ISSN 2007-9737

A Social Network Based Approach for Detection of Fake News on Twitter Data Using Machine Learning

Maryam Ahmad[1], Akmal Saeed[1], Rabee Ayaz Abbasi[1], Muhammad Tayyab Zamir[2], Fida Ullah[2], Alexander Gelbukh[2,*], Grigori Sidorov[2], Edgardo Manuel Felipe Riverón[2]

[1] Quaid-e-Azam University,
Faculty of Computer Science,
Pakistan

[2] Instituto Politécnico Nacional,
Centro de Investigación en Computación,
Mexico

maryamahmadfazaian@gmail.com,
{mzamir2023,gelbukh,sidorov,edgardo}@cic.ipn.mx

Abstract. The availability of social media together with its enhanced accessibility has led to increasingly fast spread of deceptive information while creating serious troubles for society and its citizens. The nature of fake news within modern digital settings creates doubts about its effects on public belief and political decisions as well as democratic functions. Fake news operations already existed, but technological progress combined with social media platform growth especially among YouTube and Facebook and Twitter users has created ideal conditions for fast spreading misinformation. The urgent need exists to investigate how false information spreads through multiple social media platforms because of its concerning rate of growth. This study utilizes a social networking detection method that depends on network properties through the Communities through Directed Affiliation (CoDA) algorithm. Different experiments were performed to validate the proposed approach through evaluations on the FakeNewsNet dataset. Experimental findings show an Random forest achived best results with accuracy 0.83, F1-score 0.71,precision 0.78 and recall 0.64 among all the other models from the proposed detection methods. Research findings from this study expand the field related to fake news detection through network-based perspectives that enhance existing methods using content-based and linguistic analysis approaches.

Keywords. Natural language processing, machine learning, preprocessing, fake news.

1 Introduction

Fake news is termed as misinformation, and low-quality news [1]. Fake news is defined as a news article that is verifiably and intentionally false and it can mislead the readers [2]. The extensive propagation of fake news can negatively impact individuals' lives [3].

It persuades consumers deliberately to believe in false beliefs that are shared to spread specific agendas. Fake news that promotes a particular opinion or viewpoint regarding a brand, organization, or product [2] may be false and intentionally spread to mislead the audience. The spread of fake news also affects our economies, with massive trades and market fluctuations linked to misinformation.

For example, a piece of fake news claimed that, during an explosion, Barack Obama was badly injured. This misinformation caused a significant drop in stock value [4].

Fig. 1. Example of a fake news

In the previous year, researchers from EU DisinfoLab identified around 265 pro-India websites operating across 65 countries. These were eventually traced back to the Srivastava Group of New Delhi. The researchers identified that they had identified a whole network of various coordinated NGOs of UN encouraging the interests of India and making repeated criticism on Pakistan (see Fig. 1).

The development of fake media within Geneva, Brussels and across the world is done through obscure networks of local media and ANI in around 97 countries. It aims to multiply repeating the negative content related to countries that are having conflict with India, specifically Pakistan.

These losses and events sparked the discussion around fake news. Such events have motivated research in fake news intentionally published propaganda, misinformation, and hoaxes by the organizations and individuals. All of this content is presented as factual. Such content includes social media posts, blogs, and fake online media releases, which spread false news and misguide people.

It does not include news satire such as The Shovel or as The Onion as their content is not presenting as legitimate factual news.

Rather than misinformation, their intention is satire. Social media quickly spread the real and fake news.

2 Related Work

Few datasets are publicly available. For the FND ("Fake News Detection") the agreed benchmark has not been generated yet [5]. The reason behind this is that there is not clear fake news definition, and there is trouble while collecting the relevant data for the analysis. Many researchers have emphasized the datasets creation containing a different type of statements from various social media platforms. Such statements are made by public figures or politicians, labelled with the information regarding their veracity [6].

CREDBANK [7] is considered as large dataset; it contains around sixty million tweets. These tweets are categorized into events by using techniques of topic modelling. For credibility, each event is efficiently annotated through MK(Mechanical Turk). From the non-research relevant perspective, it is good to BS Detector 14 dataset; Kaggle15 has developed it. It is considered as the web crawler with the knowledge regarding websites of fake news. The main problem is that the gold standard is not viewed since humans cannot annotate it.

In several methods concerning fake news, linguistic cues have been efficiently implemented in [8, 9]. There is different type of features, we can differentiate among semantic, lexical and syntactic linguistic features. The syntactic features include frequency of content words (verbs, nouns, adjective) and presence of particular POS patterns, these are employed in the [10]. Actual words in texts are concern of lexical features.

Word embeddings are also very helpful in several contexts in research field of NLP, and this has been implemented successfully for the FND, especially in the DL (deep learning), and ML methods [11, 12].

There are different features which are user-based are used for detection of fake news such as (account age, frequency of posts, number

of followers/friends and many more) who share or produce the news. These features are also beneficial in the detection of users as fake or real (such as bots, social spammers, etc.) [13]. Multiple studies consider the account age, profile description and URLs which are linking to numerous external resources in [10].

There are network-based features which are helpful to model the information concerning network properties where the entire news is posted or shared. It includes diffusion patterns, sub-graph properties (e.g. clustering coefficient and density) and propagation structures.

Numerous different networks are built, these networks are created on friendship status, which is present among multiple users and propagation news patterns and features are extracted based on degree and clustering coefficient of networks. Finally, extensively content features are implemented and studied for the FND.

Content plays a significant role in the detection of fake news. User-based features are also very important and used in multiple studies. Other studies have basically incorporated both user and content-based features.

3 Methodology

In this part, we discuss the setup of data collection, the coding scheme, and the whole process of selecting the tweets.

Figure 2 explains the methodology performed in the research presented in this paper

3.1 Dataset

FakeNewsNet dataset has been used in this research. This dataset includes social context, news context and dynamic information. Dataset FakeNewsNet has potential to entertain multiple promising directions of research like mitigation, fake news detection and fake account detection, etc. [8].

The preprocessing of dataset is done using preprocessing techniques. Dataset is downloaded from Github[1]. FakeNewsNet dataset is publicly available. FakeNewsNet contains two very comprehensive datasets, i.e., GossipCop and Politifact datasets. FakeNewsNet contains label data. The news article's original statement is published in PolitiFact. For experiments, we use the GossipCop dataset, which contains entertainment stories.

Table 1. FakeNewsNet Dataset

Dataset	Fake-News	Real News	Acc.	Prec.	Rec.	F1
GossipCop	770,000	330,000	83%	78%	64%	71%

The fake and real news are usually related to gossip about relationship among celebrities. From the GossipCop dataset, we extracted two fields which are User IDs and URLs. The dataset we have taken, we analyze and monitor all the tweets, which include a minimum one URL to categorize the wide spreading of fake news on twitter.

Twitter APIs are used to extract the required field from Twitter by using tweet ids, which are given in the dataset. Dataset is filtered efficiently and remove noise, missing and duplicate values from them by using data mining techniques.

We finally obtain 1 million tweets from GossipCop. User ids of each user are unique. User may share one news or more than one news, which contain minimum one URL. Finally, our dataset statistics are shown in Table 1.

3.2 Preprocessing

It is a procedure in which input data is processed, and fined data is extracted as a result. It is essential to preprocess the dataset to decrease the noise. Different preprocessing techniques have been applied to the dataset.

Our dataset may have duplicate users who are sharing the same URLs. Therefore, these users have been removed from the dataset.

FakeNewsNet dataset contains some missing values, which are not effective and efficient for the FND. Thus, we have removed such values from the dataset.

[1] https://github.com/KaiDMML/FakeNewsNet/tree/master/dataset

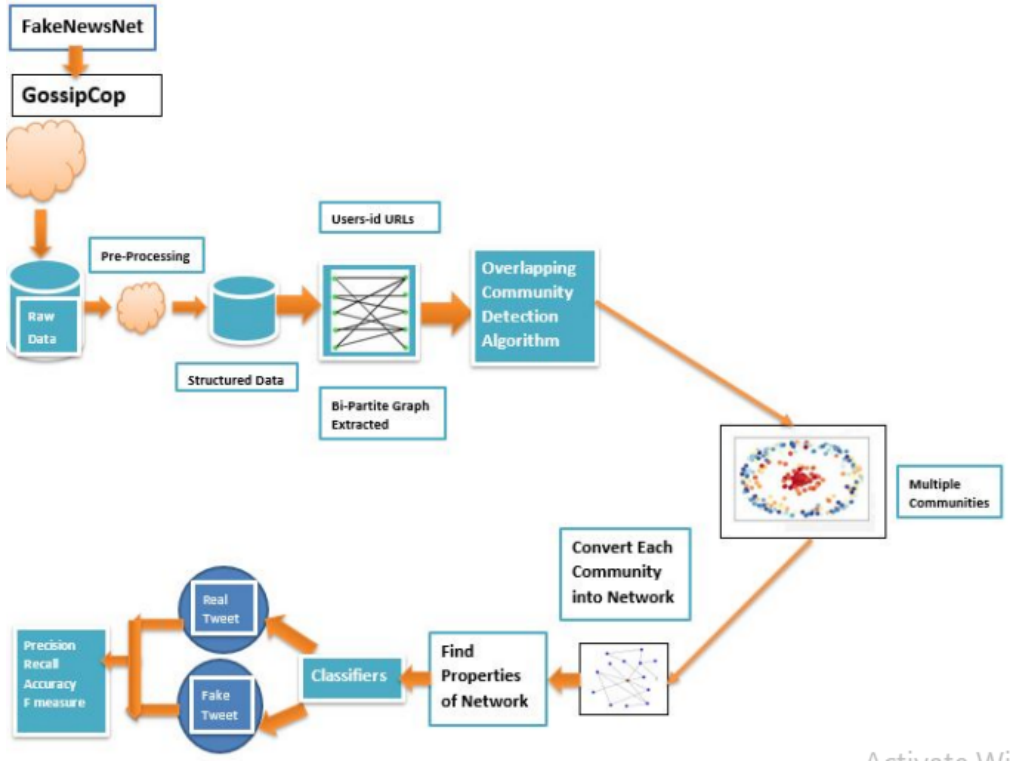

Fig. 2. Proposed methodology for detection of fake news

3.3 Bipartite Graph

In this research, bipartite graph has been used because of the fact that it represents the binary relation among users' ids and URLs. Our problem is based on URLs, so we have extracted these two fields from the dataset. In which we can partition all nodes into two groups, G1 and G2, such that all edges of the graph include an element of G1 and an element of G2. N1 and N2 represent the total number of nodes in each group.

The bipartite graph has distinct properties, and it is a special type of graph. The bipartite graph contains two sets of vertices X and Y. The set X vertices join solely with the set Y vertices. The vertices which are present within the same group will not join.

The Bi-graph term is also used for the bipartite graph. These two sets are disjoint. It implies that they have no common element within the same set. There are multiple bipartite graphs applications. In the medical field, a bipartite graph can be utilized in the detection of throat cancer, lung cancer, etc.

In our research, 02 groups are available. In one group, user ids are available, and in another group URLs are available. X denotes the user id's group, and Y represents the URLs group. Each edge connects the vertex in X to one in Y. Vertex sets X and Y are mostly known as the graph parts.

Mathematically, it can be represented by G= (X, Y, E) to represent a bipartite graph, in which each partition has the parts X and Y with E. E representing the graph edges. For bipartite graph notation is represented in Eq. 1:

$$G = (X, Y, E), \tag{1}$$

$$X = \{x_1, x_2, x_3, ..., x_{|X|}\}, \tag{2}$$

$$Y = \{y_1, y_2, y_3, ..., y_{|Y|}\}, \tag{3}$$

$$T = \{(x_i, y_j), ...)\}. \tag{4}$$

X represents the users' id as shown in equation 2, Y presents the URLs as shown in equation 3 and T represents that user id x_i has shared y_j URLs as shown in equation 4.

In the literature review, we have examined that multiple studies have used a bipartite graph for different purposes to enhance the performance of algorithms.

Bipartite graph is used in various studies. Bipartite graph is extracted from data in order to detect graph based fake reviews.

3.4 Overlapping Community Detection Algorithm

In network analysis, community detection is an essential task. This study aim to find different network partitioning which basically groups together multiple vertices that have similar connectivity patterns of community level.

One essential network characteristic is called as network community structure. It implies groups of nodes strongly connected among themselves as compared with nodes that belong to other groups.

In this research, we have used the CODA ("Communities through Directed Affiliations") algorithm. It is the method that is used for the overlapping communities; it scales to the networks with millions of network edges.

Our dataset contains huge number of tweets, CODA works well on FakeNewsNet dataset. Therefore, this algorithm has been chosen for community detection. The CODA exhibits multiple properties. CODA detects 2-modes, and it also detects cohesive, connected communities.

Multiple methods for the overlapping community detection can solely process the entire network, which contains up to 10,000 nodes. At the same time, CODA could quickly handle the entire networks that contain millions of nodes as well as millions of network edges. The CODA can be efficiently and effectively parallelized. It maximizes the scalability.

3.4.1 Network Based Features

In this research, we do not depend on any type of textual information such as user description, tweet content or user reply. We are treating each network property as a feature and based on these properties we are identifying fake and real news. From the network properties, we are solely choose the following properties:

— Density,

— Clustering Coefficient,

— Number of Nodes,

— Number of Edges,

— Betweenness ,

— Closeness Centrality,

— Eigenvector Centrality,

— Degree Centrality.

Density is defined as the total number of links that particular node has, divided by total possible links a node can have. A 100 percent density is the most excellent density. Network density explains the portion of connections of the nodes which are actual connections. The formation of density is shown in equation 5:

$$NetworkDensity = \frac{ActualConnections}{PotentialConnections}. \quad (5)$$

In equation 5, an actual connection is a connection that exists among two nodes. By contrast, a potential connection is a connection that exists among two nodes regardless of either or not this connection does.

Clustering Coefficient is the measure of the degree in graph theory to which particular nodes in a graph require to make clusters together. In the real-world networks, proofs suggest that nodes create effective groups or clusters. The formation of CC is shown in equation 6:

$$C_i = \frac{2|\{e_jk : V_j, V_k \in N_i, e_jk \in E\}|}{k_i(k_i - 1)}. \quad (6)$$

The local clustering coefficient is shown in the equation 6. In the neighborhood N_i, K_i denotes the number of vertices. C_i is considered as the local clustering coefficient for a vertex V_i.

It is then presented by the ratio of links among the vertices. Vertices neighborhood is divided by the total number of links which are possible to exist among them. Variable "e" denotes the edges, an undirected graph has the characteristic that e_{ji} and e_{ij} are considered similar. Thus, if a vertex v base i have k_i neighbors then within the neighborhood there exist k_i (k_i -1)/2.

Within the network, each community consists of a wide number of nodes. Any device or system connected to a network is known as a node. According to our research problem, each tweet is considered as a node. Each node has distinct characteristics. Therefore, a notation used for a number of nodes is N. Each particular node is represented by n_i:

$$N = \{n_1, n_2, \ldots, n_n\} where \;\; n_i \in N. \tag{7}$$

Within the network, each community consists of a wide number of edges. According to our research problem, each connection among tweets is considered as an edge. Therefore notation used for number of nodes is E. Each particular edge is represented by e_i:

$$E = \{e_1, e_2, \ldots, e_n\} where \;\; e_j \in E. \tag{8}$$

For properly understanding the networks, a centrality feature is an important tool. These algorithms basically use the graph theory to calculate the node importance in a network. Betweeness Centrality feature measures the number of times; a particular node appears on the shortest path among other nodes. In the network, this feature represents that which nodes are basically a bridge among nodes. The formula for finding betweenness centrality is written in equation 9:

$$G(V) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}}. \tag{9}$$

The node betweenness centrality is defined in the equation 9. The variable σ_{st} presents the number of the shortest paths from one node s to another node t and $\sigma_{st}(v)$ are those paths that pass via the vertex v.

Closeness Centrality indicates the closeness of the node to all other available nodes in the network. It is determined as the sum of the shortest path length from a particular node to every other node available in the network. Therefore, the closeness of the node is considered as high if it is more central.

Closeness is defined as the farness reciprocal that is represented in equation 10:

$$C(x) = \frac{1}{\sum_y d(y,x)}, \tag{10}$$

d(y,x) represents the distance among vertices x and y. When discussing closeness centrality then individual mostly refers to its normalized form, which basically demonstrates the shortest paths average length rather than their sum.

Eigen centrality measures an influence of node based on the number of connections; this node has to other nodes which are present in the network. This feature is very useful for understanding the different type of networks, such as malware propagation. For a given graph G= (V, E), the eigenvector centrality can be calculated by using equation 11:

$$x_v = \frac{1}{\lambda} \sum_{t \in_M (V)} x_t = \frac{1}{\lambda} \sum_{t \in G} a_{vt} x_t. \tag{11}$$

x is the relative centrality, vertex v score could be defined in the equation 11 where M(v) is a neighbor set of vertex v and lambda is a constant.

Importance score is assigned to each network. Degree is basically based on the number of edges held by each node. The degree of entire network is calculated. The vertex degree centrality v, for a given network G:= (V, E) with vertices $|V|$ and edges $|E|$ edges, it is formulated as:

$$C_D(v) = deg(v). \tag{12}$$

In a graph, degree centrality is calculated for all the nodes theta V^2 in a dense adjacency matrix graph representation, and for the network, edges take $\theta(E)$ in a representation of the sparse matrix.

The centrality definition on the node level can be expanded to the entire graph, in which scenario we are discussing related graph centralization. Degree measure is also used in literature.

3.5 Classifiers

Classification is a significant and essential part of machine learning. The capability to effectively classify the observations is highly valuable for different business applications. For instance, predicting whether a specific user would purchase a product. Data science contains several classification algorithms like SVM, logistic regression, decision tree, Gaussian Naive Bayes, Bernoulli NB, SGD classifier, Random Forest Classifier, and many more. In the classifier hierarchy, the random forest classifier is considered as the top classifier.

One classifier named Gaussian NB is used, it is the Naive Bayes variant that follows the continuous data and Gaussian normal distribution. Bernoulli Naive Bayes and Multinomial Naive Bayes are also used beside Gaussian Naive Bayes. Gaussian NB has been used because this classifier is very popular and simplest[2].

Another classifier, Bernoulli NB, implements the classification algorithms and naive Bayes algorithms for the data, which is entirely distributed. Every feature is supposed to be the variable of binary-value[3].

Thus, this type of class needs samples to be shown as the feature vectors of binary-valued. In the text classification case, vectors of word occurrence might be used to train and to use the Bernoulli NB classifier efficiently. On several datasets, Bernoulli NB may perform better.

One more classifier is "Quadratic Discriminant Analysis". It is similar to LDA, only except that individuals suppose that covariance matrix could be unique from each particular class. Thus, the covariance matrix can be estimated. It is separate for each specific class k; k can be 1, 2,., k[4].

QDA allows more flexibility; it tends to fit the entire data better compared to LDA. Significantly, the number of parameters maximizes with the QDA. The reason is that for each class, there is a separate covariance matrix with QDA. If there are multiple classes, but there are no numerous points, this can be the major issue.

Another Stochastic Gradient Descent (SGD) classifier is used. This classifier implements the learning routine of SGD, which supports different penalties and loss functions for the classification. A module of SGD Classifier module is given by sci-kit-learn for the implementation of SGD classification. SGD is considered an efficient and practical approach to fitting the regressors and linear classifiers based on a different function of convex loss like Logistic Regression and SVM. There are multiple benefits of SGD, such as its implementations is easy and efficient. Similarly, numerous disadvantages also exist like SGD needs several hyperparameters like several iteration and regularization parameters. SGD is very sensitive towards feature scaling.

DT classifier is very famous classifier used in regression and classification. This classifier divides the dataset into subsets; at this time, an incremental decision tree is created. There are two or more branches present in the decision node—leaf node shown as a decision or classification. In the tree, the top-most nodes correspond towards the optimal predictor named as root node.In multiple papers, this classifier is used for the detection of fake content.

The decision tree can handle both numerical and categorical data. Using this algorithm, we begin from tree root and divide the data based on the feature that basically results in the largest IG. Several famous decision tree classifiers are a random tree, CART, REPTree and ADTree.

RF is the supervised learning algorithm. It is used for regressions as well as for classification. This classifier is mainly used for the problems of classification. As the forest contains so many trees and if any forest has more trees, it means it is more robust. Similarly, the RF algorithm builds a decision tree; from each particular tree, it gets prediction and selects the optimal solution through

[2] https://dataaspirant.com/gaussian-naive-bayes-classifier-implementation-python/

[3] https://scikit-learn.org/stable/modules/naive_bayes.html

[4] https://online.stat.psu.edu/stat508/book/export/html/696

voting. This method decreases the overfitting by averaging the entire result.

Random forest classifier is more comfortable and more flexible to use. This classifier gives excellent and useful feature importance indicators. The random forest consists of multiple applications like recommendation engines, feature selection, and image classification. It can classify the applicants of loyal loans, predict disease, and recognize the fraudulent activity. RF classifier improves the accuracy and also decreases overfitting.

KNN algorithm is the kind of supervised ML algorithm that can be utilized for both regressions and classification predictive problems. KNN is considered as a lazy and slow learning algorithm in a case when size of data is huge. This algorithm does not have any specialized phase of training and it utilizes entire data for training.

"Ada Boost" classifier combines the algorithm of the weak classifier to create a robust classifier. A single algorithm might classify all objects poorly. On the other hand, if multiple classifiers are combined with the training set selection at each iteration and through assigning the correct figure of weight in the final voting, the accuracy score of classifier gets higher.

Multi-layer Perception(MLP) is the last classifier that we have used for the classification purpose in our research. Ambiguously, the MLP term is used. MLP contains at least 3 layers of nodes: an input layer, other one is a hidden layer, and the last one is the output layer. MLP is the class of the feedback ANN. MLP classifier is used for the automatic detection of various fake news.

3.6 Evaluation Measures

It is important to evaluate the Machine Learning Algorithms and Social Networking Technique used for the FND. For this purpose, several evaluation metrics have been used in various studies. In most of the papers, the confusion matrix is used. Accuracy, Recall, Precision, and F1-score have been used to determine the efficiency of their proposed algorithms.

Confusion Matrix In social networking and ML problems, different metrics are efficiently used to measure the efficiency and performance of proposed algorithms. It deals with the precise explanation of prediction outcomes. It describes several correct and incorrect values of prediction. In the confusion matrix, the following different values are used.

False Negatives (FN): It denotes the instance, which is classified as fake, but it is real.

True Negatives (TN): It denotes the instance, which is accurately classified as real.

True Positives (TP): It denotes the instance, which is accurately classified as fake.

False Positives (FP): It denotes the instance, which is classified as real, but it is fake.

The above values are very useful in measuring recall, precision, accuracy, and F1-score. These are metrics, which are used for evaluation in the literature.

Precision: Precision predicts the proportion of positive cases:

$$Precision = \frac{TP}{FP + TP}. \tag{13}$$

Recall: Recall describes the proportion of the original positives which are predicted as positive accurately:

$$Recall = \frac{TP}{FN + TP}. \tag{14}$$

F1-score: It is calculated by taking harmonic mean of precision and recall (also called F1-measure):

$$F1 - Measure = 2 \times \frac{Recall \times Precision}{Recall + Precision}. \tag{15}$$

Accuracy: Accuracy is used to evaluate the entire efficiency of social networking techniques:

$$Accuracy = \frac{TN + TP}{TN + TP + FN + FP}. \tag{16}$$

Additionally, AUC (Area Under the Curve) and ROC (Receiver Operating Characteristics) have been used in different papers.

4 Experiment Setup and Results

Experiments are conducted in this section to evaluate the efficiency and performance of our approach for fake news detection.

We use following performance metrics to evaluate the performance of our approach for the detection of fake news: accuracy, recall, precision and F1-measure.

In this section of paper, we determine the efficiency of our approach for fake news detection in a social network. Bipartite graph is extracted from the dataset.After the extraction of bipartite graph, we moved forward to the final experiments. Different communities have been extracted by applying CODA. We have extracted 200, 300, 500, 1,000, 2,000, 5,000, 10,000, 20,000, 50,000, 100,000 communities from bipartite using CODA as shown in Figure 3. CODA is quite fast algorithm.

After the extraction of network properties, we have performed multiple experiments by using scikit learn. Nine well-known classifiers are used for the prediction: Gaussian NB, SGD classifier, Bernoulli NB, Quadratic Discriminant Analysis, Random Forest classifier, DT classifier, MLP classifier and Ada Boost classifier. The comparison of the performance of all models has been made, as shown in Figure 3. These nine algorithms remarkably and efficiently perform with multiple network features.

These classifiers classify each URL either as fake or real. Different classifiers classify the URLs based on different network properties. These all properties are treated as features. The URLs which are present in each community is also labeled based on these features as either fake or real.

In Figure 3, horizontal line represents different no of communities and vertical line represents different evaluation metrics. As shown in Figure 3, different classifiers are used, and among all these classifiers, Random Forest is providing better results as compared to other classifiers.

Performance of Random Forest is efficient at 200 communities. We have used multiple features. Each feature defines the relationship among the URLs. Based on the majority of URLs, Random Forest classifies the URLs. As our dataset contains a huge number of URLs, so we have a huge number of trees in a forest, so this classifier cannot overfit the entire model.

Accuracy acquired by Random Forest is 83.2%. Precision acquired by RF is 78%, recall acquired by RF is 64% and F1-score acquired by RF classifier is 71%. While calculating accuracy, the highest accuracy (87.2%) is acquired by GNB at 100,000 communities in comparison with all other classifiers, as shown in Figure 3(a).

We have observed that GNB shows optimal results for a more significant number of communities. Accuracy is increased by maximizing the number of communities. In comparison with the other classifiers, the least accuracy (18%) is acquired by BNB at 10,000 communities, as shown in Figure 3(a). Furthermore, we have seen that BNB accuracy decreases if we keep on increasing the number of communities.

However, while calculating recall, the highest recall (97.6%) is acquired by SGD at 2000 communities in comparison with all other classifiers as shown in Figure 3(b). However, in comparison with different classifiers, the least recall is acquired by BNB at 10,000, 20,000, 50,000, and 100,000 communities as shown in Figure 3(b).

While calculating precision, the highest precision (94.7%) is acquired by MLP at 10,000 communities in comparison with all other classifiers, as shown in Figure 3(c). However, in comparison with the other classifiers, the least precision is acquired by BNB at 10,000,20,000, 50,000, and 100,000 communities, as shown in Figure 3(c).

While calculating F1-score, the highest F1-score (70.7%) is acquired by RF at 200 communities in comparison with all other classifiers, as shown in Figure 3(d). However, in comparison with the other classifiers, the least F1-score is acquired by BNB at 10,000,20,000, 50,000, and 100,000 communities as shown in Figure 3(d).

The performance of Bernoulli NB is not as good as compared to other classifiers in terms of accuracy. Gaussian NB has good performance in many communities compared to a smaller number of communities in terms of accuracy. Bernoulli NB has poor performance in comparison with other classifiers in terms of F1-score, recall and precision as the number of communities increases.

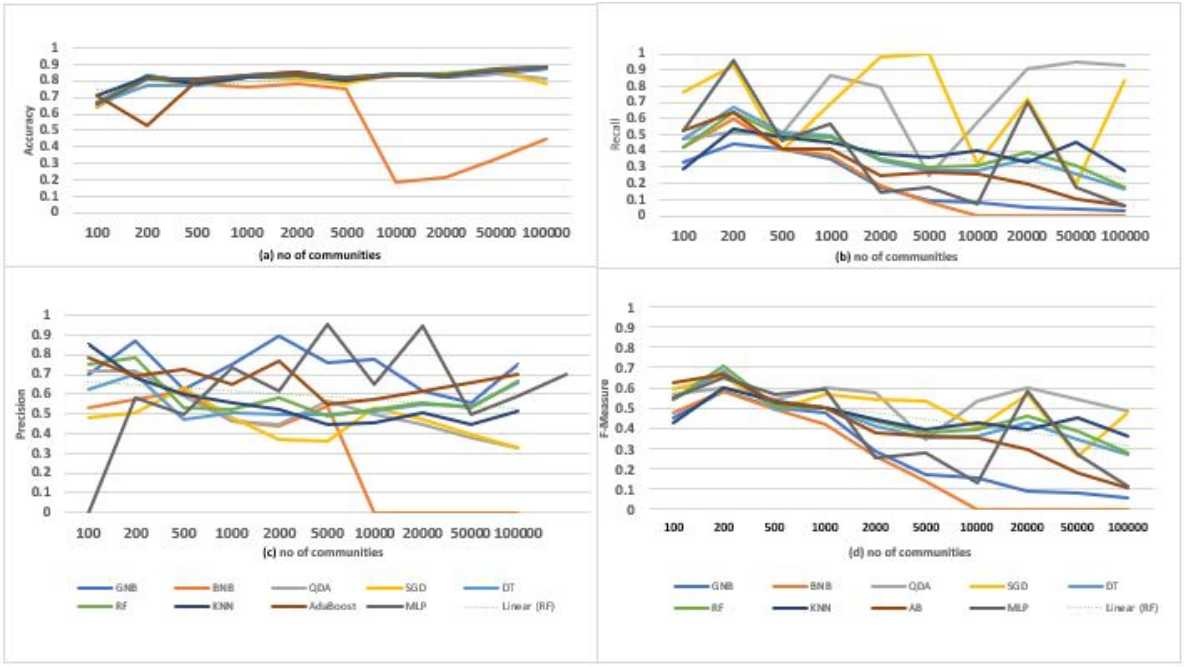

Fig. 3. Different communities detected by CODA (Communities through Directed Affiliations)

Table 2. Results of evaluation metrics used by different classifiers

Communities	Metrics	GNB	BNB	QDA	SGD	DT	RF	KNN	AB	MLP
100	Precision	0.70	0.53	0.71	0.49	0.62	0.75	0.86	0.79	0.58
	Recall	0.33	0.43	0.48	0.76	0.48	0.43	0.29	0.52	0.52
200	Precision	0.87	0.57	0.72	0.51	0.70	**0.78**	0.69	0.69	0.49
	Recall	0.44	0.60	0.51	0.93	0.66	**0.64**	0.53	0.64	0.96
500	Precision	0.63	0.62	0.59	0.63	0.47	0.54	0.60	0.73	0.73
	Recall	0.42	0.42	0.51	0.40	0.52	0.49	0.48	0.42	0.47
1,000	Precision	0.75	0.47	0.46	0.48	0.51	0.53	0.56	0.65	0.61
	Recall	0.35	0.38	0.86	0.69	0.50	0.49	0.45	0.41	0.57
2,000	Precision	0.90	0.44	0.45	0.37	0.50	0.58	0.52	0.77	0.96
	Recall	0.17	0.19	0.79	0.98	0.34	0.35	0.38	0.25	0.15
5,000	Precision	0.76	0.54	0.57	0.36	0.50	0.49	0.44	0.55	0.65
	Recall	0.09	0.08	0.25	1.00	0.27	0.30	0.36	0.27	0.18
10,000	Precision	0.77	0.00	0.50	0.53	0.51	0.52	0.45	0.58	0.95
	Recall	0.09	0.00	0.58	0.32	0.28	0.32	0.40	0.26	0.07
20,000	Precision	0.61	0.00	0.45	0.47	0.55	0.55	0.50	0.62	0.49
	Recall	0.05	0.00	0.91	0.72	0.35	0.40	0.33	0.19	0.71
50,000	Precision	0.56	0.00	0.38	0.39	0.54	0.53	0.45	0.66	0.59
	Recall	0.42	0.00	0.95	0.20	0.26	0.31	0.45	0.11	0.17
1,000,000	Precision	0.75	0.00	0.33	0.33	0.66	0.67	0.52	0.70	0.70
	Recall	0.03	0.00	0.92	0.83	0.17	0.18	0.28	0.05	0.06

Table 3. Results of evaluation metrics used by different classifiers

Communities	Metrics	GNB	BNB	QDA	SGD	DT	RF	KNN	AB	MLP
100	Precision	0.70	0.53	0.71	0.49	0.62	0.75	0.86	0.79	0.58
	Recall	0.33	0.43	0.48	0.76	0.48	0.43	0.29	0.52	0.52
200	Precision	0.87	0.57	0.72	0.51	0.70	**0.78**	0.69	0.69	0.49
	Recall	0.44	0.60	0.51	0.93	0.66	**0.64**	0.53	0.64	0.96
500	Precision	0.63	0.62	0.59	0.63	0.47	0.54	0.60	0.73	0.73
	Recall	0.42	0.42	0.51	0.40	0.52	0.49	0.48	0.42	0.47
1,000	Precision	0.75	0.47	0.46	0.48	0.51	0.53	0.56	0.65	0.61
	Recall	0.35	0.38	0.86	0.69	0.50	0.49	0.45	0.41	0.57
2,000	Precision	0.90	0.44	0.45	0.37	0.50	0.58	0.52	0.77	0.96
	Recall	0.17	0.19	0.79	0.98	0.34	0.35	0.38	0.25	0.15
5,000	Precision	0.76	0.54	0.57	0.36	0.50	0.49	0.44	0.55	0.65
	Recall	0.09	0.08	0.25	1.00	0.27	0.30	0.36	0.27	0.18
10,000	Precision	0.77	0.00	0.50	0.53	0.51	0.52	0.45	0.58	0.95
	Recall	0.09	0.00	0.58	0.32	0.28	0.32	0.40	0.26	0.07
20,000	Precision	0.61	0.00	0.45	0.47	0.55	0.55	0.50	0.62	0.49
	Recall	0.05	0.00	0.91	0.72	0.35	0.40	0.33	0.19	0.71
50,000	Precision	0.56	0.00	0.38	0.39	0.54	0.53	0.45	0.66	0.59
	Recall	0.42	0.00	0.95	0.20	0.26	0.31	0.45	0.11	0.17
1,000,000	Precision	0.75	0.00	0.33	0.33	0.66	0.67	0.52	0.70	0.70
	Recall	0.03	0.00	0.92	0.83	0.17	0.18	0.28	0.05	0.06

As shown in Table 2 and Table 3, Random Forest has good performance in terms of F1-score & precision at 200 communities in contrast with other classifiers. Random Forest classifier can handle the big data. This classifier do not suffer from over-fitting problem. Therefore, the overall performance of Random Forest is efficient as compared to different classifiers because we are using multiple network features. RF can select the essential features from the training dataset.

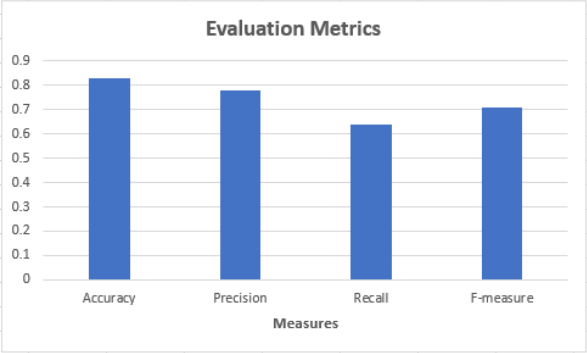

Fig. 4. Results of RF classifier on 200 communities

Different evaluation metrics are used such as recall, F1-measure, accuracy, and precision which are demonstrated as (where TP: True Positive, FP: False Positives, FN: False Negatives, and TP: True Positives) to examine and evaluate the efficiency and performance of prediction.

The results using evaluation metrics are shown in Table 2 as well in Figure 4. In Figure 4, x-axis represents the evaluation metrics and y-axis represents computed values. For our proposed method, accuracy becomes 83%, F1-score value is 71%, precision is 78% and recall is 64%.

The results highlighted in Table 2 and Table 3 refer to the optimal results of our experiments. F1-score, precision, accuracy, and recall of all classifiers are calculated.

We can see in Table 2 and Table 3, RF is performing efficiently in terms of F1-score. It is due to RF gives better performance on multiple network features in comparison with other classifiers.

4.1 Comparison with State-of-the-Art Techniques

Our experimental set up is different from the baseline papers. We have used various performance metrics to evaluate the performance of our approach for the detection of fake news: accuracy, recall, precision and F1-score. We have chosen as the baseline paper.

Fake news explainable detection was studied. Deep-hierarchical co-attention network was proposed to learn the representations of features for explainable comments/sentences discovery and for the FND.

In Figure 5, the results of different experiments are presented. The horizontal line represents different baseline techniques, and the vertical line represents evaluation metrics.The First 8 bars represent the results of 8 different algorithms used to detect fake news as addressed in by using the same dataset. Different algorithms such as TCNN-URG, text-CNN, RST, CSI , HAN, LIWC and HPA-BLSTM are used for fake news detection.

All these methods need analysis on the textual information such as user replies and tweets. The ninth bar, shown in Figure 5, represents our proposed method, which is trained on the GossipCop dataset. Our proposed method relies solely on non-textual features; our proposed method achieved comparable and efficient performance on the GossipCop dataset.

The accuracy acquired by Random Forest is more significant as compared to all other approaches used for the FND, as shown in Figure 5(a). However, in terms of precision, Proposed method has given the highest value as compared to all other approaches, as shown in Figure 5(b). The recall acquired by proposed method is lesser as compared to HAN, HPA-BLSM method, and defend method. On the other hand, recall is higher as compared to RST, test-CNN, TCNN-URG, LIWC, and CSI, as shown in Figure 5(c).

The F1-score acquired by proposed method is higher than multiple approaches that are used for the FND such as RST, test-CNN, TCNN-URG, HPA-BLSM, LIWC, HAN, and CSI. However, in terms of F1-score, RF results are lesser than the defend method, as shown in Figure 5(d). Hence, the overall performance of RF is best as compare to other methods. The results of evaluation metrics can be seen in Table 4.

5 Conclusion and Future Work

Social media is becoming popular now a days. Huge number of people consuming different type of news from different platforms of social media. Fake news is also rapidly spread through social media, which have very bad impact on society as well as on individual.

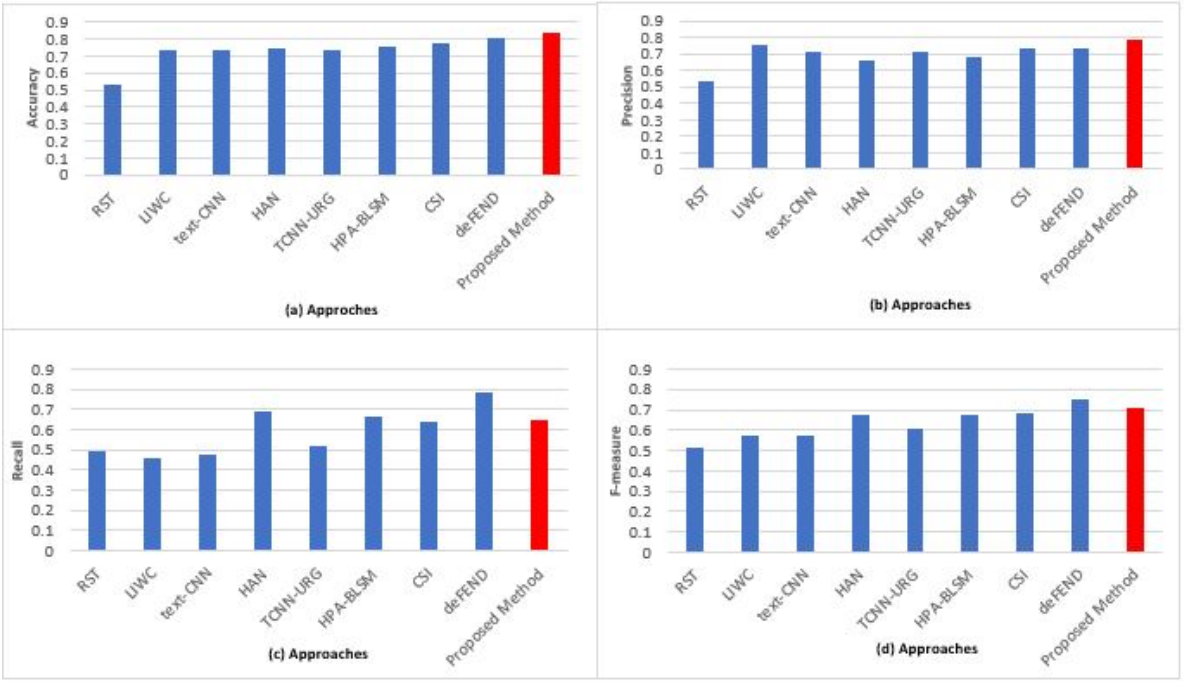

Fig. 5. Comparison of performance on the GossipCop dataset

Table 4. Results of evaluation metrics used by different techniques

Classifier	Accuracy	Precision	Recall	F1-score
RST	0.53	0.53	0.49	0.51
LIWC	0.73	0.76	0.46	0.57
text-CNN	0.73	0.70	0.48	0.57
HAN	0.74	0.66	0.69	0.67
TCNN-URG	0.74	0.71	0.52	0.60
HPA-BLSM	0.75	0.68	0.66	0.67
CSI	0.77	0.73	0.64	0.68
deFEND	0.81	0.73	0.78	**0.76**
Proposed method	**0.83**	**0.78**	**0.64**	0.71

In this paper, fake news is detected on Twitter by using social networking technique. By using social networking technique, fake news are efficiently identified. Even though this approach solely needs a number of features, which are extracted from network and does not depend on any type of textual information, this approach can achieve superior and comparable performance to other methods that need semantic and syntactic analysis.

For future work, we can identify the fake users who are involved in spreading fake news. This can be achieved either by increasing number of features or determine “universal” features that can work efficiently and effectively.

In this research, we have used network features for the FND. We can extend it by using the features mentioned above in the paper.

We can use a combination of network features and non-network features for the FND. We can identify fake users who are involved in spreading fake news. This can be achieved either by increasing the number of features or determine “universal” features that can work efficiently and effectively.

References

1. **Lazer, D. M., Baum, M. A., Benkler, Y., Berinsky, A. J., Greenhill, K. M., Menczer, F., Metzger, M. J., Nyhan, B., Pennycook, G., Rothschild, D., and others (2018).** The science of fake news. Science, Vol. 359, No. 6380, pp. 1094–1096.

2. **Pennycook, G., Rand, D. G. (2020).** The implied truth effect: Attaching warnings to a subset of fake news stories increases perceived accuracy of stories without warnings. Management Science, Vol. 66, No. 11, pp. 4944–4957.
3. **Perez-Rosas, V., Kleinberg, B., Lefevre, A., Mihalcea, R. (2017).** Automatic detection of fake news. Proceedings of the International Conference on Computational Linguistics (COLING), pp. 3391–3401.
4. **Potthast, M., Gollub, T., Hagen, M., Stein, B. (2018).** Stylometric analysis of scientific articles. Information Processing & Management, Vol. 54, No. 2, pp. 180–194.
5. **Rashkin, H., Choi, M. S., Jang, E., Volkova, S., Choi, Y. (2017).** Truth of varying shades: Analyzing language in fake news and political fact-checking. Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pp. 2931–2937.
6. **Rubin, V. L., Conroy, N. J., Chen, Y. (2016).** Fake news or truth? using satirical cues to detect potentially misleading news. Proceedings of NAACL-HLT, Vol. 1, pp. 7–17.
7. **Sauri, D., and others (2017).** BS detector.
8. **Shu, K., Mahudeswaran, D., Wang, S., Lee, D., Liu, H. (2020).** FakeNewsNet: A data repository with news content, social context, and spatiotemporal information for studying fake news on social media. Big Data, Vol. 8, No. 3, pp. 171–188.
9. **Tandoc Jr, E. C., Lim, Z. W., Ling, R. (2018).** Defining "fake news": A typology of scholarly definitions. Digital Journalism, Vol. 6, No. 2, pp. 137–153.
10. **Vosoughi, S., Roy, D., Aral, S. (2018).** The spread of true and false news online. Science, Vol. 359, No. 6380, pp. 1146–1151.
11. **Zamir, M. T., Tash, M., Ahani, Z., Gelbukh, A., Sidorov, G. (2024).** Tayyab@Dravidianlangtech2024: detecting fake news in Malayalam LSTM approach and challenges. Proceedings of the Fourth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, pp. 113–118.
12. **Zamir, M. T., Ullah, F., Tariq, R., Bangyal, W. H., Arif, M., Gelbukh, A. (2024).** Machine and deep learning algorithms for sentiment analysis during COVID-19: A vision to create fake news resistant society. PloS one, Vol. 19, No. 12, pp. e0315407.
13. **Zhou, X., Zafarani, R. (2020).** A survey of fake news: Fundamental theories, detection methods, and opportunities. ACM Computing Surveys (CSUR), Vol. 53, No. 5, pp. 1–40.

Article received on 19/05/2025; accepted on 22/09/2025.
**Corresponding author is Alexander Gelbukh.*