Sabih, "Un-compromised credibility: Social media based multi-class hate speech classification for text," *IEEE Access*, vol. 9, pp. 109465–109477, 2021.
- [19] 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, 2020.
- [20] M. Luca and G. Zervas, "Fake it till you make it: Reputation, competition, and yelp review fraud," *Manage. Sci.*, vol. 62, no. 12, pp. 3412–3427, 2016.
- [21] A. Smith and V. Banic, *Fake News: How a Partying Macedonian Teen Earns Thousands Publishing Lies*. New York, NY, USA: NBC News, 2016. Accessed: Aug. 17, 2021. [Online]. Available: <https://www.nbcnews.com/news/world/fake-news-how-partying-macedonian-teen-earns-thousands-publishing-lies-n692451>
- [22] F. Pierri and S. Ceri, "False news on social media: A data-driven survey," *SIGMOD Rec.*, vol. 48, no. 2, pp. 18–27, Dec. 2019.
- [23] J. Roozenbeek and S. van der Linden, "Fake news game confers psychological resistance against online misinformation," *Palgrave Commun.*, vol. 5, no. 1, pp. 1–10, Dec. 2019.
- [24] S. Volkova, K. Shaffer, J. Y. Jang, and N. Hodas, "Separating facts from fiction: Linguistic models to classify suspicious and trusted news posts on Twitter," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, vol. 2, 2017, pp. 647–653.
- [25] S. Kumar and N. Shah, "False information on web and social media: A survey," 2018, *arXiv:1804.08559*. [Online]. Available: <http://arxiv.org/abs/1804.08559>
- [26] B. Guo, Y. Ding, L. Yao, Y. Liang, and Z. Yu, "The future of false information detection on social media: New perspectives and trends," *ACM Comput. Surv.*, vol. 53, no. 4, pp. 1–36, 2020.
- [27] K. Shu, D. Mahudeswaran, S. Wang, D. Lee, and H. Liu, "FakeNewsNet: A data repository with news content, social context and spatiotemporal information for studying fake news on social media," 2018, *arXiv:1809.01286*. [Online]. Available: <http://arxiv.org/abs/1809.01286>
- [28] K. Shu, D. Mahudeswaran, S. Wang, and H. Liu, "Hierarchical propagation networks for fake news detection: Investigation and exploitation," in *Proc. Int. AAAI Conf. Web Social Media*, vol. 14, 2020, pp. 626–637.
- [29] J. Nørregaard, B. D. Horne, and S. Adali, "NELA-GT-2020: A large multi-labelled news dataset for the study of misinformation in news articles," in *Proc. Int. AAAI Conf. Web Social Media*, vol. 13, 2019, pp. 630–638.
- [30] 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, 2017.
- [31] S. Ahmed, K. Hinkelmann, and F. Corradini, "Combining machine learning with knowledge engineering to detect fake news in social networks-a survey," in *Proc. Spring Symp.*, vol. 12, 2019, pp. 8–20.
- [32] M. Celliers and M. Hattingh, "A systematic review on fake news themes reported in literature," in *Conf. E-Bus., E-Services E-Society*. Skukuza, South Africa: Springer, 2020, pp. 223–234.
- [33] *200 Manual Fact-Check Sites 2021*. Accessed: Aug. 17, 2021. [Online]. Available: <https://reporterslab.org/fact-checking/>
- [34] T. Mitra and E. Gilbert, "Credbank: A large-scale social media corpus with associated credibility annotations," in *Proc. Int. AAAI Conf. Web Social Media*, vol. 9, 2015, pp. 258–267.
- [35] *Manual Crowd Sourced Web Sites 2021*. Accessed: Aug. 17, 2021. [Online]. Available: <http://fiskkit.com/>
- [36] A. L. Ginsca, A. Popescu, and M. Lupu, "Credibility in information retrieval," *Found. Trends Inf. Retr.*, vol. 9, no. 5, pp. 355–475, 2015.
- [37] B. Shi and T. Weninger, "Discriminative predicate path mining for fact checking in knowledge graphs," *Knowl.-Based Syst.*, vol. 104, pp. 123–133, Jul. 2016.

- [38] V. L. Rubin, N. J. Conroy, and Y. Chen, "Towards news verification: Deception detection methods for news discourse," in *Proc. Hawaii Int. Conf. Syst. Sci.*, 2015, pp. 5–8.
- [39] M. Potthast, J. Kiesel, K. Reinartz, J. Bevendorff, and B. Stein, "A stylometric inquiry into hyperpartisan and fake news," 2017, *arXiv:1702.05638*. [Online]. Available: <http://arxiv.org/abs/1702.05638>
- [40] 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, Jul. 2020.
- [41] V. Pérez-Rosas, B. Kleinberg, A. Lefevre, and R. Mihalcea, "Automatic detection of fake news," 2017, *arXiv:1708.07104*. [Online]. Available: <http://arxiv.org/abs/1708.07104>
- [42] Y. Wang, F. Ma, Z. Jin, Y. Yuan, G. Xun, K. Jha, L. Su, and J. Gao, "EANN: Event adversarial neural networks for multi-modal fake news detection," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 849–857.
- [43] A. Edelbo Lillie and E. Refsgaard Middelboe, "Fake news detection using stance classification: A survey," 2019, *arXiv:1907.00181*. [Online]. Available: <http://arxiv.org/abs/1907.00181>
- [44] Y. Ji and J. Eisenstein, "Representation learning for text-level discourse parsing," in *Proc. 52nd Annu. Meeting Assoc. Comput. Linguistics*, 2014, pp. 13–24.
- [45] F. Pierri, C. Piccardi, and S. Ceri, "Topology comparison of Twitter diffusion networks effectively reveals misleading information," *Sci. Rep.*, vol. 10, no. 1, Dec. 2020, Art. no. 1372.
- [46] K. Shu, D. Mahudeswaran, S. Wang, D. Lee, and H. Liu, "FakeNewsNet: A data repository with news content, social context, and spatiotemporal information for studying fake news on social media," *Big Data*, vol. 8, no. 3, pp. 171–188, Jun. 2020.
- [47] J. C. S. Reis, A. Correia, F. Murai, A. Veloso, and F. Benevenuto, "Supervised learning for fake news detection," *IEEE Intell. Syst.*, vol. 34, no. 2, pp. 76–81, Mar./Apr. 2019.
- [48] C. Silverman. (2016). *This Analysis Shows How Viral Fake Election News Stories Outperformed Real News on Facebook, BuzzFeed News*. [Online]. Available: <https://www.buzzfeed.com/craigsilverman/viral-fake-election-news-outperformed-real-news-on-facebook>
- [49] R. B. Cialdini, *Influence: Science and Practice*, vol. 4. Boston, MA, USA: Pearson, 2009.
- [50] M. McPherson, L. Smith-Lovin, and J. M. Cook, "Birds of a feather: Homophily in social networks," *Annu. Rev. Sociol.*, vol. 27, no. 1, pp. 415–444, 2001.
- [51] S. Classen, "Echo chamber: Rush limbaugh and the conservative media establishment (review)," *Cinema J.*, vol. 51, no. 4, pp. 212–214, 2012.
- [52] S. Xie, G. Wang, S. Lin, and P. S. Yu, "Review spam detection via temporal pattern discovery," in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2012, pp. 823–831.
- [53] A. Mukherjee, A. Kumar, B. Liu, J. Wang, M. Hsu, M. Castellanos, and R. Ghosh, "Spotting opinion spammers using behavioral footprints," in *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2013, pp. 632–640.
- [54] Christopher Paul and Miriam Matthews, *The Russian Firehose of Falsehood Propaganda Model: Why MightWork and Options to Counter*, document PE-198-OSD, RAND Corporation, 2016.
- [55] N. Sitaula, C. K. Mohan, J. Grygiel, X. Zhou, and R. Zafarani, "Credibility-based fake news detection," in *Disinformation, Misinformation, Fake News Social Media*. Cham, Switzerland: Springer, 2020, pp. 163–182.
- [56] B. D. Horne, J. Nørregaard, and S. Adal, "Different spirals of sameness: A study of content sharing in mainstream and alternative media," in *Proc. Int. AAAI Conf. Web Social Media*, vol. 13, 2019, pp. 257–266.
- [57] C. Shao, G. L. Ciampaglia, O. Varol, K.-C. Yang, A. Flammini, and F. Menczer, "The spread of low-credibility content by social bots," *Nature Commun.*, vol. 9, no. 1, pp. 1–9, Dec. 2018.
- [58] K. Shu, S. Wang, and H. Liu, "Understanding user profiles on social media for fake news detection," in *Proc. IEEE Conf. Multimedia Inf. Process. Retr. (MIPR)*, Apr. 2018, pp. 430–435.
- [59] K. Shu, X. Zhou, S. Wang, R. Zafarani, and H. Liu, "The role of user profiles for fake news detection," in *Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining*, Aug. 2019, pp. 436–439.
- [60] J. Amador, A. Oehmichen, and M. Molina-Solana, "Characterizing political fake news in Twitter by its meta-data," 2017, *arXiv:1712.05999*. [Online]. Available: <http://arxiv.org/abs/1712.05999>
- [61] Z. Chu, S. Gianvecchio, H. Wang, and S. Jajodia, "Detecting automation of Twitter accounts: Are you a human, bot, or cyborg?" *IEEE Trans. Dependable Secure Comput.*, vol. 9, no. 6, pp. 811–824, Aug. 2012.
- [62] S. Chandra, P. Mishra, H. Yannakoudakis, M. Nimishakavi, M. Saeidi, and E. Shutova, "Graph-based modeling of online communities for fake news detection," 2020, *arXiv:2008.06274*. [Online]. Available: <http://arxiv.org/abs/2008.06274>
- [63] S. Ali Memon and K. M. Carley, "Characterizing COVID-19 misinformation communities using a novel Twitter dataset," 2020, *arXiv:2008.00791*. [Online]. Available: <http://arxiv.org/abs/2008.00791>
- [64] F. Monti, F. Frasca, D. Eynard, D. Mannion, and M. M. Bronstein, "Fake news detection on social media using geometric deep learning," 2019, *arXiv:1902.06673*. [Online]. Available: <http://arxiv.org/abs/1902.06673>
- [65] M. S. Al-Zaman, "COVID-19-related social media fake news in India," *Journalism Media*, vol. 2, no. 1, pp. 100–114, Jan. 2021. [Online]. Available: <https://www.mdpi.com/2673-5172/2/1/7>, doi: [10.3390/journal-media2010007](https://doi.org/10.3390/journal-media2010007).
- [66] O. Ajao, D. Bhowmik, and S. Zargari, "Fake news identification on Twitter with hybrid CNN and RNN models," in *Proc. 9th Int. Conf. Social Media Soc.*, Jul. 2018, pp. 226–230.
- [67] M. S. Al-Rakhami and A. M. Al-Amri, "Lies kill, facts save: Detecting COVID-19 misinformation in Twitter," *IEEE Access*, vol. 8, pp. 155961–155970, 2020.
- [68] C. P. Rodríguez, B. V. Carballido, G. Redondo-Sama, M. Guo, M. Ramis, and R. Flecha, "False news around COVID-19 circulated less on Sina Weibo than on Twitter. How to overcome false information?" *Int. Multidisciplinary J. Social Sci.*, vol. 9, no. 2, pp. 107–128, 2020.
- [69] V. Priesemann, M. M. Brinkmann, S. Ciesek, S. Cuschieri, T. Czipionka, G. Giordano, D. Gurdasani, C. Hanson, N. Hens, E. Iftikhar, M. Kelly-Irving, P. Klimek, M. Kretzschmar, A. Peichl, M. Perc, F. Sannino, E. Schernhammer, A. Schmidt, A. Staines, and E. Szcurek, "Calling for pan-European commitment for rapid and sustained reduction in SARS-CoV-2 infections," *Lancet*, vol. 397, no. 10269, pp. 92–93, Jan. 2021.
- [70] D. A. Erku, S. A. Belachew, S. Abriha, M. Sinnollareddy, J. Thomas, K. J. Steadman, and W. H. Tesfaye, "When fear and misinformation go viral: Pharmacists' role in deterring medication misinformation during the 'infodemic' surrounding COVID-19," *Res. Social Administ. Pharmacy*, vol. 17, pp. 1954–1963, Jan. 2020.
- [71] S. Albahli, A. Algham, S. Aeraj, M. Alsaeed, M. Alrashed, H. Tayyab Rauf, M. Arif, and M. Abed Mohammed, "COVID-19 public sentiment insights: A text mining approach to the Gulf countries," *Comput., Mater. Continua*, vol. 67, no. 2, pp. 1613–1627, 2021.
- [72] Abdullah-Ali-Tanvir, E. M. Mahir, S. Akhter, and M. R. Huq, "Detecting fake news using machine learning and deep learning algorithms," in *Proc. 7th Int. Conf. Smart Comput. Commun. (ICSCC)*, Jun. 2019, pp. 1–5.
- [73] J. Zhang, B. Dong, and P. S. Yu, "FakeDetector: Effective fake news detection with deep diffusive neural network," in *Proc. IEEE 36th Int. Conf. Data Eng. (ICDE)*, Apr. 2020, pp. 1826–1829.
- [74] *The International Fact-Checking Network*. Accessed: Aug. 17, 2021. [Online]. Available: <https://www.poynter.org/ifcn/>.
- [75] *Hoaxy by Indiana University*. Accessed: Aug. 17, 2021. [Online]. Available: <https://hoaxy.iuni.iu.edu/>.
- [76] *A Python Library for Twitter: Tweepy*. Accessed: Aug. 17, 2021. [Online]. Available: <https://www.tweepy.org/>.
- [77] S. Abdullah and X. Wu, "An epidemic model for news spreading on Twitter," in *Proc. IEEE 23rd Int. Conf. Tools with Artif. Intell.*, Nov. 2011, pp. 163–169.
- [78] J. Skaza and B. Blais, "Modeling the infectiousness of Twitter hashtags," *Phys. A, Stat. Mech. Appl.*, vol. 465, pp. 289–296, Jan. 2017.
- [79] F. Jin, E. Dougherty, P. Saraf, Y. Cao, and N. Ramakrishnan, "Epidemiological modeling of news and rumors on Twitter," in *Proc. 7th Workshop Social Netw. Mining Anal.*, 2013, pp. 1–9.
- [80] A. Wiratsudakul, P. Suparit, and C. Modchang, "Dynamics of Zika virus outbreaks: An overview of mathematical modeling approaches," *PeerJ*, vol. 6, p. e4526, Mar. 2018.
- [81] F. Brauer, "Compartmental models in epidemiology," in *Mathematical Epidemiology*. Berlin, Germany: Springer, 2008, pp. 19–79.
- [82] *Pandas Dataframe*. Accessed: Aug. 17, 2021. [Online]. Available: <https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.html>
- [83] A. M. Hall, "Correlation-based feature selection for machine learning," Ph.D. dissertation, Dept. Comput. Sci., Univ. Waikato Hamilton, Hamilton, New Zealand, Tech. Rep., 1999.

- [84] J. Cai, J. Luo, S. Wang, and S. Yang, "Feature selection in machine learning: A new perspective," *Neurocomputing*, vol. 300, pp. 70–79, Jul. 2018.
- [85] C. Demirkesen and H. Cherifi, "A comparison of multiclass SVM methods for real world natural scenes," in *Proc. Int. Conf. Adv. Concepts Intell. Vis. Syst.* Berlin, Germany: Springer, 2008, pp. 752–763.
- [86] C. X. Ling, J. Huang, and H. Zhang, "AUC: A statistically consistent and more discriminating measure than accuracy," in *Proc. IJCAI*, vol. 3, 2003, pp. 519–524.
- [87] V. Labatut and H. Cherifi, "Accuracy measures for the comparison of classifiers," 2012, *arXiv:1207.3790*. [Online]. Available: <http://arxiv.org/abs/1207.3790>
- [88] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, *arXiv:1301.3781*. [Online]. Available: <http://arxiv.org/abs/1301.3781>
- [89] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 1188–1196.



puting. He has authored several research articles along with chapters in different books.

KHUBAIB AHMED QURESHI is currently affiliated with the Department of Computer Science, DHA Suffa University, Karachi, Pakistan, as an Assistant Professor, and the Head of the Data Science Program. He was the Head of the Computer Science Department, HIMS, Hamdard University, Karachi. He is having 20 years of comprehensive research and teaching experience, continuing research in the area of computer science named data science, complex networks, and social com-



ground in designing novel solutions for complex systems. He is currently affiliated with the Department of Computer Science, National University of Computer and Emerging Sciences, as an Assistant Professor, continuing research in the specialized scientific area of computer science, complex networks, social computing, bioinformatics, and integrated systems. He has authored several articles along with chapters in different books.

RAUF AHMED SHAMS MALICK received the Ph.D. degree from the University of Karachi. He has been a Visiting Scholar with NIG, Japan, and UCLA, USA. He has founded several companies with state-of-the-art products related to social media, location-based analytics, and organizational networks. He is involved in complex system research and pursuing problems in the area of biological networks, networked economics, and personality traits. He has distinguished back-



MUHAMMAD SABIH received the B.E. degree in industrial electronics from IIEE-NED University, Pakistan, in 2000, and the M.S. and Ph.D. degrees in systems engineering from KFUPM, Dhahran, Saudi Arabia, in 2009 and 2014, respectively. During his research period at KFUPM, he worked for several applied research collaborations between KFUPM and MIT. His data-driven research work expanded to a funded industry-academia collaboration project between KFUPM and Yokogawa-Saudi Arabia, and turned into a U.S. patent. He also worked as an Algorithm Specialist and developed anomaly detection algorithms using Python on the pipeline inspection data at the leading Research and Technology Center (RTRC) of German ROSEN Group at Dhahran Techno-Valley (DTV), Saudi Arabia. He is currently an Assistant Professor with DHA Suffa University and actively engaged in developing solutions from industrial data utilizing machine learning methods for estimation, modeling, and compensation. He has been working in the field of computer and electrical engineering, since 2009, and a Professional Member of the International Society of Automation (ISA). He has one U.S. patent and around ten peer-reviewed articles. His current research interests include Industry 4.0, data science, modeling, and estimation for real-world problems.



HOCINE CHERIFI received the Ph.D. degree from National Polytechnic Institute, Grenoble, in 1984. He has been a Professor of computer science with the University of Burgundy, Dijon, France, since 1999. Prior to moving to Dijon, he held faculty positions at Rouen University and Jean Monnet University, France. He has held visiting positions at Yonsei University, South Korea; The University of Western Australia, Australia; National Pingtung University, Taiwan; and Galatasaray University, Turkey. He has published more than 200 scientific papers in international refereed journals and conference proceedings. He held leading positions in more than 15 international conference organizations (the general chair and the program chair) and has served in more than 100 program committees. His recent research interests include computer vision and complex networks. He is currently the Founder of the International Conference on Complex Networks and their Applications. He is also a member of the Editorial Board of *Computational Social Networks*, *PLOS One*, *IEEE Access*, *Journal of Imaging*, *Complex Systems*, *Quality and Quantity*, and *Scientific Reports*. He is also the Founding Editor-in-Chief of the *Applied Network Science* journal.

...