Fake News Detection

A Project Report
Presented to
The Faculty of the College of
Engineering
San Jose State University
In Partial Fulfillment
Of the Requirements for the Degree
Master of Science in Computer/Software Engineering

By
Abhinav Balasubramanian, Aayushi Gupta, Vineet Khatwal, Priyanka Raju
December 2020

APPROVED

DocuSigned by:
Magdalini Eirinaki
87141F0A65F14AC...

Magdalini Eirinaki, Project Advisor

ABSTRACT

FAKE NEWS DETECTION

By
Abhinav Balasubramanian, Aayushi Gupta, Vineet Khatwal, Priyanka Raju

The advancement in technology and social media platforms has eased the creation and publication of news. Publishing of News has become quicker, less expensive, and easier to access. The news outlet and social media platforms may publish fake news to build readership or even more than that as a part of psychological warfare. Fake news is a lie created out of nothing and deliberately manipulated to deceive readers. It can have irreversible social and economic consequences. For example, social media platforms like Facebook, Twitter, etc. spread fake news impacting the 2016 US Presidential election. The main challenge is to differentiate between real and fake news. This research analyzes social media platforms using community and influencer detection algorithms with users as nodes, and the relations between different users as edges. The research also attempts to build a system to catch the fake news using its language/content, that is most of the time designed to catch eyes as the aim of fake content creators is to spread it as much as possible. The influencers in the social network can be identified by estimating the centrality and prestige (importance) of a user node. The content of the news is modeled can be modelled using a variety of machine learning algorithms, most advanced of which is a Long Short Term Memory(LSTM) model. Our research aims to design a system that can detect tweets with fake news and prevent fake news from spreading further. The integrated system containing the above-mentioned approaches is a very effective tool in combating fake news on social media platform.

Acknowledgments

The authors are deeply indebted to Professor Magdalini Eirinaki for her invaluable comments and assistance in the preparation of this study.

Fake News Detection

Abhinav Balasubramanian
Department of Computer Engineering
San Jose State University
San Jose, CA
abhinav.balasubramanian@sjsu.edu

Aayushi Gupta
Department of Computer Engineering
San Jose State University
San Jose, CA
aayushi.gupta@sjsu.edu

Vineet Khatwal
Department of Computer Engineering
San Jose State University
San Jose, CA
vineet.khatwal@sjsu.edu

Priyanka Raju
Department of Computer Engineering
San Jose State University
San Jose, CA
priyanka.raju@sjsu.edu

***Abstract*—The advancement in technology and social media platforms has eased the creation and publication of news. Publishing of News has become quicker, less expensive, and easier to access. The news outlet and social media platforms may publish fake news to build readership or even more than that as a part of psychological warfare. Fake news is a lie created out of nothing and deliberately manipulated to deceive readers. It can have irreversible social and economic consequences. For example, social media platforms like Facebook, Twitter, etc. spread fake news impacting the 2016 US Presidential election. The main challenge is to differentiate between real and fake news. This research analyzes social media platforms using community and influencer detection algorithms with users as nodes, and the relations between different users as edges. The research also attempts to build a system to catch the fake news using its language/content, that is most of the time designed to catch eyes as the aim of fake content creators is to spread it as much as possible. The influencers in the social network can be identified by estimating the centrality and prestige (importance) of a user node. The content of the news is modeled can be modelled using a variety of machine learning algorithms, most advanced of which is a Long Short Term Memory(LSTM) model. Our research aims to design a system that can detect tweets with fake news and prevent fake news from spreading further. The integrated system containing the above-mentioned approaches is a very effective tool in combating fake news on social media platform**

I. Introduction

According to A. Gelfert [1] "Fake news refers to deliberate representation of false or misleading claims". While all of the studies on Fake News Detection give their own supporting arguments, we find the first definition as the most convincing and applicable. Although fake or misleading information has been around us since long, after the 2016 US presidential elections, this topic grabbed more attention. Since then, it has been an ever growing field of research. Social media platforms such as Facebook and Twitter are many times criticized for not being able to contain 'fake news', which have proven to be a major source of panic or misbelief amongst the general public.

Most of the current state of art focuses on analyzing textual data by using semantic techniques and user behavior analysis(comments, retweets,etc.) to obtain a fake news detection system [2]. While this idea seems to be an efficient approach, it does suffer from certain shortcomings. This content-based approach that uses different kinds of words and sentence structures can be surpassed by smart propagators of fake news. These propagators know how to use language effectively so that they can spread the misinformation and never get caught. Also, the approach of identifying and flagging users who have shared fake news can be misleading. Such an approach tends to have a lot of false positives. Someone might have simply shared some news that he/she came across without checking its authenticity but they might not be a regular propagator of fake news. Also, there are several fact-checking online resources, like Wikipedia, snopes.com, fact-check.org, and few others, available and accessible to every person. Despite the convenience and availability of such fact checking resources, it is a monotonous and tiresome activity to manually verify the sources for the originality of the news forecasted.

The solutions proposed above are effective in certain situations but they either lack the right context or tend to identify fake news only after it has spread to a great deal of people. This largely defeats the purpose of a fake news identification model. A better solution would be something that can identify and stop the spread of fake news in social networks at the point of origin and that detects it early. Our project aims at doing that by identifying the sources of fake news using graph mining techniques and flagging such sources to prevent fake news at the origin itself. Such a system will create a flag, but alone will not ban the account, thereby preventing false positives. To further strengthen our diagnosis, our model be using the tweet content and the topic a tweet belongs to and identifying the credibility of the tweet as well as the topic. The Natural Language Processing(NLP) model will utilize the tweet contents and the tweet topic to analyze the credibility of any tweet content.

Also, we would analyze the propagation of Fake News with the help of Influencer and Community Detection Algorithms. This could help in identifying the important users in the social graph. A combination of the above approaches will help to

capture fake news from all the important angles and we will obtain a greater accuracy than all these approaches working individually.

II. Related Work

The exchange of information around the world has become more convenient for users with the advancement of Internet accessibility. News is one such media that fosters a huge impact on society. However, the credibility of such information is extremely debatable, especially on social media. Long-established newspapers and conventional news channels have become less predominant with the growth of social media. There has been a lot of study about the spread and detection of fake news in both traditional and social media. This section contains a brief overview of various literature surveys in relation to fake news propagation and its detection.

This section has been further branched into three sub-sections. The first part includes literature that was reviewed during the topic selection phase of research. It provides an overview of Fake news and its associated characteristics. The second module includes articles that were analyzed during the problem definition phase. This section throws light on the existing problems associated with the propagation and detection of Fake news in social media platforms. The third section summarizes about existing approaches that could be employed to solve the identified problem. They were explored extensively during the system design phase of the research work.

A. Fake news and its characteristics

The impact of spreading inaccurate news is rapidly occurring, and its ability to affect millions of readers is increasing. Several concerns are raised about readers' inability to distinguish the facts as real or hoax. Thus, the article [2] aims at identifying the basic characteristics of fake news and provides means to analyze them with the help of data mining and machine learning techniques. This article recognizes that propaganda, racism, and religion are important factors that influence users to share negative stories, which leads people to question the authenticity of the content being disseminated.

The perception of an individual to differentiate a real event from that of fake news is diminishing as a result of consuming the so-called "Fake news". Corporate agencies are adversely affected in the fear of their brand names being jeopardized and as a result, have a negative impact on their products and services [3] [4]. People and organizations can also use fake news to influence people's decisions in general elections and other important events [5]. Therefore, the authors in [6] clarifies these social challenges that need to be reviewed and discussed.

B. Effects of Fake news in Social media

Since the outbreak of social media platforms, fake news has been a growing problem. Although the spread of fake news in traditional media is a problem that needs to be solved, with the growth of the Internet and network connections, social media has surpassed traditional media in spreading fake news. With an increased use of online media on a daily basis and the poor means to authenticate the credibility of the users, the detection of fake news in social media has become more significant. Hence, it becomes essential to analyze the behavioral traits of users who propagate fake news and the evolution pattern of its spread.

Article [7] tells us that the fake news audience includes a small number of unfaithful Internet users. These accounts' determination helps monitor the news shared by these accounts and determine the quality of the news shared. This article also identifies the role of social media in disseminating fake news because users are more likely to access social media than the original news site. All these incidents show that it is very important to find fake news on social media sites, because many users on social media sites are not serious and sometimes even irresponsible.

In addition to determining the credibility of the user, the authenticity of the news disseminated by the user also needs to be reviewed. N. Ruchansky et al. [8] proposed three main factors that can be used to analyze the authenticity of news. They are, the content provided by the news article, the response the article receives from the readers, which could be likes, comments or share in social media, and the source from which the fake news originates.

Referring to Twitter media, the author in [9] summarized the origin of fake news. They found that most root tweets from ordinary users contained links to untrusted news sites and were modified as they spread to Twitter through different users.

C. Fake news detection

Due to the seriousness of the problem, several methods are currently being explored to understand the spread of fake news and classify the news as fake news on social media. Some of these techniques mainly focus on detecting only the intensity of the misrepresentation of data [10]. A. Bondielli and F. Marcelloni [11] summarizes on the existence of categories of misinformation in the form of clickbait, social spam, fake reviews along with fake news and rumors. Natural Language Processing (NLP), Social Network Analysis (SNA), and Data Mining (DM) are some of the ongoing analyses for the web-based on distorted news. Content-based and Context-based feature extraction is employed to collectively obtain data to analyze the existing classification algorithms such as Support Vector Machine (SVM), Random Forest, Recurrent Neural Network (RNN) and Convolutional Neural Network (CNN) to identify fake news [11] [12].

Based on the article [11], below section summaries the various approaches based on the field of research:

1) Deep Learning: Since user and news content analysis are two important aspects of dissemination and detection of fake news, most of the current literature focuses on applying machine learning models to these aspects.

On some text classification data sets, deep learning models provide higher accuracy than ML classification algorithms.

Authors of [13] have applied CNN on the LIAR dataset from PolitiFact.com and achieved higher fake classification accuracy. They also demonstrated a hybrid model that combines the context features and text features of news articles using CNN, and has achieved significant improvements in the detection of fake news.

Article [14] aims to apply recursive and convolutional neural networks to analyze changes in user characteristics during the spread of fake news. The devised model was tested on three datasets (Twitter15, Twitter16 and Weibo) and it was found to be performing better than the previously available fake news detection systems. The scope of this article includes analyzing the characteristics to detect users who are more prone to spread or believe in fake news.

Authors of [15] explain a deep learning technique called Geometric Deep learning. According to this paper, fake and real news spread differently on social media and this could be harnessed for automatic fake news detection.

2) Text Mining: The authors of the paper [16] have used a text mining approach to process unstructured textual political news data from multiple sources into a structured data set. A comparative study is carried out across 23 supervised machine learning algorithms on the preprocessed dataset. Their results show that the decision tree algorithm can provide better classification accuracy of fake news. Although multiple algorithms are evaluated, the entire data set falls under one category of news, politics and so results can be biased features.

Since social media and networks make up a great deal of fake news in modern days, several techniques that focus not only on the content of the article but also on the user interactions have been devised off late. Article [8] proposes a Capture Score Integrate Model (CSI) which aims at considering the source, content and response of the news article and it was found to be more successful in detecting fake news than other previously proposed models, which worked on analyzing the text present in the news article. This model's performance was due to the consideration given to the spread of fake news across social media networks. The 'Capture' aspect of the model focuses on extracting the temporal representation of the news articles while the 'Score' aspect of the model focuses on providing a score to the user based on the social interactions and behavior. These two models are then integrated together in the 'Integrate' aspect of the model to achieve a powerful model that focuses on both news and user analysis in social media networks.

Although Natural Language Processing (NLP) based methods work well, they lack a social context for fake news, and smart writers may surpass them. M. Andrews and G. Vigliocco of [17] propose the 'Hidden Topic Markov Model', which is an extension of the Bayesian bag-of-words model. This model will help us to identify the hidden intentions of fake news authors based on the hidden topics that the news actually talks about. Otherwise, such topics may not be noticed and may have the desired effect of delivering false information to millions of readers.

3) Graph Mining: Since most of the social media content and users are interrelated by their interactions in the form of retweets, likes, follows, comments, etc., a social graph can be constructed using the social media and can be analyzed using Graph Mining techniques.

Many business organizations nowadays use social media for advertising due to increased reach and reduced cost of advertising when compared with the traditional media. The author in [18] proposes the use of graph mining techniques on social media to identify influencers in the social networks who could be employed by business organizations for promoting their products. It further identifies that a social media network built on the basis of mentions is capable of identifying influencers much more efficiently than a social network built on the basis of followers. The 'mentions' aspect of the social network is capable of capturing information about the intensity of user interactions that the 'follows' aspect fails to capture. This matches with our project's aim of using graph mining and community detection to identify how strongly an user is connected with other users in the network and how widely a fake news spread by such users propagates over social media.

Pudjajana et al. [19] uses social network analysis to identify influencers in social media. The Facebook hoax dataset is used as input, and various centrality measures are applied to the graphs derived from the input dataset. Each centrality measure is then allocated a weight that could be used to find the final weighted SNA- value that could be used to rank the influencers. Hierarchical modelling is then applied on the ranked influencers to predict the spread of the hoax on Facebook.

Article [9] describes the detection of fake news using the evolution tree, which is basically network analysis. It provides information about 'root content', 'content hierarchy' and 'content evolution'.

However, according to the article [3], the authors convey that all the above mentioned methodologies fail to precisely identify fake information from the corpus amount of data that is constantly being updated by the internet and web services.

III. Problem statement

The transmission of news and information has witnessed a lot of evolution in this digital era. Social media has overtaken television, radio, newspapers and other means of communication. Social media provides the facility of broadcasting or disseminating news immediately. People who see the news on social media will involuntarily think that the news is fact, provided that the news is shared by close circles. There are a lot of fake social media accounts who try to imitate legitimate accounts. Because social media is used by people of all ages, these accounts can win the trust of certain age groups [27].

As a result, the spread of fake news in social media is on the rise. Fake news about politics and public figures creates a stressful environment for many people. Thus, the fake news detection and propagation analysis is the current need of the hour. Existing fake news detection systems either focus on content-based models or focus on fact-checking models. Some

systems focus on identifying the source of the problem by analyzing the spread of fake news. Combining the results of the content-based and the propagation-based methods may result in a more effective and accurate model, which has not been studied much.

The current propagation detection methods used to analyze misinformation mainly focus on the relationship and interaction between the source reader and the target reader. While these play an important role, they miss out on the interpretation of the various important and crucial factors like longevity of a user's social media handle, activity of the user during his social media lifetime. In addition, the tweet content can be checked to see if it can help us flag a tweet as real or fake news. A combination of the above two approaches can perform better to form a more efficient fake news detection system

IV. Proposed solution

A. Dataset

1) Credbank: CREDBANK dataset - This dataset consists of tweet data collected from October 2014 to February 2015. It first classifies tweets into events and non-events. Event tweets will be the tweets containing news. Further, topics are extracted from these tweets and then these topics are rated by 30 raters on a scale of 5 from -2 to 2. -2 meaning less credibility and 2 meaning most credibility. The data is spread across the following four files:[1]

1) *Streaming Tweet File:* This file contains more than 169 million records beginning from 10-Oct-2014 and 26-Feb-2015. The streaming tweets contains the data which was scattered around different time bins when these were collected.
2) *Topic File (Event/Non-Event related):* The topic file contains more than 62,000 records for different topics. The file has 3 terms for each topic. Each term is a representation of the top three terms corresponding to running topic modelling (LDA) over the streaming tweets.
3) *Credibility Annotation File:* The file has more than 1300 events and each record is assigned a credibility ratings of it being real or fake, this file plays a very important role in classifying the text as real or fake.
4) *Searched Tweet File:* This file has more than 80 million tweets and provides us with the tweet id for different topics. We can use the tweet id to extract the tweets using tweepy API and use the data with the credibility ratings for the topic to train our model.

From these we have used this dataset to create our content based model. Based on the tweet IDs derived from this labelled dataset, we extracted tweets using Python's tweepy library.

To extract the dataset for the tweets, we had to access twitter data using the twitter dev account. 'Tweepy' library in python helped us access the data by providing access_token, access_token_secret, consumer_key and consumer_key_secret for a specific developer account. We use cred_event_SearchTweets.data and cred_event_TurkRatings.data files from the CREDBANK dataset to access the data related to the tweets. We used the topic_key and Cred_Ratings from cred_event_TurkRatings.data to create a list of hashmaps where the list elements were a combination of the key as the topic_key and ratings as the average of all the Cred_Ratings for a specific tweet. If the average rating of the topic_key was ¿ 0, we classified the topic as 'Real' else as 'Fake'. To extract the tweets we relied on cred_event_SearchTweets.data file to provide us the tweet id to fetch the tweet using the tweepy library. We mapped the file's data with the topic key and fetched the tweets for all the topics in the list created from the file cred_event_TurkRatings.data. This allowed us to fetch the topic key, tweet id, tweet content and the average ratings for a tweet and save as a csv file. To overcome the challenge of fetching only 180 tweets per 15 min as set by twitter api rate limiting, we deployed the script to EC2 instance so that tweets could be downloaded for a long period of time without any hindrance.

2) Russian Troll Dataset: As a result of Russian influence in the 2016 US Elections, twitter suspended more than 3000 twitter accounts which were believed to be operating as social media troll accounts. The tweets and other data related to these accounts were also deleted. However, a team at NBC News successfully reconstructed a small subset of the deleted data and used the data to research as to how these accounts were used during the elections. The dataset comprises two CSV files and is available to the public for their analysis. [2]

1) *tweets.csv :* This file has the data related to the user's tweet.
2) *users.csv:* This file has the data related to the user's account.

3) FakeNewsNet: The data set considered is from the FakeNewsNet[3] repository. This data set contains data belonging to various aspects of Twitter social media, such as tweets, user profiles, user followers and followers. All these attributes can be divided into two categories: false and true, which helps to analyze the difference between fake news and real news. The data extracted are from the PolitiFact [4] and Gossip Cop[5] fact checking websites.

The twitter user profiles like username, bio, location, followers and following of a user, tweet and retweets can be obtained using the repository. The library provides a configuration, which can be tuned to extract the required features. This dataset can be used to analyze user graphs to detect influencers in Twitter using social network analysis (SNA) and community detection.

[1]CREDBANK Dataset Link: Mitra, Tanushree, and Eric Gilbert. "CREDBANK: A Large-Scale Social Media Corpus with Associated Credibility Annotations." Ninth International AAAI Conference on Web and Social Media. 2015.

[2]Russian Trolls Dataset Link: https://www.kaggle.com/vikasg/russian-troll-tweets

[3]FakeNewsNet Dataset Link: https://github.com/KaiDMML/FakeNewsNet

[4]Gossip Cop Dataset Link: https://www.politifact.com/

[5]Gossip Cop Dataset Link: https://www.gossipcop.com/

The representation of twitter attributes and their relationship with each other in the social network is as shown in Fig.1.

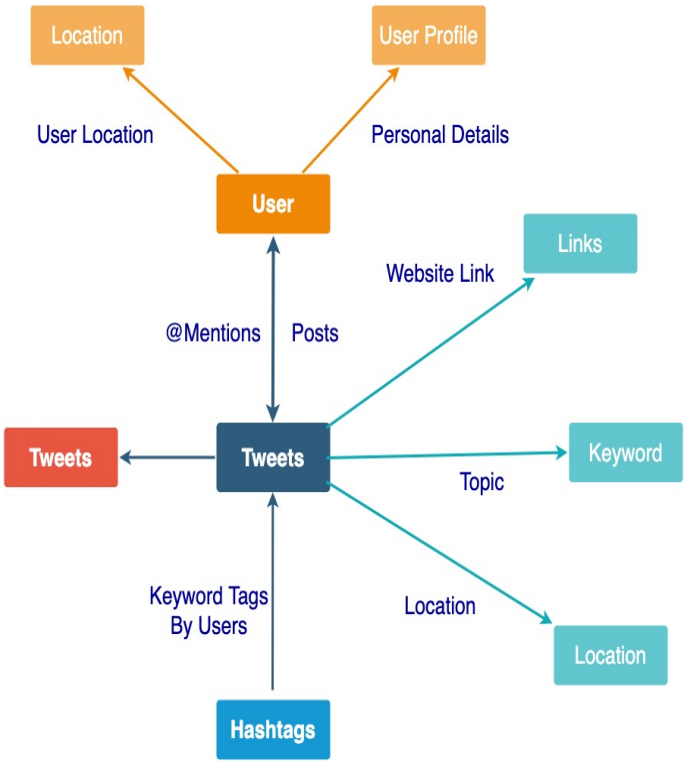

Fig. 1. Twitter Components

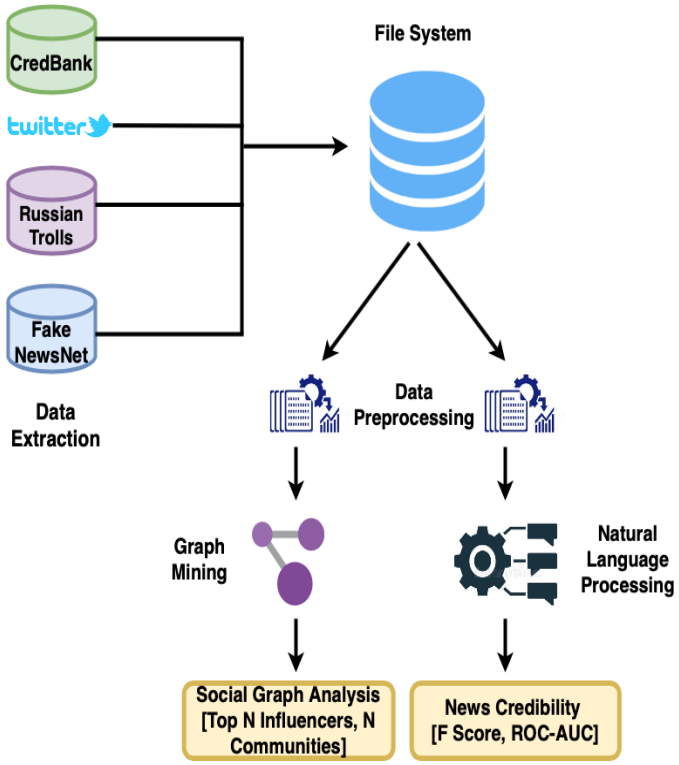

Fig. 2. Fake News Detection Architecture

B. Architecture

The Fig.2. shows the overall architecture of our Fake News Detection System. The data scraped from three main sources: CredBank, FakeNewsNet, Russian Trolls and Twitter (using python's tweepy library) are accumulated over time in a common data storage, from where it is retrieved for training the models. User-based features such as *user profile*, *following*, *followers* and *user activities* are extracted from the FakeNewsNet to perform analysis on the spread of the fake news using social graph mining. This includes identifying the important aspects that could be added as the weights to the edges in social graph, identifying the influencers who could be the potential source or root cause of the fake news, community detection to identify the behavioural aspects of the group of users considered for the analysis to identify potential subgroups within them which helps in understanding the speed at which the fake news propagates within a particular sub-group and also understand the relationship that exists between the subgroups.

Tweet ids and corresponding labels are retrieved from CredBank dataset and tweets corresponding to these Tweet ids are extracted from Twitter using python's tweepy library. For each topic/event, we have extracted 100 tweets, which gives us enough data for every topic. The labels for the tweets are given manually by 30 labelers. We use an average of these labels as our final label. As a pre-processing step extra characters are removed from the tweet and then it is prepared by converting to numerical indexes so that it can be used by our machine learning model. This processed data is used to train our model using python's keras library. The LSTM model identifies the content features and uses them to predict the probability of a tweet being fake or real.

V. Methodology

A. Predicting Fake News Propagation in Twitter

Since our project aims at identifying the fake news on Twitter which is a social media platform, Link analysis is considered as one of the effective approaches for detecting fake news propagation through social media. Social graphs are the graphs that could be constructed from social media data. Social media generally involves a huge set of users interacting with each other by means of some action (comment, like, share, retweet, mention, follow). By making all these users as nodes and their interactions with other users as edges, the dataset collected from the social media could be made a social graph.This section is further divided into the two major subsections: Introduction to Social Graph and Analysis and Fake News Propagation Analysis using Social Graph

1) *Introduction to Social Graph and Analysis*
This research aims at identifying and analyzing two important aspects of social graphs: Influencers and Communities within the network. This section provides an overview of social graph components and the need for influencer and community detection in social networks.

a) *Twitter Social Graph components*

i) *Nodes:* A node is basically an entity whose interaction forms the basis of a social graph. Since our project focuses on data extracted from Twitter, Nodes will be individual user profiles on Twitter.

ii) *Node Attributes:* Node attributes represent the properties associated with each individual node. For Twitter, this may be attributes associated with the user's profile, such as name, ID, user profile, location, etc.

iii) *Edges:* The edges in the social graph represent interactions that occur between nodes in the graph. For Twitter, three main types of interactions occur between different user levels, namely: user-to-user interaction, user-to-channel:

- User to User Interaction
- User to Channel(i.e. Topic) Interaction
- Channel to Channel Interaction

iv) *Edge Attributes:* Edge attributes define attributes associated with edges, and these attributes can be used to define their corresponding node relationships. Most Twitter interactions occur in one of the following ways, making it the best choice for edge attributes.

- Mentions
- Retweets
- Follows

v) *Edge Weights:* Edge weights allow us to rank different mentioned edge attributes by assigning high weights (higher priority) to certain edge attributes and low weights (lower priority) to certain edge attributes. In Twitter, the edge weights can be calculated based on user profile attributes like active years, number of followers and following, favorite count, tweet count and account validity. These defined weights can be stored in the network as part of the topology (information can be used as part of the network structure) or semantic structure.

b) *Influencer Detection*

For understanding the spread of fake news in a social network like twitter, it becomes really important to identify the source/origin of fake news detection. Social network analysis could be employed on the social graph to find the influential user nodes that can contribute towards the origin and spread of fake news detection. Some social network analysis metrics that can be considered are

i) *Degree Centrality:* Degree Centrality measures how well connected a particular node is with all the other nodes in the network. The formula for

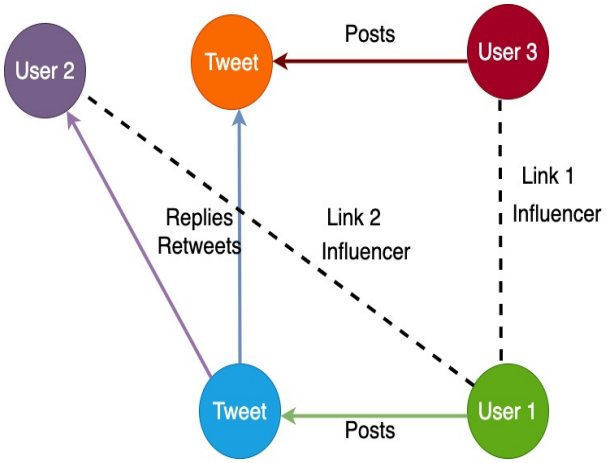

Fig. 3. Influencer Link Analysis

degree centrality is given by

$$DC(i) = \sum_{j=1}^{n} a_{ij} \tag{1}$$

In the above equation a_{ij} is equal to 1 if an edge exists between the nodes *i* and *j* and a_{ij} is equal to 0 if no edge exists between the nodes *i* and *j*. The above computed degree centrality can be normalized as follows

$$nd_i = (\sum_{j} a_{ij})/(n-1) \tag{2}$$

In the above equation (n-1) indicates the maximum number of edges the particular node in consideration has with all the other nodes in the network. After normalization, the maximum value for degree centrality becomes 1 and the minimum value becomes 0. The degree centrality could further be split up into indegree and outdegree centralities. For a_{ij} in indegree centrality the direction of edge is from *j* to *i* and it is the vice versa in out degree centrality. The indegree centrality provides information about a user's popularity in social media (follows and mentions from other user nodes to this nodes indicates popularity). The outdegree centrality provides information about the intensity of interaction with the other users (Mentions and follows from the considered user to other users in the network indicates the interaction capabilities of the considered user).

ii) *Closeness Centrality:* Closeness centrality indicates how closely a particular node is located to all the other nodes in the system. Vertices with high values of closeness centrality could

be considered as influencers. The formula for closeness centrality is given by

$$CC(i) = 1/\sum_{j=1}^{n} d(i,j) \tag{3}$$

In the above equation, *n* indicates the total number of vertices and *d(i,j)* indicates the minimum distance of the node *i* from node *j*.

iii) *Betweenness Centrality:*

$$BC(i) = (\sum_{i \neq j \neq l} g_{jl}(i))/g_{jl} \tag{4}$$

The betweeness centrality is used to find the minimum number of edges that pass through the particular node considered. In the above equation, g_{jl} indicates all the edges that pass from node *j* to *l* and $g_{jl}(i)$ indicates all the edges from node *j*to node *l* that pass through the intermediate node *i*.

iv) *Page Rank:*

$$PR(p_i) = (1-d)/N + d(\sum_{p_j \epsilon M(p_i)} PR(p_j)/L(p_j)) \tag{5}$$

The PageRank aims at identifying the importance and influence of a node in a social graph by identifying the number of other nodes in the network to which it is connected in a recursive fashion.

All the above mentioned centrality measures taken from [18] are applied on the social media graphs and can result in different ranks for each user node in social networks. The final ranks of each user node in a social network is determined by the majority voting obtained from the individual centrality measures.For example, if three out of the four centrality measures provides a high score to an user, that user is given a higher priority. The ranked user nodes could be split into two sections with the first section having a very small percentage of nodes in the system and the second section having the rest of nodes in the system (2.5%:97.5% is the split up used by the literature [19]). The 2.5% of the ranked user nodes are identified to be the influencers in the social networks.

c) *Community Detection*

The social network is all about connection and interaction between users. In addition, various people communicate with each other on social media, and some of them may have the same affinity for certain topics. Therefore, dividing users into smaller subgroups helps to analyze and identify how quickly certain tweets spread within a smaller group and the transition of certain tweets from one group to another. Therefore, all of the above mentioned aspects can be considered by implementing community detection. Community detection along with influencer detection would not only help in identifying the most important users in the network but would also help in estimating the speed at which the tweets propagate across the networks and how different sub-groups interact with each other. In the future, this will also help identify the behavioral aspects of new users in the network by implementing unsupervised clustering techniques. Few of the popularly used community detection algorithms are explained briefly below with their respective pseudo-codes.

Input: $G = (V, E)$.
Output: the result of community detection.
(1) Initialization: assign a unique label to each node in the network, $c_i(0) = i$.
(2) Iteration of label propagation:
(a) Set $t = 1$;
(b) Arrange the nodes of the network in random order, and store the results in the vector X.
(c) For each node $v_i \in X$, let $c_i(t) = f(c_{i1}(t), \ldots, c_{im}(t), c_{i(m+1)}(t-1), \ldots, c_{ik}(t-1))$, where $v_{i1}, \ldots, v_{im}$ are neighbors of v_i those have already been updated in the current iteration and $v_{i(m+1)}, \ldots, v_{ik}$ are neighbors those are not yet updated in the current iteration. The function f here returns the label that the maximum number of its neighbors has. If multiple labels simultaneously have the maximum number, then randomly select one of them to assign to the node.
(d) If the label of every node does not change anymore, then stop the algorithm. Else, set $t = t + 1$ and go to Step (b).
(3) Community division: divide all nodes share the same label into a community; the type of labels indicates the number of communities.

Fig. 4. Label Propagation Pseudo-code [20]

i) *Label Propagation Algorithm:* In the label propagation algorithm, each of the nodes in the graph is assigned a unique label initially. The labels are propagated through the graph for various iterations and at the end of each iteration, the label for a particular node is changed based on the labels present on the majority of neighboring nodes. This process is repeated till convergence is achieved or the maximum iteration specified is reached. This is based on the principle that a label will propagate more easily in a densely connected graph resulting in formation of multiple sub communities within the graph [23]. The Fig. 4. shows the pseudocode of Label Propagation Algorithm.

ii) *Louvain Modularity Algorithm:*

Louvain is a hierarchy based community detection technique where multiple nodes are converted into a single node (community) recursively such that the modularity score for a community formed is maximum. This modularity score defines the quality with which a node is placed within a community when compared to a random network. This also facilitates the creation of overlapping communities where multiple smaller sub-communities could be grouped together to form a larger community

```
G the initial network
repeat
   Put each node of G in its own community
   while some nodes are moved do
      for all node n of G do
         place n in its neighboring community includ-
         ing its own which maximizes the modularity
         gain
      end for
   end while
   if the new modularity is higher than the initial
   then
      G = the network between communities of G
   else
      Terminate
   end if
until
```

Fig. 5. Louvain Modularity Pseudo-code [21]

```
STRONGLY-CONNECTED-COMPONENTS (G)
1  call DFS (G) to compute finishing times f[u] for each vertex u
2  compute G^T
3  call DFS (G^T), but in the main loop of DFS, consider the vertices
         in order of decreasing f[u] (as computed in line 1)
4  output the vertices of each tree in the depth-first forest formed in
line 3 as a
         separate strongly connected component
```

Fig. 6. Connected Components Pseudo-code [22]

[24]. The Fig. 5. shows the pseudocode of Louvain Modularity Algorithm.

iii) *Connected Components Algorithm:* The connected component algorithm uses depth first search and is one of the earliest graph algorithms that operates in linear time. It aims at breaking down the graph into its strongly connected components. These strongly connected components do not have any connections to the nodes outside the community formed [25]. The Fig. 6. shows the pseudocode of Connected Components Algorithm.

2) *Fake News Propagation Analysis using Social Graph*
The methodology adapted for Fake News Propagation Analysis is shown in the Fig. 7. For constructing the social graph, the source users and their followers were extracted from FakeNewsNet dataset. The attributes of these users were extracted from Tweepy API. The final source-target data comprises around 500 thousands data entries. However, due to the computational constraints of the Neo4j database, the below methodology has been applied on approximately 16000 records, where only the top 25% of followers were considered for every source user. The ranking of the followers were computed based on the quantile ranges of their profile features as shown in Fig. 8
All the above mentioned steps were performed for both the Fake and Real News with around 1 million data entries(500 thousands each).

a) *Twitter Social Graph components*

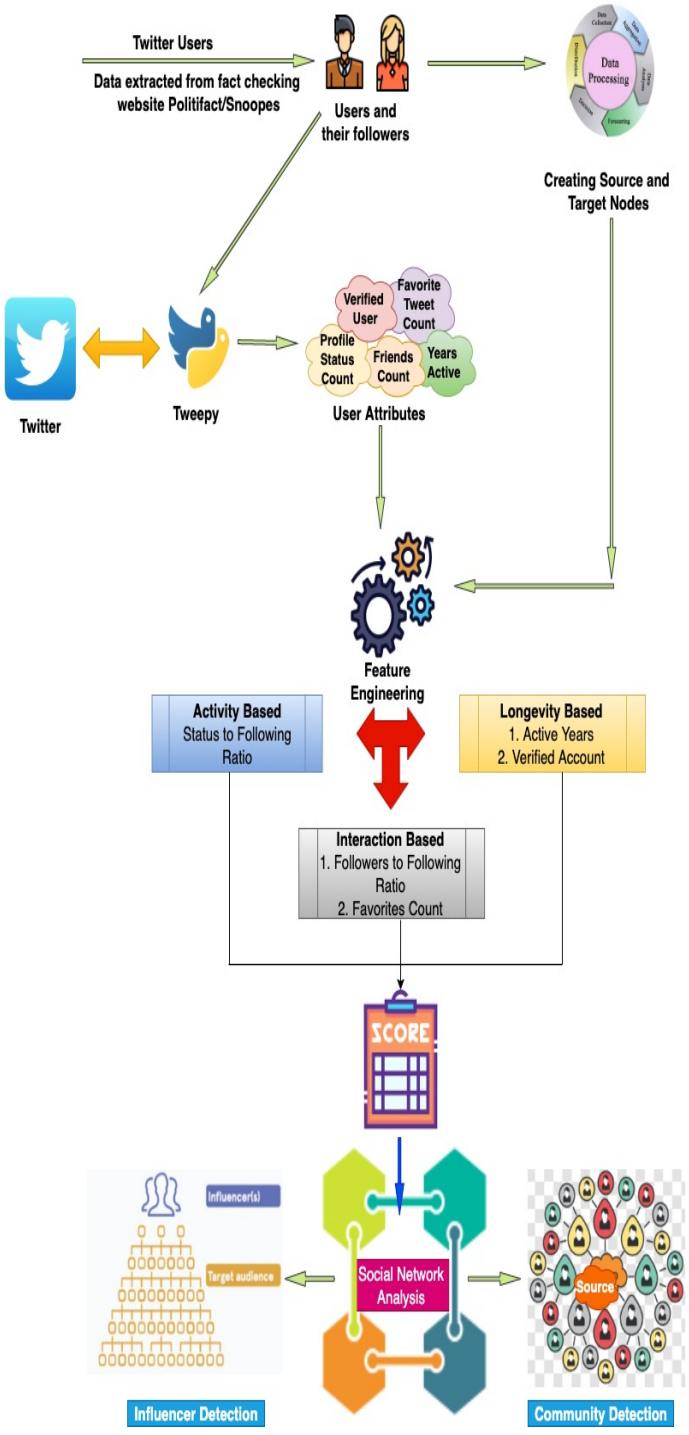

Fig. 7. Fake News Propagation Analysis Methodology

i) *Nodes:* The Nodes used in the fake news propagation analysis are the users and their followers extracted from the FakeNewsNet dataset.

ii) *Node Attributes:* The node attributes are the user profile characteristics like followers count, following count, favorites count, status count, profile creation time, account verification status

extracted from the Tweepy API.

iii) *Edges:* The twitter interaction that was taken as the edge for the social network analysis is *followed by*.

iv) *Edge Weights:* The weights added to the edges plays a very important role in determining the influencers and communities. The metrics/attributes generally used as weights in social networks fall under three main categories:

- Activity Based
- Interaction Based
- Longevity Based

Each of these metrics have its own set of advantages. Our approach aims at combining all these weighting techniques by assigning score to each of these methods. The scores are computed based on the quantile ranges for each of the user attributes. The figure shows the scoring criteria for each of the user attributes.

b) *Influencer Detection*

The centrality measures that has been used for identifying the influencers are:

- *Degree Centrality*, due to its ability to identify the number of connections a node shares with other nodes in the social network.
- *Page Rank*, due to its ability to measure the influence of a node and its relationship with other nodes recursively.
- *Between Centrality*, due to its ability to identify the control a node on the propagation of information within a social graph.
- *Closeness Centrality*, due to its ability to identify the efficiency with which a node can propagate information within a social graph.

Once all the centralities are computed for all the nodes in the graph, a majority voting process is conducted by comparing the values of individual centrality measures. The priority given to the individual centrality measures are in the order of:

- Pank Rank
- Betweenness Centrality
- Closeness Centrality
- Degree Centrality

The top 2.5% of the users are considered to be the influencers (400 users for a dataset of 16000 rows).

c) *Community Detection* The community detection algorithms implemented comprises of:

i) *Label Propagation:* Label Propagation is considered due to its ability to quickly deduce subgroups from the given data with the help of semi-supervised learning. This makes it more suitable for real-time social media applications.

ii) *Louvain Modularity* The Louvain Modularity algorithm is considered due to its ability to work well with large datasets with densely populated sub-groups. This makes it more suitable for detecting communities consisting of a large number of users in social networks (such as Twitter). Creation of intermediate communities was also enabled in this algorithm to identify overlapping clusters.

Attributes	Category	Quantile Range	Score (percentage)
Active years	Longevity Based	Min to Quantile 1	5
		Quantile 1 to Quantile 2	10
		Quantile 2 to Quantile 3	15
		Quantile 3 to Max	20
Verified status	Longevity Based	False	0
		True	30
Status to Following Ratio	Activity Based	Min to Quantile 1	5
		Quantile 1 to Quantile 2	10
		Quantile 2 to Quantile 3	15
		Quantile 3 to Max	20
Followers to Following Ratio	Interaction Based	Min to Quantile 1	5
		Quantile 1 to Quantile 2	10
		Quantile 2 to Quantile 3	15
		Quantile 3 to Max	20
Favorites Count	Interaction Based	Min to Quantile 1	2.5
		Quantile 1 to Quantile 2	5
		Quantile 2 to Quantile 3	7.5
		Quantile 3 to Max	10

Fig. 8. Fake News Propagation Analysis Edge Weight Scores

iii) *Connected Components* The Connected Components algorithm is considered due to its ability to identify the groups of users who share a common interest or affinity to a source user or a tweet among other users in a social graph.

B. Identifying fake news from content

As another part of our fake news detection system, we are using the tweet content to identify if a tweet containing a news is real or not. It is important here to understand the purpose of spreading fake news, which is to gain politically, economically (click bait) or socially. Thus fake news propagators want such news to spread as much as possible, otherwise their purpose will not be served. And thus fake news content is emotion generating and very catchy most of the time. Many times this content is created systematically using a specific structure. Using this part of our system , we will explore such "systematic" features from the tweet content.

To do this, we are using a Long Short-Term Memory (LSTM) model, which is designed to model temporal sequences of words and their long-range dependencies more accurately than conventional RNNs. LSTM [26] models contain memory units or blocks in the hidden layers. These blocks store the temporal state of the network and also contain special gates or multiplicative units which play a major role in controlling transmission of information. Each memory block contains an input gate and an output gate. The input gate has main control over the input activations and their flow into the memory cell. On the other hand, output gate has the ability to control the how the cell activations flows into the remaining network. The memory units in LSTM has a unique characteristic which condenses the content stored previously in the memory and adds the new information to it which leads to better results, this provides LSTM great advantage over the common recurrent unit.

We combined the tweet data from the above two sources to obtain a balanced dataset with approximately 40 thousand records each of fake and real news. Before creating the model, we needed to bring this data to a format suitable to be fed into the model for training. First, we removed extra characters(length ¡ 3 characters), hyperlinks, stopwords and converted each tweet to a list of words. To explore the two labels of data, we created word clouds, highlighting important words in the two categories.

Using Keras Tokenizer, we then converted the above text to vector sequences and created indexes for all the unique words. Using the maximum length of a tweet we padded shorter vectors to obtain vectors of equal lengths. Now using 80% of our data(training set), we trained a Glove embedding vector. Word embedding algorithms can figure out relationships from the text data. They use the idea of context and learn by seeing what word occurs near other words. GloVe is an unsupervised learning algorithm which helps us in capturing the global statistics as well as local statistics of a corpus. It helps us in representing word in the form of vectors. With glove, we get a linear substructure for the word vector space as we train

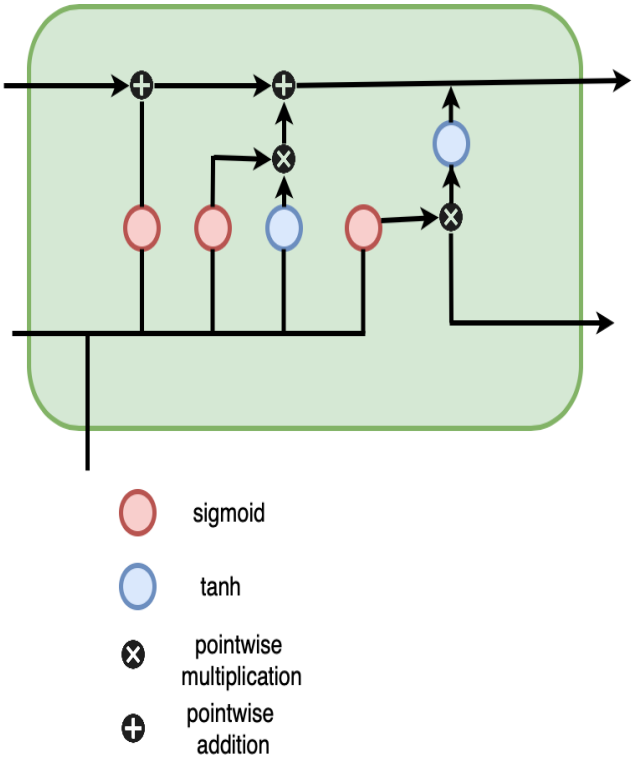

Fig. 9. LSTM Model

the model on an an aggregated global word-word co-occurence statistics from a corpus.

As we don't have sufficient corpus data, we used a pre-trained 100 dimension glove vector. We tested with a couple of pre-trained word vectors - glove.twitter.27B.100d.txt and glove.840B.300d.zip and chose the former as it performed better for our data set. An embedding matrix of 74707 x 100 was obtained.

Our LSTM model Fig. 10 is a sequential model, i.e. each neuron of a layer gets input from each neuron of the previous layer. It consists of four layers including one embedding layer that uses the above embedding matrix, an LSTM layer (with 60 neurons), a GlobalMaxPool1D layer that progressively reduces the spatial size of the representation to reduce the amount of parameters and computation in the network, a dropout layer that randomly sets input units to 0 with a frequency of rate(0.1 here) at each step during training time, which helps prevent overfitting. Next, a dense layer with 50 neurons and ReLu activation function is used and the output layer is again a dense layer but with 2 neurons(as this is a binary classification problem) and sigmoid activation(since sigmoid outputs a value between 0 and 1).

We are using an Adam optimizer to train our model [Fig. 10]. Adam combines the advantages of two other extensions of stochastic gradient descent:

1) *Adaptive Gradient Algorithm (AdaGrad):* AdaGrad allows a learning rate for each parameter, which helps us improve the performance of sparse gradients.

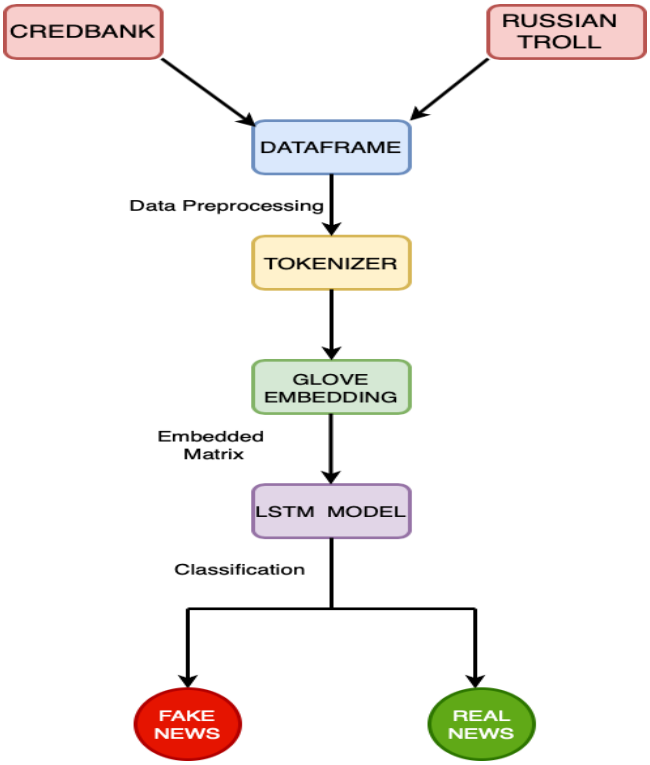

Fig. 10. LSTM Model Approach

2) *Root Mean Square Propagation (RMSProp):* Also, it is also possible to maintain the learning rate of each parameter. These rates are used based on the recent gradients for the weight.

Adam realizes the benefits of both AdaGrad and RMSProp. RMSProp relies on parameter learning rates on the basis of the average first moment or mean but Adam will also leverage the average value of the second moments of the gradients or the uncentered variance. As a result this algorithm calculates an exponential moving average of the gradient and the squared gradient.

The loss function used for compiling the model is cross-entropy. Since our model is a binary classifier, we are using a categorical cross entropy, in keras this is provided as "categorical_crossentropy". It minimizes the distance between two probability distributions - predicted and actual. Cross Entropy is defined as-

Cross Entropy is defined as-

$$J(w) = -[\sum_{n=1}^{N} y_i log(y_p) + (1 - y_i) log(1 - y_p)]/N \quad (6)$$

where-

w: weights of the neural network.

y_i: is the true label.

y_p: is the predicted label.

Once the model is constructed, we trained it using various combinations of batch size, dropouts and number of epochs and finally a batch size of 128 trained with 10 epochs gave the best result.

Node	Degree Centrality	Page Rank	Betweenness Centrality	Closeness Centrality
19672966	412	167.8201207	167.8201207	0.2251199232
21237884	378	165.8195036	165.8195036	0.1836982333
20974554	386	158.559881	158.559881	0.2232423176
22024242	375	158.03263	158.03263	0.2283327927
20984200	358	157.9282189	157.9282189	0.1663397921
18089606	351	152.3735109	152.3735109	0.180462982
19664367	344	144.1751621	144.1751621	0.2140967153
20454503	359	141.9790613	141.9790613	0.222621043
20974456	319	141.1663721	141.1663721	0.1519721077
20189103	318	129.536741	129.536741	0.2283327927
21508174	271	117.5290013	117.5290013	0.1842318765
20718719	253	111.8786056	111.8786056	0.1636978815
19408187	251	110.407103	110.407103	0.172943691
20197657	267	105.5688746	105.5688746	0.1969752085
15218168	225	99.54188712	99.54188712	0.1694305382
786560	224	99.05558889	99.05558889	0.1753736921
791638	223	97.60186227	97.60186227	0.1776350652
8396592	217	96.36345174	96.36345174	1
31471559	224	95.28318185	95.28318185	0.2046544757
19413876	218	94.18530783	94.18530783	0.1897439353

Fig. 11. Top 20 Influencers for Fake News

VI. Results

A. Evaluation Methodology

1) Propagation of Fake News:

a) *Top N Influencers in Twitter:* In order to predict the credibility of a news posted in social media, the credibility of the user who posted it needs to be analyzed. One important aspect that helps in understanding the credibility of the user is the longevity of his social media profile. Another important feature that adds more confidence to the longevity of a user account is the volume of activities performed by him throughout his account lifespan. Additionally, the number of followers and following that would be proportional to the user account shelf-life would add more genuinity to a user profile. Thus, all these aspects could play a major role in understanding the propagation of fake news and identifying the root users from whom the fake news originates. Therefore, ratios computed based on these factors are used as weights to the edges in the social graph constructed based on the social media interactions between users.

The top 20 influencers from the list of 400 influencers identified using the Influencer detection methodology performed for both Fake and Real News users is shown in the figures [11][12].

From the Fig.11 and Fig.12, we can see that the betweenness centrality values for the real users are exponentially higher than that of the fake users.This has been achieved because we have

Node	Degree Centrality	Page Rank	Betweenness Centrality	Closeness Centrality
57419364	343	142.0523125	7718611.921	0.2079099437
16255515	235	99.98408311	7562899.863	0.2035352719
1426071613	224	99.45528214	24976	1
627845910	224	99.23058991	3064020	0.1281323134
454826519	223	99.01363314	24753	1
117450420	224	98.8893086	3458391	0.1565569231
12849302	224	98.63145361	3224372.812	0.1691351543
15993745	224	98.44213372	3078206.935	0.1601146648
10178	224	98.43875614	6214982.665	0.1703833995
24907632	225	98.39113428	4286240.186	0.1837257113
20719692	224	98.32068279	3255922.258	0.1667649856
3032	224	98.21905196	4315292.895	0.1722135886
114563326	222	96.82940673	3835445.321	0.1763081192
21765306	223	96.38668474	3997980.449	0.1892996242
18615244	222	95.30369497	9746000.612	0.1903348585
1667244020	222	94.38564013	5351975.774	0.2033858487
22296258	220	93.51597339	4529136.916	0.1934934138
24645871	223	91.26625818	3907636.107	0.1937966059
36972302	210	91.05378707	3747334.392	0.1881937368
2152741188	205	90.83190523	2804898	0.1362165777

Fig. 12. Top 20 Influencers for Real News

taken into account the user profile attributes. From the dataset, we could observe that most of the users who share real news have a verified account and regular activities in their profile with substantially more followers than people who share fake news. This increases the betweenness centrality score for the users who share real news.

However, while looking at the PageRank metrics results, we could observe that the user who propagates fake news has higher importance than the user who propagates real news. This denotes that fake news is capable of grabbing more attention in social networks when compared to real news.

b) *Identified communities:* Even though the influencer detection provides a means to identify the root source of fake news propagation, it doesn't provide any information about the similarities and preferences between users. Understanding such similarities and differences would help in identifying the pace at which fake news propagates within a group of similar users. This would help in preventing the transmission of unwanted information. This can be achieved with the help of louvain modularity, label propagation and connected components techniques. The Fig.13 shows the comparative results of the label propagation, louvain modularity and connected components algorithms for both real and fake news.

From the Fig.13, we observe that the connected components algorithm is capable of forming a community with larger size in both fake and real news in comparison to the LPA and Louvain Modularity. This could be because the users within the

Community Algorithm	Type of Users	Total Number of Communities	Size of the largest Community	Size of the smallest community	Number of overlapping communities formed
Label Propagation	Fake	47	4031	7	N/A
	Real	41	8805	2	N/A
Louvain Modularity	Fake	122	373	7	40
	Real	127	303	2	37
Connected Components	Fake	24	14080	7	N/A
	Real	26	13853	2	N/A

Fig. 13. Comparative analysis of Community Detection

Community Algorithm	Type of Users	Top 3 Influencers (user id)	Community label in which Influencer node is present	Size of the community	Community Rank based on size (Nth largest community)
Label Propagation	Fake	19672966	21104	4031	1
		21237884	18748	2177	3
		20974554	21104	4031	1
	Real	57419364	1	8805	1
		1426071613	150	1610	2
		16255515	14889	225	8
Louvain Modularity	Fake	19672966	6509	350	4
		21237884	9440	373	1
		20974554	3865	338	6
	Real	57419364	8420	303	1
		1426071613	14889	225	2
		16255515	2592	218	13
Connected Components	Fake	19672966	0	14080	1
		21237884	0	14080	1
		20974554	0	14080	1
	Real	57419364	0	13853	1
		1426071613	14888	255	3
		16255515	0	13853	1

Fig. 14. Significance of Influencers in Community Detection

largest community might share similar interests or affinity to a certain user or tweet.

We can also observe that the louvain modularity forms the largest number of communities in both real and fake news. The number of overlapping communities formed is 33% of the total number of communities formed.

However, when we look at the results of Label Propagation, we could see that the size of the largest community formed is significantly lower for the fake news in comparison to real news. This could be because the real news comprises more densely populated user networks facilitating the propagation of labels across the social graph.

The Fig.14 shows the significance of the top 3 influencers identified by the influencer detection methodology on the community detection.

From the Fig.14 we observe that the top 3 influencers identified for both real and fake news always fall within the largest community. This signifies the importance of an influencer and community detection in identifying and understanding the propagation of information in social media.

Overlapping Community Label	Sub-community - 1 Label	Sub-community - 2 Label	Size of Overlapping Community
3865	3865	8557	338
3527	3527	61773	315
15234	15234	7618	289
10268	10268	11284	190
4692	4692	13139	184
15399	15399	5272	165
11010	11010	14275	153
5132	5132	11607	147
1732	1732	14096	145
11010	11010	11747	139

Fig. 15. Top 10 Overlapping Communities in Fake News

Overlapping Community Label	Sub-community - 1 Label	Sub-community - 2 Label	Size of Overlapping Community
9415	9415	10590	182
13326	13326	3143	173
4314	4314	7508	165
10420	10420	6221	160
11654	11654	8253	152
13724	13724	6540	151
11956	11956	11982	143
7067	7067	4167	139
663	663	1	137
11654	11654	14614	137

Fig. 16. Top 10 Overlapping Communities in Real News

The Fig.15 and Fig.16, show the Top 10 overlapping communities based on size determined by the Louvain Modularity algorithm for both real and fake news.

From the Fig.15 and Fig.16, we could observe that the hierarchical clustering is capable of grouping the smaller communities into larger communities based on the similarities identified by the user attributes. The Fig.17 and Fig.18 shows the visualization of the communities formed by various community detection algorithms for real and fake news respectively.

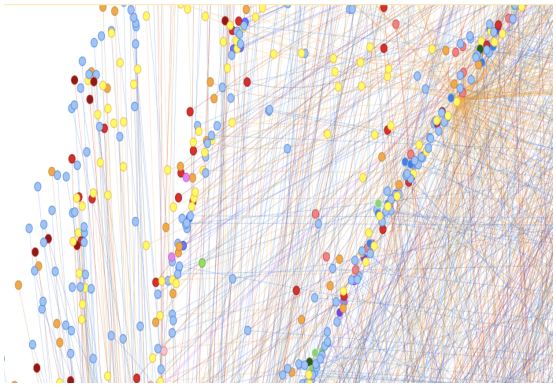

Fig. 17. Real User Community visualization

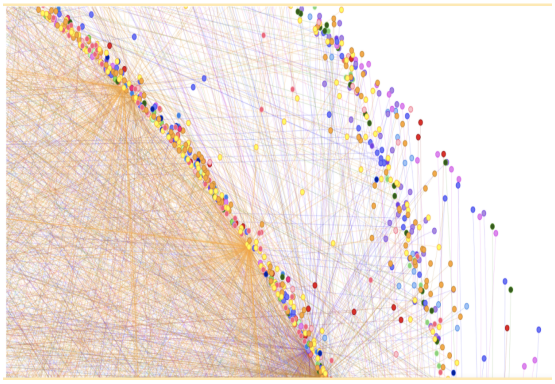

Fig. 18. Fake User Community visualization

2) Content Based Model:

To evaluate a binary classifier, a confusion matrix, F1 score and ROC_AUC are used.

a) *Confusion Matrix:* Confusion Matrix helps determine the effectiveness of a model. It consists of a 2*2 table with 4 different possible combinations based on the actual result and the predicted result. We can use the confusion matrix to calculate many important parameters like Precision, Recall, Accuracy and AUC-ROC Curve. The main components of a confusion matrix are:

Component	Predicted Value	Actual Value
True Positive [TP]	True	True
True Negative [TN]	False	False
False Positive [FP]	True	False
False Negative [FN]	False	True

Confusion Matrix

b) *F1 Score:* Models with Low Precision and High Recall are difficult to compare. Hence, we can use F1 Score as it measures Recall and Precision simultaneously. This allows us to use the Harmonic Mean of the Precision and the Recall

$$F1Score = 2*[Precision*Recall]/[Precision+Recall] \quad (7)$$

c) *ROC and AUC:* ROC (Receiver Operating Characteristic) is a probability curve that tells how effective the model is when distinguishing between various classes. Area under the curve(AUC) for ROC is used to represent the degree or measure of separability in a dataset. A model with higher AUC will be better at classifying the 1s as 1 and 0s as 0.

The confusion matrix [Fig. 19] of our content based model is shown below:

ACTUAL

PREDICTED

Confusion Matrix	Real	Fake
Real	TP 7033	FP 1068
Fake	FN 292	TN 7908

Fig. 19. Confusion Matrix

As can be seen from the above confusion matrix, the false positives are high. This might be a result of the extreme nature of our false news dataset(Russian Trolls) and thus our model caught those features very well. The main focus of our project is to catch the fake news. Even if some true news is classified as fake, this does not defeat our purpose. Our True positives and True negatives are high and fake negatives are low and thus our model works well for our purpose.
Correspondingly, the F1 score is ~0.91 and the ROC AUC is also ~0.91. Fig. 20 shows our model training and validation accuracy whereas Fig. 21 shows the learning curve for our model.

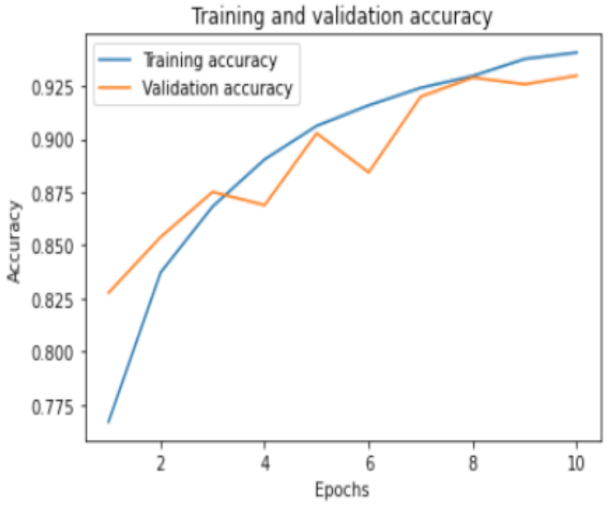

Fig. 20. Training and Validation Accuracy

Usually as we increase the number of epochs (or more training data), the training accuracy increases and it remains higher than the validation accuracy.For most part of the training this phenomenon takes place in our learning. As we can see in

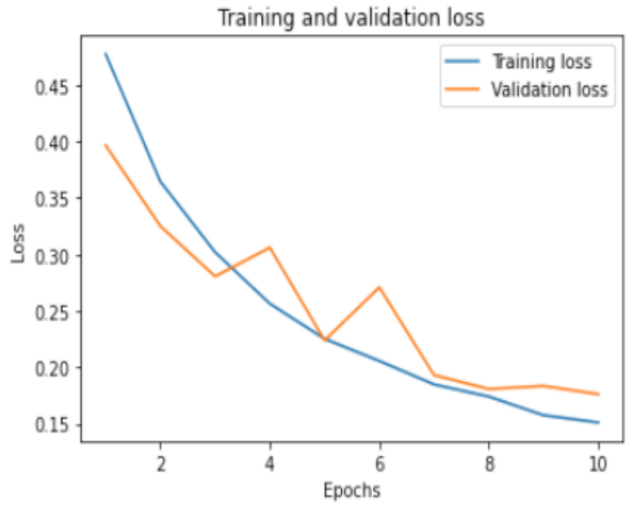

Fig. 21. Training and Validation Accuracy

the above figure, the validation loss(or training accuracy) is lower in the initial training. This is not normal but happens from time to time. It can happen due to the following reasons:

i) *Dropout:* Dropout layers are "on" during training, but they are turned "off" (skipped) during validation and testing. Therefore, dropping out has little effect on training errors.
ii) *Smaller datasets:* Such datasets have smaller intrinsic variance so the model properly captures patterns inside of the data and train error is greater because the inner variance of the training set is greater than validation set.

VII. Conclusion and Future Work

Thus, the methodology adopted has successfully been able to showcase the importance of influencer and community detection techniques in understanding the propagation of information in a social network. The influencer detection methodology applied to the twitter user profile uses the majority voting technique within the individual centrality measures to identify the influencers. However, a more powerful and accurate influencer detection engine could be built by computing the weighted centrality measure from the individual centrality measures. The community detection method applied to Twitter user profiles is based on current existing algorithms. However, a more powerful community detection model that uses an ensemble of all the algorithms can be constructed to analyze the spread of fake news, which helps to identify fake news more effectively. The detection technology can be made more real-time by creating a browser extension. The browser extension will extract all tweets and user profiles from the URLs of the currently viewed social media sites, and perform fake news propagation and detection analysis.

For the content-based model, the amount and quality of data played a big role. Initially we explored various data sources that gave us unbalanced dataset and that did not give us good accuracy. Finally we combined positive labelled data from CredBank and negative labelled data from the Russian trolls dataset to obtain a balanced dataset. Once we had the dataset in place, pre-processing was smooth. Further, GloVe embedding helped us improve the performance by around 15% as compared to the WordtoVec embedding. After tweaking the hyper parameters, we obtained an F1 score of 0.91.

The approach currently being implemented has two methodologies - Fake News Propagation analysis and Fake News Detection. While the fake news propagation analysis gives importance to the users and their interactions, the fake news detection system focuses more on the content. Suggestions for future work would be to combine both these user based and content based aspects of fake news to build a more efficient fake news detection engine.

REFERENCES

[1] A. Gelfert, "Fake News: A Definition,"*Informal Logic*, vol. 38, no. 1, pp. 84–117, 2018.

[2] K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu, "Fake News Detection on Social Media,"*ACM SIGKDD Explorations Newsletter*, vol. 19, no. 1, pp. 22–36, Sep. 2017. https://doi.org/10.1145/3137597.3137600

[3] C. Zhang, A. Gupta, C. Kauten, A. V. Deokar, and X. Qin, "Detecting fake news for reducing misinformation risks using analytics approaches," *European Journal of Operational Research*, vol. 279, no. 3, pp. 1036–1052, Dec. 2019. https://doi.org/10.1016/j.ejor.2019.06.022

[4] M. Visentin, G. Pizzi, and M. Pichierri, "Fake News, Real Problems for Brands: The Impact of Content Truthfulness and Source Credibility on consumers' Behavioral Intentions toward the Advertised Brands," *Journal of Interactive Marketing*, 45, pp.99-112, 2019. https://doi.org/10.1016/j.intmar.2018.09.001

[5] A. Guess, J. Nagler, and J. Tucker, "Less than you think: Prevalence and predictors of fake news dissemination on Facebook," *Science Advances*, vol. 5, no. 1, p. eaau4586, Jan. 2019. https://doi.org/10.1126/sciadv.aau4586

[6] Á. Figueira and L. Oliveira, "The current state of fake news: challenges and opportunities," *Procedia Computer Science*, vol. 121, pp. 817–825, 2017. https://doi.org/10.1016/j.procs.2017.11.106

[7] J. L. Nelson and H. Taneja, "The small, disloyal fake news audience: The role of audience availability in fake news consumption," New Media Society, vol. 20, no. 10, pp. 3720–3737, Feb. 2018.

[8] N. Ruchansky, S. Seo, and Y. Liu, "CSI," *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management - CIKM '17, 2017*. https://doi.org/10.1145/3132847.3132877

[9] S. M. Jang, T. Geng, J.-Y. Queenie Li, R. Xia, C.-T. Huang, H. Kim, and J. Tang, "A computational approach for examining the roots and spreading patterns of fake news: Evolution tree analysis," *Computers in Human Behavior*, vol. 84, pp. 103–113, Jul. 2018. https://doi.org/10.1016/j.chb.2018.02.032

[10] V. L. Rubin, Y. Chen, and N. J. Conroy, "Deception detection for news: Three types of fakes," *Proceedings of the Association for Information Science and Technology*, vol. 52, no. 1, pp. 1–4, 2015.

[11] A. Bondielli and F. Marcelloni, "A survey on fake news and rumour detection techniques," *Information Sciences*, vol. 497, pp. 38–55, Sep. 2019. https://doi.org/10.1016/j.ins.2019.05.035

[12] N. J. Conroy, V. L. Rubin, and Y. Chen, "Automatic deception detection: Methods for finding fake news," *Proceedings of the Association for Information Science and Technology*, vol.52, no. 1, pp. 1–4, 2015. https://doi.org/10.1002/pr a2.2015.145052010082

[13] W. Y. Wang, "'Liar, Liar Pants on Fire': A New Benchmark Dataset for Fake News Detection," *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics* (Volume 2: Short Papers), 2017. https://doi.org/10.18653/v1/p17-2067

[14] Y. Liu P. and Y.- fang Wu Brook, Liu, Yang P. and Yi-fang Brook Wu, "Early Detection of Fake News on Social Media Through Propagation Path Classification with Recurrent and Convolutional Networks.", Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence,AAAI (2018). AAAI, 2018.

[15] F. Monti, F. Frasca, D. Eynard, D. Mannion and M. M. Bronstein, "Fake News Detection on Social Media using Geometric Deep Learning",February 2019, available: https://arxiv.org/abs/1902.06673

[16] F. A. Ozbay and B. Alatas, "Fake news detection within online social media using supervised artificial intelligence algorithms," *Physica A: Statistical Mechanics and its Applications*, vol. 540, p. 123174, Feb. 2020. https://doi.org/10.1016/j.physa.2019.123174

[17] M. Andrews and G. Vigliocco, "The Hidden Markov Topic Model: A Probabilistic Model of Semantic Representation," *Topics in Cognitive Science*, vol. 2, no. 1, pp. 101–113, 2010.

[18] P. E. N. Lutu, "Using Twitter Mentions and a Graph Database to Analyse Social Network Centrality," 2019 6th International Conference on Soft Computing Machine Intelligence (ISCMI), 2019.

[19] A. M. Pudjajana, D. Manongga, A. Iriani, and H. D. Purnomo, "Identification of Influencers in Social Media using Social Network Analysis (SNA)," *2018 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI), 2018*.

[20] Y. Xing, F. Meng, Y. Zhou, M. Zhu, M. Shi, and G. Sun, "A Node Influence Based Label Propagation Algorithm for Community Detection in Networks," The Scientific World Journal, vol. 2014, pp. 1–13, 2014.

[21] Hal.inria.fr, 2020. [Online]. Available: https://hal.inria.fr/inria−00492058/document. [Accessed: 28- Nov- 2020].

[22] D. Singh, "Graph: Strongly Connected Components," Medium, 23-Aug-2020. [Online]. Available: https://medium.com/@deepika9410/graph−strongly−connected−components-779b463e8c8b. [Accessed: 28-Nov-2020].

[23] The Label Propagation algorithm - Chapter 6. Community detection algorithms. [Online]. Available: https://neo4j.com/docs/graph-algorithms/current/algorithms/label-propagation/. [Accessed: 28-Nov-2020

[24] The Louvain algorithm - Chapter 6. Community detection algorithms", Neo4j.com, 2020. [Online]. Available: https://neo4j.com/docs/graph-algorithms/current/algorithms/louvain/. [Accessed: 28- Nov- 2020]

[25] The Connected Components algorithm - 9.3. Community detection algorithms. [Online]. Available: https://neo4j.com/docs/graph-algorithms/current/labs-algorithms/connected-components/. [Accessed: 28-Nov-2020]

[26] T. Li, M. Hua and X. Wu, "A Hybrid CNN−LSTM Model for Forecasting Particulate Matter (PM2.5)," in IEEE Access, vol. 8, pp. 26933−26940, 2020, doi: 10.1109/ACCESS.2020.2971348.

[27] E. Loos and J. Nijenhuis, "Consuming Fake News: A Matter of Age? The Perception of Political Fake News Stories in Facebook Ads," Human Aspects of IT for the Aged Population. Technology and Society Lecture Notes in Computer Science, pp. 69–88, 2020.