# Identifying Misinformation Spreaders on Twitter using Ego-Centric Network Embeddings


By

Atta Ullah

**Department of Computer Science**
**Quaid-i-Azam University**
**Islamabad, Pakistan**
**November, 2023**

Dedicated to

***my father, whose unwavering support has been my cornerstone of strength, and to my mentor and guiding light, my mother, whose wisdom and encouragement shaped this work. I am also grateful to my parents for their endless love and encouragement throughout this journey.***

# Declaration

I hereby declare that this dissertation is the presentation of my original research work. Wherever contributions of others are involved, every effort is made to indicate this clearly with due reference to the literature and acknowledgment of collaborative research and discussions.

This work was done under the guidance of Dr. Rabeeh Ayaz Abbasi, Department of Computer Sciences, Quaid-i-Azam University, Islamabad.

Date: November 30, 2023

Atta Ullah

# Abstract

Social Media platforms such as Twitter remove distance barriers and play an important role in broadcasting information due to their ease of use, speed, and accessibility. As a result, a huge volume of information is generated every day. However, it causes the spread of misinformation, or false information which leads to catastrophic events and spreads uncertainty around in societies. Some organizations like Snoops and PolitiFact check the authenticity of social media posts related to politicians and celebrities. But misinformation is not limited to only politicians and celebrities and therefore we need an automatic approach to detect misinformation on time. To address this sensitive problem, Researchers used machine and deep learning models such as SVM, XGBoost, Random Forest, Decision Trees, Recurrent Neural Networks, etc., by extracting stylometry features such as sentence length, sentence segmentation, part-of-speech tagging, tokenization, and linguistic features such as sentimental features, word frequency, and bag-of-words. Most of these approaches only use the contents of the tweets for detecting misinformation. Since false information is engineered to influence a wide range of users, it is difficult to detect misinformation purely based on contents. Similarly, fine-tuning model on one dataset does not perform well on another dataset because of differences in domains such as fine-tuning a model on "COVID-19" related datasets cannot perform well on political statements related datasets. Therefore, more information such as social context or propagation feature is required to detect misinformation. In this thesis, we propose a model to distinguish

between misinformation spreaders and regular Twitter users by utilizing the propagation feature of tweets (i.e., how the information flows in the network?). The proposed model is based on Graph Neural Network (GNN) and consists of two parts. First, we generate an ego-centric graph up to 3 hops and then apply a state-of-the-art GNN model to detect misinformation spreaders. Experimental results show that the deep learning classifier "Deep Graph Convolutional Neural Network" (DGCNN) outperforms in term of Mathew's Correlational Coefficient (MCC). The DGCNN consists of three parts: In the first part the layers of Graph Convolutional Network are used to learn embeddings for each user by aggregating their neighborhood information, then a Sort-Polling layer is used to sort the vertex features, and then use a traditional convolutional layer and a dense layer to learn embeddings on graph level. Experimental results show that propagation features are valuable features to detect misinformation. We compare our results with other baseline models and the proposed model outperforms the baseline models in terms of Accuracy, ROC-AUC, and MCC.

# Acknowledgment

All praise to **ALLAH**, the Beneficent, the Merciful, and respect for his **Prophet Muhammad (peace be upon him)**, who made us recognize our creator. I am grateful for the most Beneficent and Merciful **ALLAH**, who has always helped me in all life matters. I have always been rewarded more than I tried. Praise to **ALLAH** almighty for giving me the courage and strength to complete this task. All esteem is for His **Prophet Muhammad (peace be upon him)**, a role model for humanity whose teachings have served as a beacon light for society in times of despair and darkness and provide us regular guidance in every sphere of life.

First and foremost, I am deeply indebted to my supervisor, **Dr. Rabeeh Ayaz Abbasi**, head of the Department of Computer Sciences, Quaid-i-Azam University, Islamabad, for his unwavering dedication, insightful feedback, and invaluable guidance throughout the research process. His expertise, patience, and encouragement played a pivotal role in shaping the direction of this thesis. **Dr. Rabeeh Ayaz Abbasi** is an inspiration to me. I never met a man like him. Furthermore, I thank **Dr. Akmal Saeed Khattak** for the research group meetings and guidance, encouraging me to complete this dissertation. I am grateful to all research group members, especially **Irfan ul Haq**, who guided me in the research process.

Without mentioning **Dr. Anwar Said**, this acknowledgment is incomplete. He

gave me advice in every matter of my life. I am very grateful for his guidance and for shaping my future. I appreciate his efforts in guiding me throughout the research process. Furthermore, I am thankful to my teachers and faculty members of computer science: **Dr. Onaiza Maqbool**, **Dr. Khalid Saleem**, **Dr. Ghazanfar Farooq Siddiqui**, **Dr. Shuaib Karim**, **Dr. Muazzam A Khan Khattak**, **Dr. Muddassar Azam Sindhu**, **Dr. Ayyaz Hussain**, **Dr. Umer Rasheed**, **Dr. Syed Muhammad Naqi**, **Ms. Memoona Afsheen Malik**, and **Ms. Ifrah Farrukh Khan** for their guidance and moral support.

I want to thank my parents for their unwavering support. Their belief in me fueled my determination to persevere. They have always pushed me forward in all possible manners. Their unconditional love and support made the academic journey possible for me. Let me mention my love and respect for my sisters; they are all I have. I remember my late brother, **Muhammad Hussnain**, whose absence makes me sad. His absence is deeply felt among us, and the void he left behind is a poignant reminder of his enduring impact.

Furthermore, I am grateful to my grandparents, uncles, and cousins for their unconditional love and support. Let me present respect by mentioning their names: **Inzar Gul**, **Zahir Gul**, **Naseer Ullah**, **Zakir Ullah**, and **Rafi Ullah**. They always stand with me in every matter of life. My life is empty without my maternal uncles **Abdullah** and **Ihsanullah**, who always treat me like friends. They always support me emotionally and financially. I am deeply indebted to them.

My gratitude extends to my colleagues and friends, especially **Imran ul Haq**, **Anwar Sadad** and **Samiullah**, who supported me emotionally and intellectually during this academic journey. Their camaraderie and discussions were instrumental in shaping

my ideas and maintaining my motivation.

Friends profoundly influence shaping one's personality, and I am truly fortunate to have companions like **Hassan Saeed**, **Abdullah**, **Kashif**, **Inam Bashar**, **M. Faisal**, and **M. Rashid Minhas**, who consistently fill my days with happiness. They are more than just friends; they are akin to brothers who have always stood by my side steadfastly. It's with a mixture of affection and sorrow that I also remember my late friend, **Umer Rahman**, whose absence is deeply felt among us. His memory remains alive in our hearts, and his absence is a void that is keenly sensed by all.

I am also profoundly grateful for the guidance and camaraderie of my senior friends, **Mehran Yousaf**, **Nouman Khan**, **Jameel Arif**, **M. Almas Khan**, **Zulqernain**, **Waqar Ahmad Khan**, **Islam Kamran** and many more who have not only enriched my journey with their wisdom but have also been a source of inspiration for me.

Last but not least, my affection for my classmates **Fouzia Qureshi**, **Ubaid Jadoon**, **Laraib Khan**, **Aqleem Abbas** and many other made the amazing journey of my studies even better. While I have mentioned specific names, I recognize that there are countless others who have played a role in shaping my academic and personal growth. Thank you all for being a part of this journey.

Thanks and Regards,

*Atta Ullah*

# Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

## 1.1 Background

We are in the information age, and much false news flows daily. Fake news is not new terminology but got attention during the 2016 US presidential campaign. During those days, hundreds of websites published false stories and biased stories. Similarly, the top 20 stories, based on fake news, generated 711,000 shares, comments, and reactions on Facebook. Fake news is becoming a threat to democracy, journalism, and freedom of speech. Due to the massive use of fake news in 2016, the Oxford Dictionary declared the "post-truth" as the word of the year. In 2017, the Collin dictionary declared fake news as the word of the year. If we look at the origin of fake news, it is from before the printing press media. So, the question is, why the fake news get attention globally? It's because it is now easier to spread false news or false information faster on social media (e.g., Facebook, Twitter) than traditional media such as newspapers and television. Due to the cheaper use of technology, People mainly depend on social media to get news. According to Pew Research [1], 72% of American people use social media, and 48% of them get their news from social media. They further claim that seven out of ten Twitter users get their news from different

[1] https://www.pewresearch.org/journalism/2021/11/15/

sites and post it on Twitter. The study results are shown in Figure 1.1. Social media is now a vulnerable platform that can potentially accelerate fake news dissemination. It virtually narrows the physical distance between people and allows rich operations like posting, commenting, and sharing information quickly.

People spread fake news for personal or social benefits. It causes multiple challenges which can influence the lives of people. However, fake news detection on social media platforms is quite a challenging problem due to the massive amount of information on social media. This enormous amount of information makes it difficult to differentiate between fake and genuine news.

## 1.2 Misinformation and Fake news

There is no single definition of fake news. Therefore, we will discuss different definitions used in the literature and examine the related or overlapping concepts of fake news. A simple definition of fake news is when false or misleading information is presented as news [Duffy et al., 2020]. If we look into the details, three characteristics can help determine if a given information is fake news. 1) Authenticity (contains any non-factual statement or not). 2) Intention (aiming to mislead), and 3) news (whether the information is news or not). We will classify different related or overlapping concepts of fake news on these three characteristics. In the existing literature, the term fake news connects to different terms and concepts such as misinformation, rumors, satire news, disinformation, false news, click-bait, cherry-picking, and deep fakes [Zhou and Zafarani, 2020]. Table 1.1 defines these terms regarding the three characteristics authenticity, intention, and news.

'Fake news' is often highly associated with politics, which blurs the issue's significance. Experts advise avoiding using fake news terms or at least limiting its use. 'False information' is more suitable as it covers a diverse range of misinformation

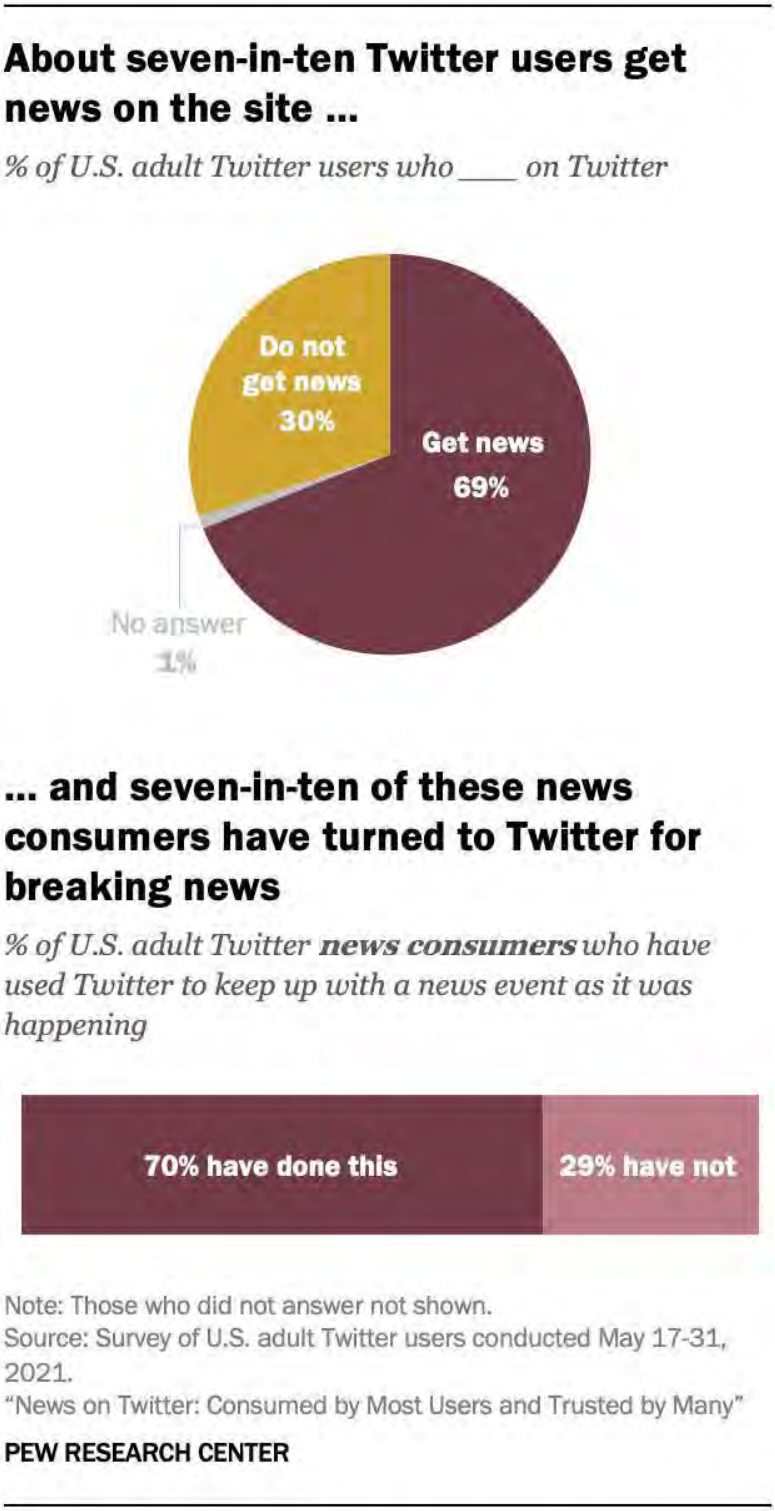



Figure 1.1: Pew Research Center report on the news got by Twitter users

Table 1.1: Comparison between fake news and their related concepts

| Related Concept | Authenticity | Intention | News |
|---|---|---|---|
| Misinformation | Non-factual | Undefined | Undefined |
| Disinformation | Non-factual | Mislead | Undefined |
| Propaganda | Non-factual | Mislead | Undefined |
| Deceptive news | Non-factual | Mislead | Yes |
| False news | Non-factual | Undefined | Yes |
| Satire news | Non-unified | Entertain | Yes |
| Rumor | Undefined | Undefined | Undefined |
| Clickbait | Undefined | Mislead | Undefined |
| Biased News | Undefined | Mislead | Undefined |
| Imposter Content | Undefined | Mislead | Undefined |
| Manipulated Content | Undefined | Mislead | Undefined |
| Deep fakes | Non-factual | mislead | undefined |
| Cherry-picking | Commonly factual | Mislead | Undefined |

and disinformation. It covers the topics of health, economics, and the environment across all platforms and genres. Based on the intention, the authors [Guo et al., 2020] divide the false information into two categories; misinformation and disinformation. Misinformation refers to inaccurate information which is created accidentally during an event. It is done without the purpose of misleading people, and it gets propagated intentionally or unintentionally.

On the other hand, disinformation is inaccurate or false information to mislead people. Fake news comes under the umbrella of disinformation, which is verifiable false news. The spread of both (misinformation and disinformation) has created significant challenges for society in the past. It has affected the ability to progress in crucial areas such as public health, climate change, and democracy. Throughout the discussion, we will use misinformation or false information as umbrella terms encompassing all false or inaccurate information disseminated via social media.

Now the question is, why are people influenced by misinformation and conspiracy theories? The possibility of people believing fake news depends on several factors. The first one is confirmation bias. People believe misinformation or conspiracy theories that align with their pre-existing beliefs, even if they are inaccurate. Due to

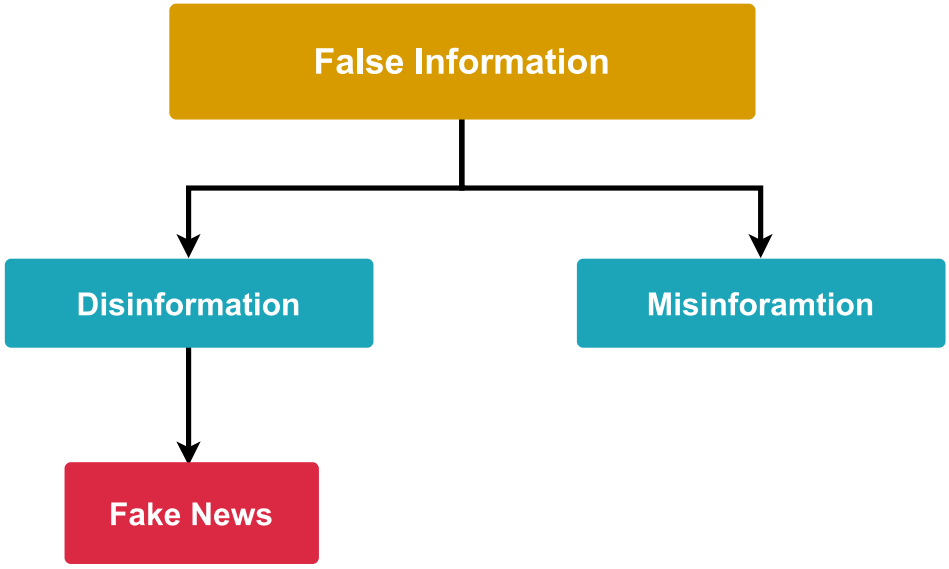



Figure 1.2: False news and their related concept

an individual's personal preferences or political motivation, they believe in false information. The second one is the lack of critical thinking and media literacy skills; People may accept information without adequately evaluating its source or accuracy. False information often contains emotional or sensational stories that trigger strong emotions in readers, further boosting the individual's belief in its authenticity. Furthermore, people usually associate credibility with a well-known source, leading them to accept information without verifying it.

Various actors can use false information for their benefit, including financial and political gain. Someone can use it to destroy the reputation of a company or person. Political organizations may spread false information to influence public opinion and undermine trust in the system. Scammers and fraudsters may use fake news to spread misinformation and scam people out of their money. Foreign governments may engage in information warfare campaigns, using fake news to manipulate public opinion. It

can be a profitable business. For example, click-bait websites profit from fake news by driving traffic to their sites. Social media companies may also benefit from fake news spread as it increases user engagement and generates more revenue from advertising. It is crucial to be cautious and critically evaluate the information we come across to avoid the possible catastrophe caused by false information.

## 1.3 Approaches to combat fake news

There are several ways to combat fake news. They are divided into two categories. 1) Fact-checking and 2) Automatic Detection.

### 1.3.1 Manual Fact checking

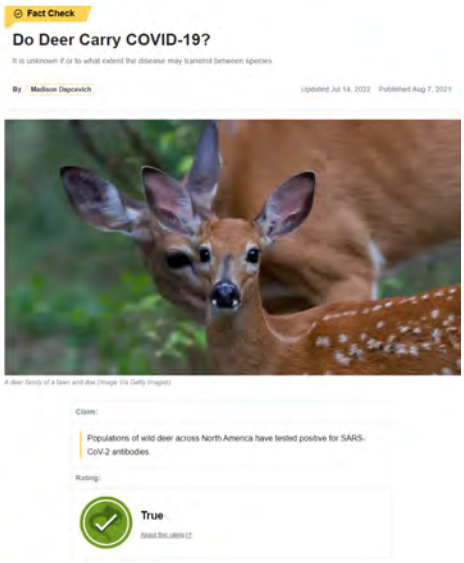
Fact-checking was initially developed in journalism to evaluate the authenticity of news by extracting knowledge from the news content and comparing it with known facts. Manual-fact checking can be divided into two subcategories; Expert-based and crowd-sourced fact-checking.

1. **Expert-based fact-checking** needs domain experts to check the authenticity of the news and it is time-consuming and costly. Some examples of the fact checking sites are PolitiFact[2], Factcheck.org[3] and snoops[4] etc. Figure 1.3 shows the example of fact-checking on PolitiFact and Snoops.

2. **Crowd-source fact checking** needs a large group of people to check the authenticity of the news, typically an online source. It involves assigning fact-checking to various people, allowing for the use of a broader range of perspectives and experiences in the verification process.

[2] `https://www.politifact.com/`
[3] `https://www.factcheck.org/`
[4] `https://www.snopes.com/`


(a) Fact checking on politifact




(b) Fact checking on snoops

Figure 1.3: A manual fact-checking on Politifact and Snoops

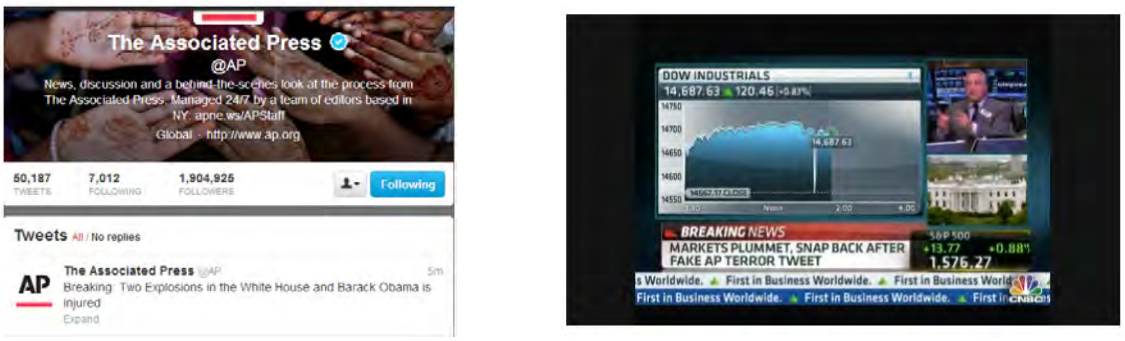
### 1.3.2 Automatic Detection

The second one is automatic or algorithmic fake news detection. It uses algorithms (mostly machine learning algorithms) to detect false information or fake news. It is further divided into 1) Content-based methods, 2) context-based methods, 3) propagation-based methods, and 4) a hybrid combination. In content-based methods, we classify the news as true or false based on the news content. In context-based methods, we consider time and location features, and in the propagation, we classify the user or news based on how their post propagates in the user network. In the last one, we combine all these methods to verify whether the information is accurate or false.

## 1.4 Motivation

Due to the large amount of information available on social media, it is challenging to distinguish between genuine and false information. Manual fact-checking is time-

consuming and costly, and we cannot manually check whether each piece of information on social media is true or false. False news spreads on social media very fast and can cause social harm. In 2021 a survey was conducted on financial decision-making. Three out of five American people say that false news is a threat to their financial decision-making. A news channel posted about an explosion in the white house and said that the 44th president of America, Barack Obama, was injured. After seven minutes, they posted that the account had been hacked, and the news was false. Within these seven minutes $136.5 billion were wiped out from stock value [Zhou and Zafarani, 2020]. According to the study conducted by the Global Disinformation Index[5], the sites that spread fake news in European countries earn more than $76 million annually.




(a) A fake news on Twitter

(b) Their impact on stock

Figure 1.4: A fake news and its impact on stock

The importance of combating fake news increases with the current ongoing COVID-19 pandemic. According to WHO[6], in the days between 9 January 2023 to 5 February 2023 (28 days), approximately 10.5 million new cases arrived, out of which 90,000 deaths reported globally. Approximately 754 million people were affected, of which 6.7 million died worldwide. It is a huge pandemic, and many false information and conspiracy theories have been made about it. A study by the American Journal of Tropical Medicine and Hygiene discovered that 5800 patients were hospitalized due

[5] https://www.disinformationindex.org/
[6] https://www.who.int/publications/m/item/3

to false information on social media. Some people died due to drinking methanol or taking drugs because they got misinformation that these products can help in recovery from COVID-19 disease. Some fact-checking groups extracted popular posts and found that such posts claimed that eating garlic, consuming bananas, and many more products can prevent the disease [Kim and Tandoc Jr, 2022]. Authors [Szebeni et al., 2021] discuss several events during the COVID-19 pandemic across different countries, showing the impact of fake news and misinformation. For instance, in response to the conspiracy theories claiming that 5G cellular networks were the cause of the disease, [Biradar et al., 2023] over 200 incidents of attacks against telecom workers were reported in the UK.

Furthermore, many mobile telecom masts were set on fire in the Netherlands due to such misinformation. These events highlight the dangerous consequences of false information spread and the importance of combating fake news. To avoid these consequences caused by false information, we need to identify the fake news on time. Due to the massive volume of information, manual fact-checking is not scalable. With the advent of automated technologies, academia, and industry have grown interested in designing automatic fake news detection solutions.

With the fast development of machine learning algorithms and deep learning technologies, they can be a significant alternative to manual fact-checking, also achieving the requirement to detect the news quickly. The researchers got attention to use machine learning techniques to identify fake news or users who spread it. There is plenty of existing work. Existing methods for detecting fake news or false information can be classified into three broad categories; content-based, social context-based, and propagation-based. In content-based methods, fake news or false information is classified based on content. Most works on fake news detection use content-based approaches that depend on linguistic (lexical and syntactical) features capturing deceptive cues or writing styles. Researchers used content extracted from a user's posts

to identify misinformation spreaders directly [Wu et al., 2019], and a text classifier can be used to categorize false information or misinformation. In the paper [Jain and Kasbe, 2018], they prepared their dataset, manually labeled the data, and applied the Naive Bayes classification model to classify whether the post on Facebook is fake or real. Similarly, authors [Helmstetter and Paulheim, 2018] automatically collected large amounts of data containing hundreds of tweets and labeled these data as trustworthy and untrustworthy sources. Then they applied different machine learning models like naive Bayes, SVM, Neural Network, Random Forest, and XGBoost.

Furthermore, the authors in the article [Mehta et al., 2021] used a model based on a natural language processing framework, BERT (Bidirectional Encoder Representations from Transformers). The author fine-tuned the model on two datasets, LAIR [Vo and Lee, 2018] and LAIR Plus [Alhindi et al., 2018]. In the paper [Reis et al., 2019], authors used machine learning models including k-Nearest Neighbors (KNN), Naive Bayes (NB), Random Forests (RF), Support Vector Machine with RBF kernel (SVM), and XGBoost (XGB). The authors used a benchmark dataset Buzzfeed [Santia and Williams, 2018]. The authors demonstrate that the XGB classifiers give the best results. The underlying problem with content-based methods is the changing nature of the content's style, patterns, subjects, and platforms. Models trained on one dataset might not perform well on another due to differences in contents, style, or language. Context-based solutions for detecting misinformation spreaders have been developed to handle such challenges [Ullah et al., 2023].

To address the challenges in content-based detection, a group of researchers consider the context-based solution. The context-based methods detect the misinformation spreaders based on their profile information like followers, favorites, descriptions, etc. The authors of paper [Liu and Wu, 2020] present a neural network classifier for determining the integrity of news using social media tweets, retweets sequences, and Twitter user profiles. Context-based techniques have been divided into two methods:

stance-based and propagation-based. In the propagation-based methods, we detect the user based on how the information propagates in the network. Recently, studies observed that the propagation networks of fake news and real news are different. Based on these assumptions, researchers are developing a model to detect misinformation spread based on propagation patterns. Author [Monti et al., 2019] developed a fake news detection model based on a geometric deep learning approach. The authors generalize the convolution neural network model for graphs. The model was trained and tested by a verified fact-checking organization that works on social media content.

Furthermore, the author [Raza and Ding, 2022] trains an encode decoder model on content and propagation. The authors claim that experimental results show that the model gives high accuracy. Furthermore, the authors [Tuan and Minh, 2020] train a GNN model(Graph convolutional neural network) on a Mediaeval fake news data 2020. On the same dataset of medieval, the authors [Schaal and Phillips, 2020] train GNN models, graph convolutional neural networks, and Graph isomorphic Networks. The results of these models are not good. In 2020 the author of the paper [Saikia et al., 2022] presented a hybrid model combining content-based and context-based features to identify misinformation spreaders. They used a transformer model (bidirectional representation) to learn text features for the content-based part. They used a graph convolutional neural network to learn to embed and concatenate these two embeddings for context-based. The authors called it feature fusion and passed the resultant embeddings to a neural network to classify if the content is fake or real. Furthermore, the authors [Song et al., 2021] claim that they present a novel frame temporal propagation-based fake news detection framework. The model combines time information, content semantics, and structure. They developed a graph-based methodology on the temporal aspect of the news and then trained a temporal graph attention neural network (TGAT). Similarly, the author [Silva et al., 2021] presents

Propagation2Vec, a cutting-edge propagation-based fake news detection algorithm. The authors developed a hierarchical attention mechanism to encode the propagation pattern. The author combines a complete and partial graph network, combines the embeddings generated by Propagation2Vec of these two, and gives it to a classifier. Similarly, the author [Verma and Agrawal, 2022] presents a model PropFND (Propagation based Fake News Detection), which combines user profile feature and propagation feature to detect misinformation spreaders. They claim that the SVM classification model performs well while classifying the user.

In the literature, we found that the currently available datasets are news-related and politician or celebrity statements. There is no such database related to conspiracy theories or misinformation. The author introduced a dataset in the MediaEval 2022 workshop related to COVID-19 conspiracy theories and misinformation. The dataset contains both the propagation-structure as well as the content-based task. We selected the graph-based study in the workshop and proposed a solution. Our motivation behind choosing the graph-based task was the challenges that occur in content-based and the fact that misinformation and accurate information propagate differently on the network.

## 1.5 Overview of Twitter

Twitter is a free social networking service that allows users to publish short messages known as tweets. Tweets can contain text, videos, images, or links. It is a microblogging site that allows registered users to use brief messages to post, share, like, and respond to tweets. Non-registered users can only view tweets. Tweet has a limit of 280 characters. In 280 characters or less, Twitter users communicate their thoughts, news, real-time information, and jokes. The Twitter platform was launched in 2006. According to Statista, as of December 2022, Twitter had over 368 million monthly

active users globally. Figure 1.5 shows a decrease of 335 million users.

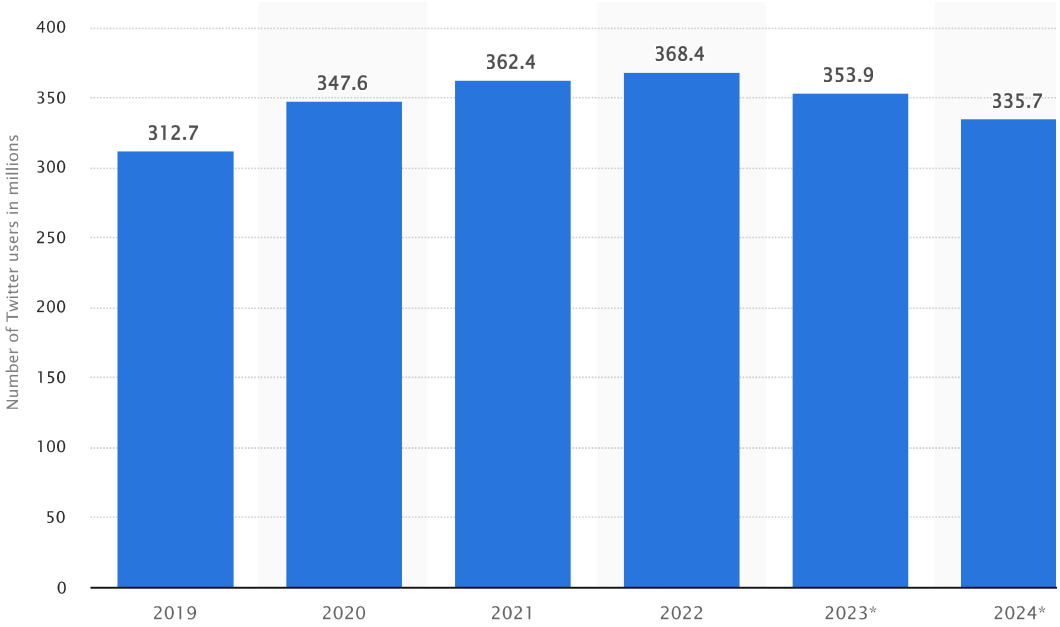



Figure 1.5: Statista 2022 report of Twitter users

### 1.5.1 Following users

On Twitter, every user can follow every other user. The "following" service allows users to view other user's tweets in their news feeds.

### 1.5.2 Post a tweet

When users tweet, their messages are uploaded to their profiles and appear in their follower's feeds. Members can use hashtags in their messages to relate tweets to a conversation thread or a more general subject. The hashtag is written as #keyword and serves as a meta-identification. It allows the relevant keyword to be used to find the message.

### 1.5.3 Interacting on Twitter

Users can retweet another user's tweet, which forwards it to their follower's timelines. They can also leave comments or respond to tweets.

Figure 1.6 shows a Twitter post. We divide the overall image into small red sub-blocks for understanding. The first block surrounds the user's tweet or text. The second block shows the date and time when the tweet was posted. It also shows how many users have seen the tweet. The third block shows the number of retweets and the fourth block shows the likes of the tweet. In the fifth block, a user can reply to a tweet and the sixth block contains the replies to the tweets.


Figure 1.6: User interaction on Twitter

## 1.6 Problem Definition

Detecting fake news has been a research topic for quite some time, with traditional approaches focusing on content-based and social context-based methods. However, recent studies have highlighted differences in the propagation patterns of fake and real news on social media platforms [Zhao et al., 2020]. Thus, it gives us insight that there is a difference between the propagation pattern of misinformation spreaders and regular social media users. Based on these assumptions, researchers paid attention to using propagation-based methods to differentiate between real users and misinformation spreaders.

Based on this assumption, we generate a propagation graph of social media users who spread the news. Let a set of user $U = \{u_1, u_2, u_3, \cdots, u_n\}$, tweets $T = \{t_1, t_2, t_3, \cdots, t_m\}$ and a set of followers $F = \{f_1, f_2, f_3, \cdots, f_n\}$ where $f_i \subset U$ is a set of users following the user $u_i \in U$ and $f_1 \cap f_2 \cap f_3 \cdots \cap f_n \neq \varnothing$. Let a user $u_i$ post a tweet $t_j$. Then three cases occur. First, the set of users $f_i$ who follow that user $u_i$ read the tweet $t_j$ may retweet them, then the other sets of users who follow those users read the retweeted tweet and may retweet it again. This way, a link is generated between these users. In the Second action, a user may read the tweet and give a reply on the tweet $t_j$, and other users may give replies to this reply. This way, a link is generated, and the third is that a user may read the tweet and quote it, and the other user may read the quoted tweet and give a reply or retweet it. This way, a link is generated. The combination of these links generates the **propagation graph**. It is possible that a user $u_x$ who is not following another user $u_i$ (i.e., $u_x \notin f_i$) retweet, quote, or reply to $u_i$, in that case there will be a direct link from $u_x$ towards $u_i$ in the propagation graph. Direct link is created due to the limitation of Twitter API, as it cannot be established through the API whether $u_x$ directly retweets $u_i$ or does it through some intermediate users (nodes).

As shown in the figure 1.7, a user $u_1$ makes a tweet a $t_1$ and his follower $f_i$ some of

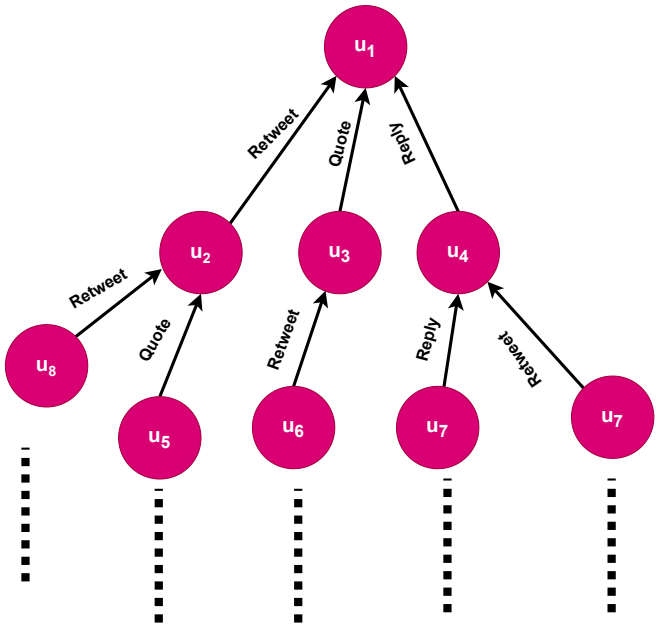



Figure 1.7: The propagation pattern of news on Twitter

them $\{u_2, u_3, \cdots, u_z\}$ read the tweet and retweet them, quote them or give reply on the tweet and then the follower of $u_2$, $u_3$, $\cdots$, $u_z$ denoted by $f_k$, $f_l$, $\cdots$, $f_m$ read the tweet and retweet, quote or reply and then the process continues, and a propagation pattern is generated.

### 1.6.1 Problem Statement

Identify the user on Twitter based on propagation patterns, whether a user is a misinformation spreader or a regular user.

### 1.6.2 Problem Formulation

We construct the undirected ego-centric graph $G_i = (V_i, E_i)$ based on the propagation pattern, where $V$ denotes the set of nodes or users where $|V_i| = m$ and $E_i$ represent the edges between nodes. Each user has particular profile features and is represented by $X_i \in R^{m \times d}$ is a set of vectors with d dimensions. There is an edge between nodes (user) when one node retweets the tweet, quotes the tweet or gives a reply to the tweet

of the other user/node. The sum of the count of the number of retweets, routes, and replies calculates the weight of the edge. The number of retweets means how many users retweet the tweet from that user, the number of routes means how many paths that two connected nodes occur, and the number of replies means how many users respond to the tweet.

$$a_{ij} = \Sigma_1^q Retweet_i + \Sigma_1^w Route_j + \Sigma_1^r Reply_k \tag{1.1}$$

So the adjacency matrix of the graph will be represented by $A_i = [a_{ij}] \in R^{n \times n}$. We have one graph for each user and for their propagation, so the overall dataset is represented by $D = \{(G_1, y_1), (G_2, y_2), ..., (G_i, y_i)\}$. Where $G_1, G_2$ is the graph constructed by the above procedure and the $y_i \in \{0 :$ regular user, $1 :$ misinformation spreader$\}$.

## 1.7 Graphs Representation Learning

### 1.7.1 Graph

A graph is a collection of nodes and edges. The node represents an entity like people in social media, items in a shop, etc., and edges represent connections between these nodes based on some relationship like friends in social media, etc. The node is also known as vertices or points; the edges are links, lines, or arcs. Figure 1.8 shows an example of a graph with its adjacency matrix and features matrix. Value in the adjacency matrix is 0 or 1 (or can be a numeric value based on the weight given). 1 means an edge is present between these nodes, and 0 means the edge does not exist.

Graphs represent many real-world applications, such as computer networks, telephone networks, circuit diagrams, and routes within a city or connecting cities. Social networking services like LinkedIn, Twitter, and Facebook can be modeled as graphs.

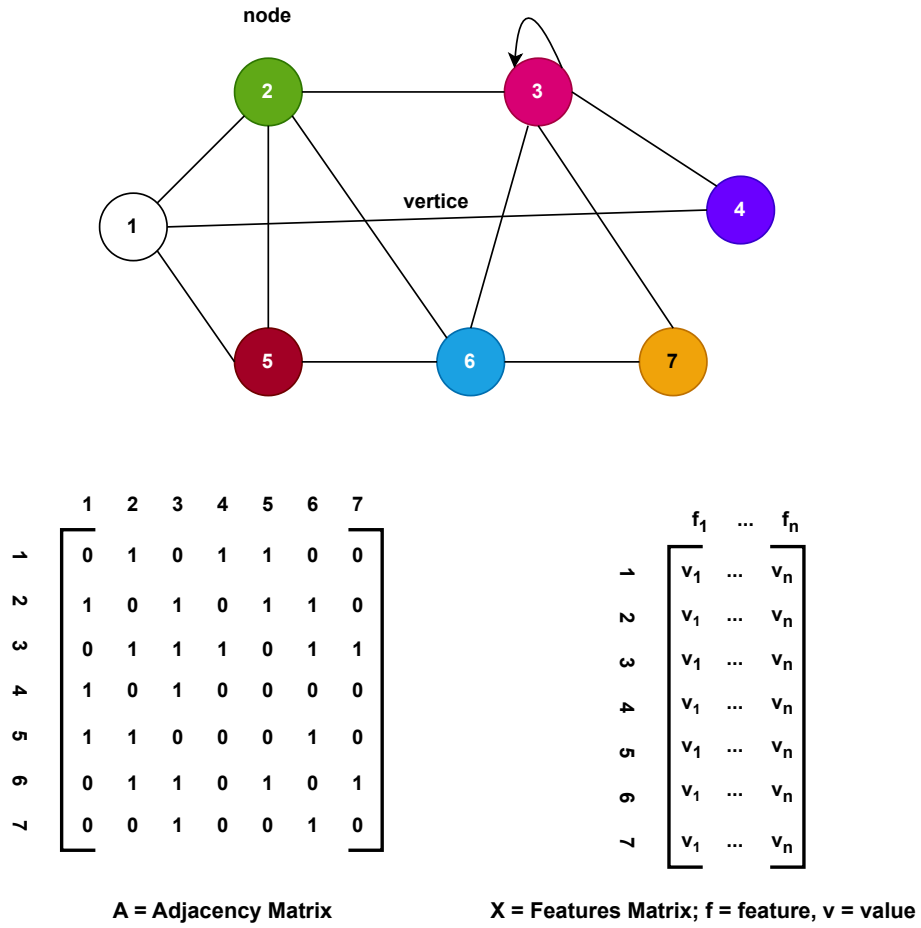



Figure 1.8: Graph representation

Each user on Twitter can be represented as a node, and their follow, follower relationship can be represented as edges. Each node can have attributes like id, name, gender, location, Etc.

The Graph is a powerful tool for modeling and analyzing social networks. Social media is a platform where groups are related to one another through interactions such as friendship, collaboration, or economic relationships. By modeling social media as a graph throughout these relationships, researchers can gain insights into the structure and behavior of social networks. Graphs can be used for a variety of purposes in social networking, including:

1. **Identifying key people:** Graphs can be used to identify important people

in a social network. It can be done by analyzing users' interaction on social media, and researchers can find the most influential people and how they spread information and influence other people or which groups are the target of those people.

2. **Community detection:** Graphs can detect communities in a social network. Researchers can group people into clusters or communities based on their interests, behaviors, and other features. They can find communities involved in specific events that support the violence or those against the violence.

3. **Recommendation systems:** The recommender system can be modeled as a graph. A social network uses a recommender system to recommend friends, other relevant posts, activities, organizations, jobs, etc. It also can be used in E-commerce shops to recommend items, shops, etc., or it may be used in terrorism to recommend places, shops, etc. [Khan et al., 2023].

4. **Network analysis:** Graph can be used to model social media data, giving us insights into a social network's overall structure. It enables us to analyze sentiments, detect communities, find influential people, and be used for event detection. Graph-based models capture the interactions and information flow between users in social media networks by analyzing the degree of connectedness between individuals.

### 1.7.2 Graph Representation Learning

Graph representation learning, also known as graph embedding or graph embedding learning, focuses on capturing a graph's structural and semantic features and representing them in a continuous, low-dimensional vector space known as embedding. This process aims to transform graph data into a vector representation that machine learning algorithms can efficiently process.

Graph representation learning techniques address the unique challenges posed by graph-structured data, where nodes represent entities like users, documents, proteins, etc., and edges represent relationships or interactions between these entities. Traditional machine learning algorithms usually struggle with structure data because they consider the input instances to be independent and identically distributed, which is not the case for structure data. Graph representation learning methods can be broadly classified into two categories:

**Feature-based methods:** These use graph-specific features and properties to derive node or graph-level representations. Features can include node attributes, graph topology, or statistical measures computed on the graph. Feature-based methods typically involve engineering handcrafted features and applying traditional machine learning algorithms, such as clustering or classification, on these features.

**Embedding-based methods:** These methods aim to learn low-dimensional representations, often called embeddings or latent vectors, that encode a graph's structural and semantic information. Embedding-based methods learn to map nodes or graphs to continuous vector spaces, where the geometric relationships between vectors reflect the underlying graph structure. Based on these embeddings, we perform multiple tasks like node classification, Graph classification, or regression.

## 1.8 Our Contributions

- Ego-centric Graph Extraction
- Model outperform sate-of-art methods.
- Neighbourhood sampling
- Training over three methods (GCN, GraphSAGE, and DGCNN)
- Proposed end-to-end framework (ECMSD Framework)

## 1.9 ECMSD Framework

The misinformation spreader detection framework is named ECMSD (Ego-Centric Misinformation Spreader Detection). The proposed framework consists of the following components/steps. First, tweets were extracted from Twitter based on some targeted keywords of COVID-19, and their retweets, replies (only direct ones), quotes, and quoted replies were also extracted. The profile feature of the users (number of followers, numbers of following, description length, account verified, etc.) was considered. A graph is constructed to analyze follower networks to identify the most likely retweet path a tweet may have taken and their replies, quotes, and quoted replies. Then we apply a pre-processing on the feature of the users. In the second component, we use a graph representation learning approach to learn the embeddings for each of the users on the node level by taking aggregation of the neighborhood nodes and on a global level (Graph level). Then the embeddings are given as input to the multilayer perceptron layers, and then a classifier is used to classify them into the regular user or misinformation spreaders.

## 1.10 Summary

This chapter introduces fake news, the origin of the term 'fake news', and how they spread on social media. We also discuss the motivation behind this study and possible challenges. The chapter also provides a hypothesis based on the motivation. Furthermore, the proposed framework is briefly discussed and a brief overview of the challenges involved in this approach is also discussed.

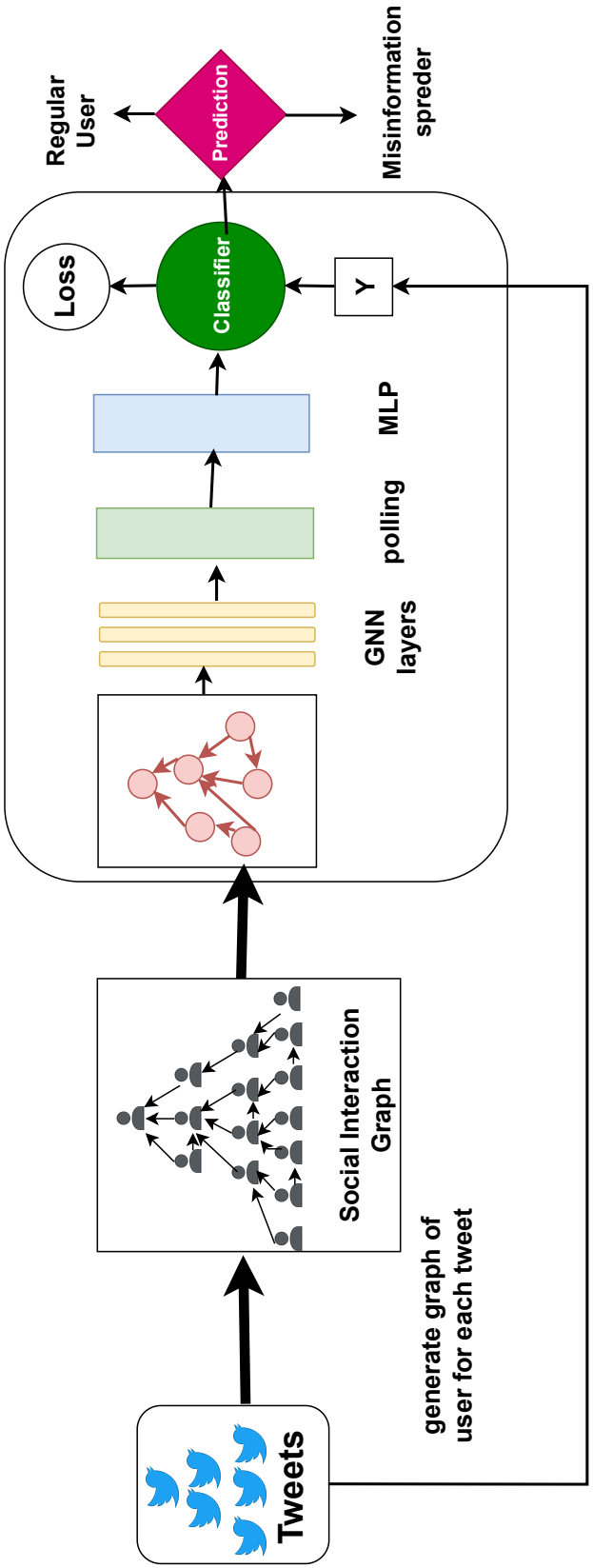



Figure 1.9: The proposed framework ECMSD

# Chapter 2

# Related Work

In recent years, machine learning and deep learning have seen rapid advancements, resulting in the emergence of various automatic detection methods proposed in the literature. In this chapter, we have discussed the existing studies used to detect misinformation or false information detection and the dataset used for this task. Section 2.3 discussed the existing studies on the area, and section 2.2 examined the benchmark dataset used in this area.

## 2.1 Misinformation Detection Techniques

The increasing global adoption and use of social media platforms have produced a climate conducive to the effective dissemination of online false news. There is a massive amount of information shared on social media daily. The information generated on social media is in greater variety, huge volume, and with increasing velocity following the definition of big data. The data consists of both genuine and fake information. Thus, it provides a platform for the easy share of false information. The government and other organizations are trying to identify this false information to avoid the consequences of false news. With the advent of machine learning and deep learning models, features are extracted to identify misinformation. This section

will discuss different deep-feature extracted to identify false information. Figure 2.1 shows the types of deep features. We will explain these features one by one.

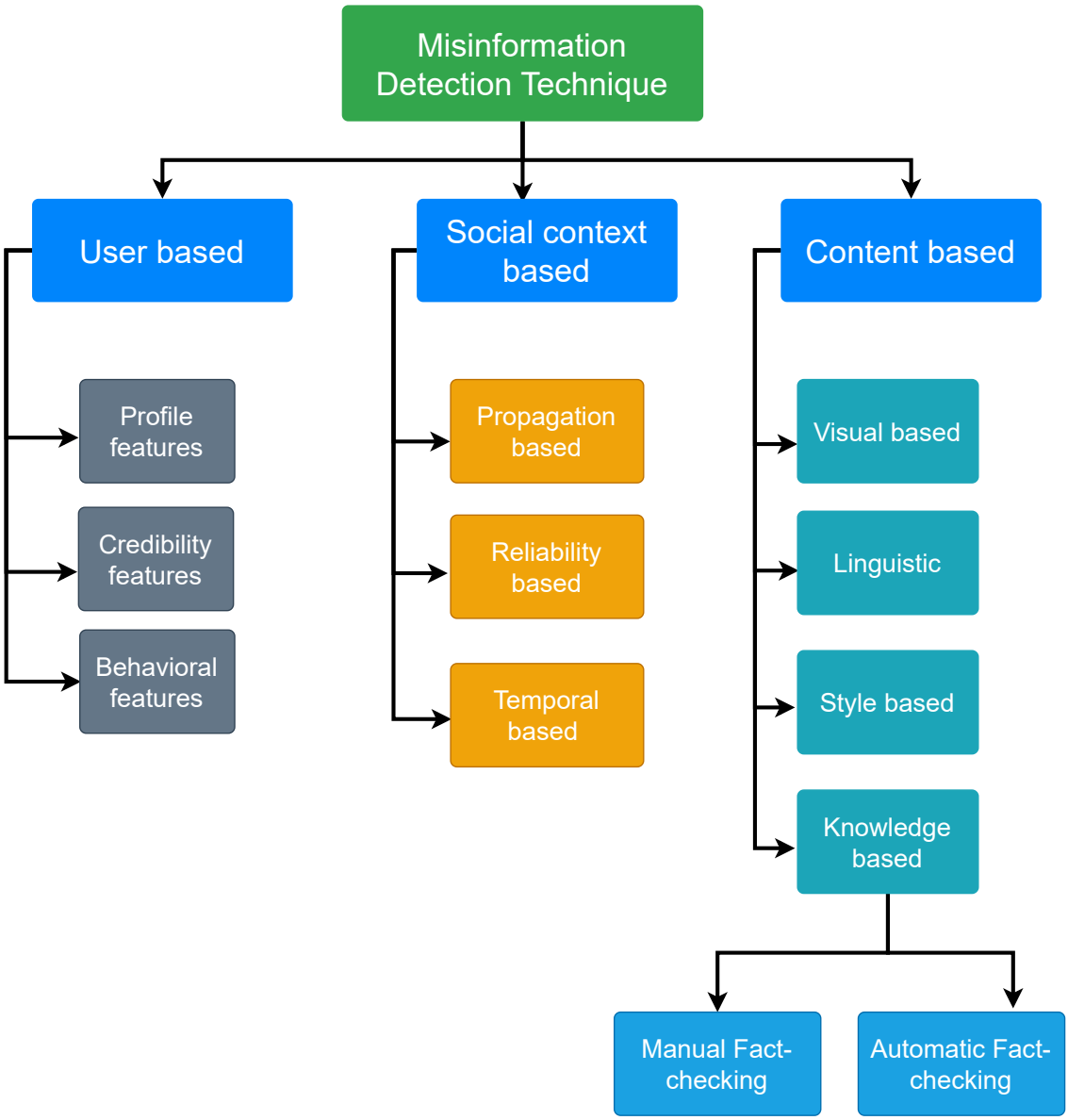



Figure 2.1: Misinformation detection techniques

### 2.1.1 User-Based Detection

User-based features are used to detect misinformation spreaders or regular users. The purpose of the user-based feature is to capture the user's unique characteristics. These

features are categorized into three categories: profile features, credibility features, and behavioral features.

1. **Profile features:** User-profile features include the user's name, geography, location, description, account creation date, verified or not, etc. Researchers are trying to distinguish between regular users and misinformation spreaders based on these features.

2. **Credibility features** Credibility features include the number of followers, number of followings, number of statuses, number of friends, length of description, verified or not, etc.

3. **Behavioral features:** According to [Sundararaj and Rejeesh, 2021], social media user behavior (SMUB) refers to numerous user actions and social relationships that represent the tendency of users to utilize social media services based on a complete evaluation of affective requirements, social impact, and other aspects. User behavior features seek to identify patterns in user behavior for both misinformation spreaders and regular users. The user anomaly score is a standard user behavior attribute computed by the number of interactions in a time window divided by the user's monthly average in [Zhao et al., 2014] for online misinformation spreaders detection.

### 2.1.2 Content-Based Detection

Fake news is identified by analyzing the news content in content-based detection. The content features include text, images, and videos. In literature, most of the work is done on text-based detection. The content feature is categorized into linguistic, visual, style-based, and knowledge-based features.

1. **Linguistic and syntactic-based:** Linguistic and syntactic-based features are natural language's primary components, structure, and semantics. Despite in-

tentionally generating misleading content to spread fake news, linguistic and syntax-based features remain valuable for analyzing misinformation. These features can be categorized into three levels: word-level, sentence-level, and content-level.

2. **knowledge-based:** Knowledge-based methods use fact-checking methods to check the authenticity of a given claim in a social media post. The claim's authenticity in the post is checked against external knowledge or fact. The fact-checking methods are categorized into two: manual fact-checking and Automatic fact-checking.

   (a) **Manual fact-checking:** Manual fact-checking is divided into two types: expert-based and crowd-sourced fact-checking. In the expert-based domain, experts are required to check the authenticity of the news, and in crowd-sourced based, many people act as fact-checkers and vote for the authenticity of the news. I discuss these approaches in detail in section 1.3.1.

   (b) **Automatic fact-checking:** Due to the massive amount of information generated, manual fact-checking is not scalable. Therefore, to address the scalability issue, an automatic fact-checking method is developed that depends on natural language processing (NLP) and machine learning (ML) techniques. In the upcoming section 2.3, I discuss these methods in detail.

3. **Style-based:** A style-based technique, similar to the one used for knowledge-based false news identification, can be used to identify false news. This method [Orabi et al., 2020] evaluates the writer's intent to deceive the audience rather than the credibility of the news item itself. Fake news providers are typically driven by a desire to influence vast groups of people by broadcasting false and misleading information. To make the names more memorable, they are usu-

ally entirely capitalized, and there are far more proper nouns and fewer stop words [Álvaro Figueira and Oliveira, 2017]. Style-based techniques capture the characteristics of writing styles that distinguish genuine users from misinformation spreaders to detect false news. [Hoy and Koulouri, 2021] investigate the writing style of hyperpartisan news to examine false information. Detecting stylistic deception in textual texts [Hoy and Koulouri, 2021] is the most significant contribution.

4. **Visual-based:** The false information is identified based on visual content such as images and videos. Recent research ( [Singhal et al., 2019], [Singh et al., 2020], [Yang et al., 2023], [Raj and Meel, 2021]) investigates visual-based aspects for identifying false information. Because visual content can increase the credibility of the news, false information producers use graphics content to mislead users [Guimarães et al., 2021].

### 2.1.3 Context-Based Detection

The phrase "social context" refers to the overall social environment and activity system in which news is distributed. This includes how social information is shared and how people interact. Furthermore, social networking sites increasingly dominate communication and knowledge transmission [Kondamudi et al., 2023]. Again, the social context of an internet community may provide helpful information for differentiating real news from false news. Examples of these social sites include Facebook, Twitter, Instagram, and other social networking sites. People's knowledge and updates are altering due to social media platforms. Social media platforms allow active online individuals to read about current issues, share personal experiences, and advocate for specific topics and themes.

In context-based methods, the misinformation spreader is detected based on how information propagate in the social network. The essential social context features are

user profiles, posts, replies, and network architecture. Recent studies show that the spreading patterns of real and false news on social networks are different [Hoy and Koulouri, 2021].

## 2.2 Automatic Detection Methods

Automatic detection utilizes machine learning algorithms to give false information. In the literature, we found that researchers use a lot of machine-learning models. We divide these algorithms into three categories: 1) Traditional Machine Learning, 2) Deep Learning, and 3) Geometric deep learning models. This section will discuss the machine and deep learning models used to identify misinformation or false information. Figure 2.1 shows the most used models in the literature.

### 2.2.1 Machine Learning

1. **Support vector machine:** Support Vector Machine (SVM) is a supervised learning method that chooses a better solution w.r.t a new parameter 'margin' from available options. The algorithm is mostly used for classification tasks and typically works better with binary classification. There can be multiple classification boundaries; SVM chooses a decision boundary with a maximum margin between data points of data classes. It results in better predictions in the case of certain data distributions.

2. **Logistic Regression:** Logistic regression is a statistical supervised learning method and an extension to a linear regression learning algorithm for categorical data classification. It typically fits a polynomial on data and applies log-odds to perform classification.

3. **Decision Tree:** Decision Tree is a relatively most explicit form of rules learning

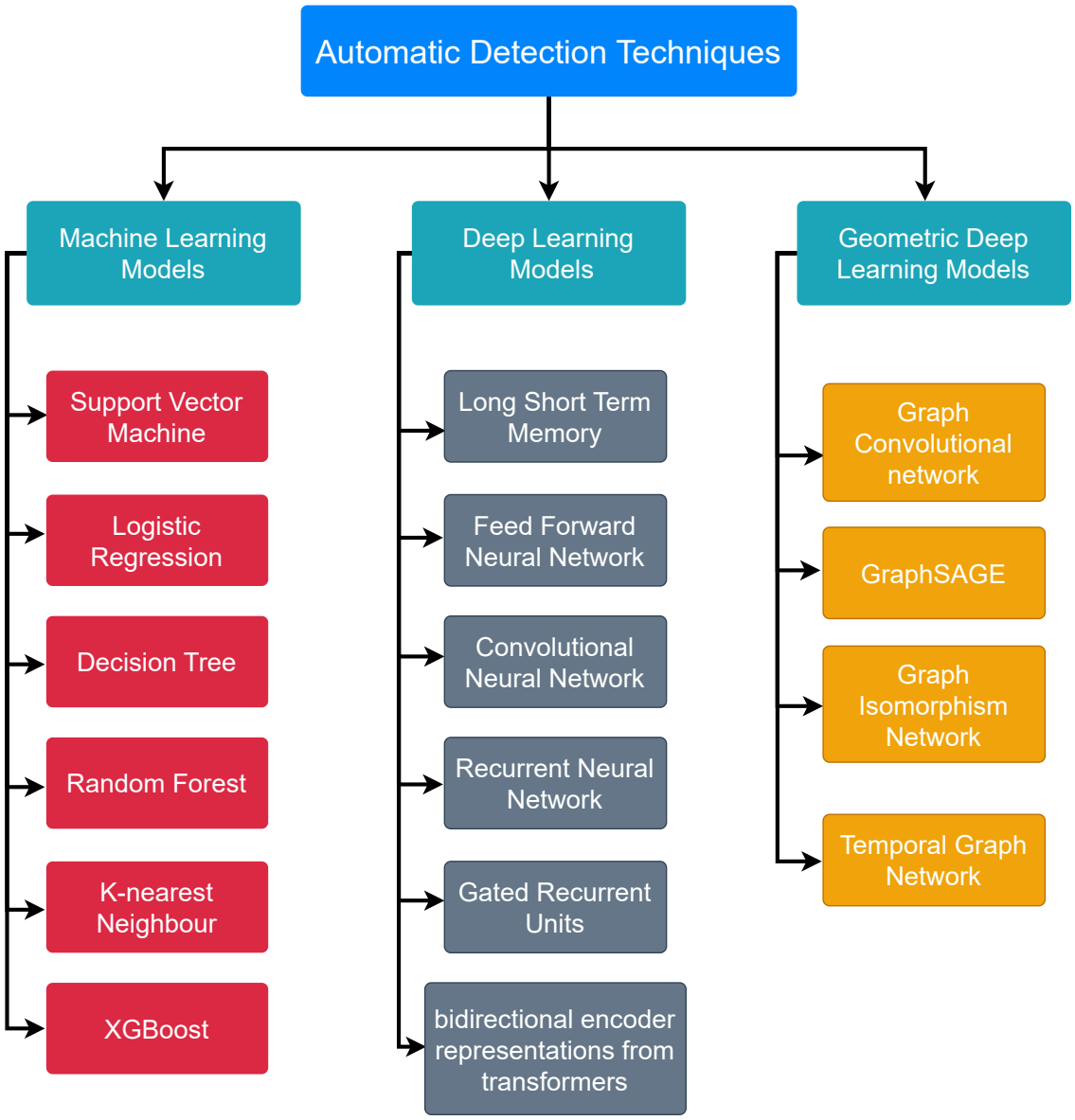



Figure 2.2: Automatic detection methods

from given data. Visualization can be considered a flow chart where nodes represent features, branches represent rules, and leaf nodes are outputs. The rules are learned with a supervised iterative process. The algorithm is suitable where data possess clearer mapping between rules and results. The algorithm is used for both classification and regression tasks.

4. **Random Forest:** Random forest is an extension of the Decision Tree algorithm

for more complex learning tasks. With many decision trees coupled together, it virtually makes a forest shape. The algorithm works on the principle of selecting random data from available datasets and trains several decision trees. At the inference time, results obtained from different trees are pooled to form a finalized result.

5. **K-nearest neighbour:** K-nearest neighbor is a simple yet effective supervised machine learning algorithm that performs prediction tasks based on the proximity of the targeted data point. The value K represents the number of neighbors being considered. It can be intuited as if two things are similar in many ways; they are likely to be identical in other ways.

6. **Naive Bayes:** Naive Bayes is a supervised learning algorithm based on the Bay's rule. Naive refers to the assumption that features in data are conditionally independent. The naive Bayes algorithm and its variations are widely used in Natural Language Processing (NLP) applications.

7. **Ada-Boost:** Ada-Boost works on the concept of virtually booting the performance, which is to combine weak classifiers to form a strong one. It has been mainly used for binary classification tasks. The working principle for the model is to train a model from the available training dataset and then build a second model to overcome performance deficiencies in the first model.

### 2.2.2 Deep Learning

Deep Learning (DL) is a sub-area in the Machine Learning (ML) study area. It is known for directly processing human understandable data (structured and unstructured).

As per its utilization in our problem, we consider DL to provide an inference method with a suitable data representation. In short, DL acts as a bridge between the

available data and the downstream inference method. In a nutshell, DL transforms the data into a suitable form that inference methods can efficiently process. This 'suitable form' is generally called 'data representation'.

DL methods substantially vary among different types of data. For example, for image data, different variations of CNNs have been a famous choice in practice. RNNs are widely used to process data with some sense of sequence. Likewise, methods from the area of Geometric Deep Learning are dedicated to graph-structured data.

### 2.2.3 Geometric Deep Learning

Geometric Deep Learning (GDL) is a branch of Deep Learning (DL) that deals with network graph data. The area comprises methods to find low dimensional representations from data encoded as graph structures. There are two major types of strategies: Graph Kernels and Random Walk based approaches and Graph Neural Networks (GNNs). GDL methods have popular applications in Bio-Chemistry, Recommender Systems, and processing of Large Social Networks.

In recent years, the graph neural network (GNN) models have shown state-of-the-art results on the graph data. Researchers incorporate social context features and utilize GNN models to identify misinformation spreaders. The most used models are graph convolutional network (GCN), GraphSAGE, Graph isomorphism network, Temporal graph network, and many more.

## 2.3 Literature Review

As in section 1.4, we discussed that in existing literature, detecting fake news or false information could be classified into three broad categories based on content, social context or propagation-based, and hybrid. In this section, we give a comprehensive discussion of the existing work.

### 2.3.1 Content-based Detection

In literature, plenty of solution has been proposed based on news content. The misinformation has been identified by analyzing the contents of the news articles (like text images). The proposed approaches use machine learning methods to extract stylometry features such as sentence length, sentence segmentation, part-of-speech tagging, and tokenization and linguistic features such as sentimental feature, word frequency, and bag-of-words [Wu et al., 2019]. Based on these features, the authenticity of the information is identified.

Many authors have collected users' posts and manually labeled them as misinformation or real information. They have extracted different features and used machine learning methods (SVM, Random-Forest, Naive Bayes, Decision Trees, Neural Networks) to identify the user's posts as real or fake. For example, [Helmstetter and Paulheim, 2018] have collected hundreds of thousands of tweets and automatically labeled them as trustworthy or untrustworthy sources based on their origin source. Furthermore, they have extracted user-level features (frequency of tweets), tweet-level features (word count), text features (bag-of-words utilizing TF-IDF), topic features, and sentimental features for the detection of false information. However, the dataset they have used comes with several challenges, which are noisy and inaccurate data. Recently, MediaEval introduced a benchmark dataset for misinformation detection in a workshop entitled "FakeNews: Coronavirus and 5G conspiracy". Using this dataset, the work [Moosleitner et al., 2020] uses n-gram to extract textual features for distinguishing real or misinformation. The performance using n-gram for feature extraction does not provide promising results because it did not capture the dependencies in long sentences. Bi-LSTM models capture long dependencies in the text to achieve good results as [Raj and Mehta, 2020] have used Bi-LSTM to identify real or misinformation.

Similarly, BERT is used for both capturing long dependencies and attention mech-

anisms to adjust various weights. In the study [Andrey Malakhov, 2020], they have used the BERT model to extract textual features for elegant detection of misinformation using a simple linear layer. Like-wise [Kaliyar et al., 2021] used BERT to incorporate text semantics in detecting misinformation by utilizing CNN layers with different kernel sizes. Similarly, in the study [Amer et al., 2022], they performed comparative experiments using machine learning classifiers, deep learning models like LSTM GRU, and transformers like BERT. They have utilized word embeddings(Word2vec, GloVe) in these experiments.

Furthermore, they conclude that the deep learning model outperforms machine learning classifiers, and the BERT transformer model outperforms in accuracy. The word2vec does not provide embeddings for out-of-vocabulary; therefore, Facebook developed FastText to provide embeddings for out-of-vocabulary and faster than word2vec. Thus, the proposed work [Hathnapitiya et al., 2023] utilized FastText embeddings and Recuurent-Nerual Network (RNN) to differentiate between real and misinformation. However, there are some challenges to identifying misinformation based on their contents. As we discussed in section 1.4, the main challenge of content-based detection is that tuning the model parameters on one dataset cannot perform well on another dataset because of the contents domains, such as if we are utilizing COVID-19-related content and fined tuned the weights of the model. Then, testing on political datasets may not outperform because of the difference in their contents, text style, and even language [Ullah et al., 2023].

### 2.3.2 Social Context or Propagation-based Detection

To tackle these problems and these challenges in detecting misinformation spreaders, researchers move to the social context or propagation-based solution. Propagation-based methods try to model how information spreads on social media over time. It has been demonstrated that tracking how information stories circulate on social media,

e.g., tweets, retweets, quotes, and replies on Twitter, can improve the performance of fake news detection models. The propagation patterns of misinformation and accurate information have been analyzed to be different [Liu and Wu, 2018, Vosoughi et al., 2017, Shu et al., 2020b]. Similarly, the study [Vosoughi et al., 2018] examined how truthful and misleading information spreads over social media. After extracting several propagation-based properties, the authors discovered that false information diffuses more quickly, deeply, and widely than truth information across all categories. The author has identified that incorrect information is more likely a novel (fiction stories). Therefore, they conclude that people like novels, indicating that people are likelier to share false information. Plenty of studies are based on propagation-based methods, which used a variety of models such as geometric deep learning [Malhotra and Vishwakarma, 2020, Monti et al., 2019], propagation tree kernel [Ma et al., 2017], RNNs [Liu and Wu, 2018], and CNN [Liu and Wu, 2018] that have been utilizing the propagation-pattern of news or information.

In work [Shu et al., 2019], they have proposed a TriFN (tri-relationship embedding) framework that models the relationship among publishers, news pieces, and users. These features have the potential to improve the performance of misinformation detection. They have utilized the non-negative matrix technique that user-news source interaction, news-source interaction, and user-user interaction. However, this method can be expensive regarding graph node counts and unscalable for graphs that expand over time due to a lack of inductivity. However, the model outperforms other baseline models. GNNs have been intensively used in recent years for inference problems with graph data and have produced outstanding results in many application domains, such as protein-protein interaction. Therefore, in the work [Monti et al., 2019], they have developed a model utilizing geometric deep learning with two graph convolutional layers, two fully connected layers, and a soft-max layer for misinformation detection. To fine-tune the parameters of the proposed model, they have utilized

news stories that spread on Twitter and verified by a fact-checking organization.

Furthermore, in manuscript [Nguyen et al., 2020], the author has proposed a detection framework named Factual News graph (FANG). The proposed framework utilizes GNNs to model user interactions, news, and news sources. Similarly, they have examined that geometric deep learning methods, such as GNNs, are well suited for capturing the user interaction landscape, including the social figures they follow, the news subjects they like or oppose, etc. Therefore, the author generated a heterogeneous graph such that each node represents a news source, a news item, or a social user. Various connections between nodes take advantage of particular characteristics. For instance, an advantage of user news is the user's opinion of the news item. The nodes are labeled as false or real. 0 represents Real news, and 1 illustrates False information. They have proposed a framework with three loss functions. (a) Unsupervised proximity loss used in a social context. Suppose two graphs are highly related in a social context. In that case, their loss will be slight and cause similar. (b) self-supervised Stance Loss represents the user's stance towards the news, and (c) Supervised Fake news loss is used to enhance the model performance of fake news detection. The task of the proposed framework is to represent news based on its integrity and differentiate between fake news and real news. Similarly, in the 2020 MediaEval workshop, they presented a benchmark dataset entitled "FakeNews: Coronavirus and 5G conspiracy," in which different participants proposed methodologies for the fake news detection problem. The author [Tuan and Minh, 2020, Schaal and Phillips, 2020] utilized a variant of the GNN model(Graph convolutional neural network) to capture the propagation structure of the information. Still, the performance of this model is not good in terms of MCC and accuracy. Similarly, the author of paper [Shu et al., 2020b] proposed a mode named Hierarchical propagation networks for fake news detection (HPFN). The author constructs the propagation pattern from the news article. The author extracts two types of propagation patterns. The first one

is how the news spreads on social media (reposting), and the second propagation pattern is extracted based on comments and replies. The author extracted three feature structural, temporal, and linguistic features. Based on these feature and propagation graphs, they classify the news as real or fake. The author shows that the model improves performance.

The attention mechanism gives more weight to the relevant and less weight to the less relevant parts. This consequently allows the model to make more accurate predictions by focusing on the most essential information.

The attention mechanism in Natural Language processing improves the performance of those tasks. Therefore, the attention mechanism is introduced in GNNs, in which high weights are given to relevant and small weights are assigned to less relevant parts. Thus, it makes accurate predictions by focusing on important information. Therefore, the author [Silva et al., 2021] proposed a method named Propagation2Vec, a cutting-edge propagation-based fake news detection algorithm. They have developed a hierarchical attention mechanism to encode the propagation pattern. The author combines a complete and partial graph network, combines the embeddings generated by Propagation2Vec of these two, and gives it to a classifier. However, the temporal feature is essential in identifying misinformation spreaders because the graphs evolve with respect to time. Therefore, in manuscript [Song et al., 2021], the author has proposed a novel temporal propagation-based fake news detection framework that focuses on the temporal feature of news propagation. The presented framework combined time information, content semantics, and structure. They have developed a graph-based methodology on the temporal aspect of the news and utilized the temporal graph attention neural network (TGAT) to identify misinformation spreaders.

The user profile feature can be utilized to identify misinformation spreaders. Therefore, in work [Verma and Agrawal, 2022], the author has proposed a Propagation-

based Fake News Detection (PropFND) model. The task of PropFND is to identify the news as real or fake based on their propagation graph and user profile feature. For the fine-tuning of the model, the author combined the features and utilized different classifiers, such as SVM. They conclude that SVM outperforms state-of-the-art models. They have noticed that the real news is propagating for an extended period compared to the fake news.

Similarly, in the work [Wu and Hooi, 2023], they have proposed a model named DECOR. The work's main contribution is that he generates a method to decrease the weight of noise edges via learning a social degree correction mask. Then, he fine-tunes GNN models such as GCN, GIN, and GraphConv.

### 2.3.3 Hybrid Models

Hybrid methods are the combination of both content and propagation-based feature models. Due to challenges in content-based, the researcher moved to the propagation-based solution. In contrast, some researcher worked both on content-based combined propagation-based, which again make it domain-specific as the textual feature are related to a specific domain. Many researchers combine contents and their social context, such as propagation patterns. They used BERT models to extract textual features and fine-tuned a machine-learning model for the contents. For the propagation feature, they used geometric deep learning to capture the network embeddings, concatenate the embeddings of both models, and utilize a machine learning classifier to identify misinformation. In the work [Matsumoto et al., 2021], they have fine-tuned a Graph Transformer Network with two Mlp Layers on Politifact and GossipCop datasets. To extract linguistic features from contents, they have used the BERT model. They have claimed that the model gives promising results. Similarly, the author [Raza and Ding, 2022] used an encode decoder model for content and social context features. They have claimed that experimental results show that the

model gives high accuracy. Furthermore, the author [Saikia et al., 2022] proposed a hybrid model combining content and context-based features. They used a BERT to extract features from contents and a graph convolutional network to learn context- or propagation-based feature embeddings and concatenate them. The authors called it feature fusion and passed the resultant embeddings to a neural network to distinguish between real and misinformation.

## 2.4 Datasets

This section discussed some benchmark datasets used for fake news detection. Following are some popular datasets used for fake news detection.

### 2.4.1 FakeNewsNet

The author published the dataset [Shu et al., 2020a]. The author collects multiple pieces of news from fact-checking websites PolitiFact (includes news items about politics ) and GossipCop ( includes entertainment articles) to obtain ground-truth labels of the news. After collecting news articles and their ground truth, the author uses Twitter's Search API to fetch the user who directly shares the post with news headline titles. Thenfetech, the user, responds to the post with replies, retweets, and likes. The author also fetches the user's metadata, like their profile feature, and captures their social engagement.

### 2.4.2 BuzzFeedNews

The BuzzFeedNews dataset is used in the paper [Horne and Adali, 2017]. This dataset contains the news or posts and their title and is labeled as fake or real. The data were collected from three sources in the 2016 US presidential election. The dataset contains only text data and is valuable for testing linguistic approaches for detecting fake

Table 2.1: Literature review

| Reference | Year | Content | Social Context | Technique | Dataset | Results |
|---|---|---|---|---|---|---|
| [Ma et al., 2017] | 2017 | ✓ | ✓ | Kernal-based methods GRU, SVM-TS | Twitter15 | Acc: 0.75 |
| [Helmstetter and Paulheim, 2018] | 2018 | ✓ | ✗ | Naive Bayes, Decision Trees, Support Vector Machines (SVM), Neural Networks, Random Forest and XG-Boost | Self-extracted data | F1: 0.94 |
| [Liu and Wu, 2018] | 2018 | ✓ | ✓ | recurrent neural networks and convolutional neural networks | Weibo, Twitter15, Twitter16 | Acc: 0.92, Acc: 0.84, Acc: 0.86 |
| [Monti et al., 2019] | 2019 | ✗ | ✓ | GCN | self extracted dataset from snoops, politifact, buzz feed | ROC-AUC: 92.70 |
| [Shu et al., 2019] | 2019 | ✓ | ✓ | non-negative matrix decomposition | BuzzFeed, PolitiFact | Acc:0864 and F1:0.88, Acc:0.878 and F1:0.88 |
| [Moosleitner et al., 2020] | 2020 | ✓ | ✗ | n-gram + SVM | Media Eval Fake news dataset | MCC: 0.432 |
| [Raj and Mehta, 2020] | 2020 | ✓ | ✗ | SVM, NB, KNN, LSTM, Bi-LSTM | Media Eval Fake news dataset | MCC: 0.4179 |
| [Silva et al., 2021] | 2020 | ✓ | ✓ | GraphSAGE | PHEME | AUC:0.75 |
| [Tuan and Minh, 2020] | 2020 | ✓ | ✓ | GCN, BERT | Media Eval Fake news dataset | MCC; 0.151 |
| [Malhotra and Vishwakarma, 2020] | 2020 | ✓ | ✓ | LSTM, RoBERTa, GCN | Twitter 15,Twitter 16 | ACC: 0.866, Acc: 0.865 |

| Reference | Year | Content | Social Context | Technique | Dataset | Results |
|---|---|---|---|---|---|---|
| [Tuan and Minh, 2020, Schaal and Phillips, 2020] | 2020 | ✓ | ✓ | GCN, BERT | Media Eval Fake news dataset | MCC: 0.1375 |
| [Kaliyar et al., 2021] | 2021 | ✓ | ✗ | CNN + BERT | COVID-19 Fake News | Acc: 98.90 |
| [Silva et al., 2021] | 2021 | ✗ | ✓ | GCN | Politifact, GossipCop | Acc:0.879 and F1: 0.893, Acc: 0.892 and F1: 0.874 |
| [Matsumoto et al., 2021] | 2021 | ✓ | ✓ | BERT + Graph tranformer Netwrok + 2 MLP layers | Politifact, GossipCop | Acc:0.9379 and F1: 0.9132, Acc: 0.9535 and F1: 0.9064 |
| [Song et al., 2021] | 2021 | ✗ | ✓ | Temporal Graph Attention (GNN) | Weibo,FakeNewsNet, Twitter | Acc: 0.968, Acc: 0.935, Acc:0.923 |
| [Raza and Ding, 2022] | 2022 | ✓ | ✓ | Encoder Decoder | NELA-GT-19,Fakeddit | Acc: 0.748 and F1:0.749 |
| [Amer et al., 2022] | 2022 | ✓ | ✗ | LSTM, GRU, BERT | ISOT | Acc: 0.991 |
| [Saikia et al., 2022] | 2022 | ✓ | ✓ | (BERT + 3 mlp & GCN + pooling + 1 MLP layer1) + 2 mlp layers | Politifact, GossipCop | Acc:0.917 and F1: 0.915, Acc: 0.933 and F1: 0.930 |
| [Rawat et al., 2023] | 2023 | ✓ | ✗ | count vectorizer + (Random Forest, SVM and Nave Bayes) | Liar | Acc: 70 |

news. The author concludes that the title structure and proper nouns are important distinguishing features between fake and real. They also conclude that fake news is specially targeted at those people who do not read beyond the titles.

### 2.4.3 LIAR

The LAIR dataset introduced in paper [Wang, 2017] by researchers at the University of California, Santa Barbara, as a benchmark dataset. The data are collected from the fact-checking website named POLITIFACT. The dataset contains 12.8K short statements labeled as truthful, false, or 'pants on fire' (i.e., a flagrant lie). The dataset can be used for identifying surface-level linguistic patterns. The limitation of the dataset is that it only has one category of material (political statements).

### 2.4.4 PHEME

The dataset [Kochkina et al., 2018] contains the Twitter threads about the eight newsworthy events between 2014 and 2015. Each thread has the tweet and their retweets, replies, and mentions. Each tweet is classified as Rumers and Non-rumors. The dataset has been utilized in various social media analyses and misinformation identification projects.

### 2.4.5 CREDBANK

The CREDBANK dataset presented in paper [Mitra and Gilbert, 2015] is a collection of Twitter posts. The dataset contains tweets, topics, and events. The dataset consists of more than 60 million tweets grouped into 1049 real-world events, each annotated by 30 human annotators. Researchers can gain insights into how rumors and disinformation propagate and how credibility might be estimated by analyzing the linguistic content of social media posts.

### 2.4.6 COCO

The COCO dataset [Langguth et al., 2023] is a collection of Twitter posts. The dataset contains various tweets covering various topics, including conspiracy theories related to the COVID-19 pandemic. The dataset includes 3495 tweets, which were manually labeled concerning 12 different categories. It contains the textual content of the tweets.

### 2.4.7 WICO Text

The WICO Text dataset [Pogorelov et al., 2021] is a collection of Twitter posts. The dataset contains various tweets covering various topics, including conspiracy theories related to the COVID-19 pandemic and 5G technology. The dataset contains more than 10,000 tweets manually labeled concerning four different categories. The tweets are classified into Non-conspiracy, 5G conspiracy, Other conspiracy, and Undecidable.

### 2.4.8 WICO Graph

The WICO (Wireless Networks and Coronavirus Conspiracy) dataset [Schroeder. et al., 2021] contains the subgraphs extracted from 3,000 manually classified Tweets from Twitter. The subgraphs are generated based on follower networks and distinguished into three categories: 5G misinformation, those that spread other conspiracy theories, and Tweets that do neither. Each node represents a user on Twitter, and the attributes of each node are its profile features.

Table 2.2 shows the dataset used for fake news detection. There are lots of datasets used in this area, and the names of these datasets are FakeHealth [Dai et al., 2020], Fake_or_real_news, COVID-19 Fake News[1], twitter16, twitter16 and many more. We cannot cover all the datasets in this desertion.

[1] `https://www.kaggle.com/datasets/arashnic/covid19-fake-news/discussion`

Table 2.2: Different Dataset Used for fake news detection

| Dataset Name | Area | Dataset Type | Content Type | # of classes |
|---|---|---|---|---|
| FakeNewsNet | Entertainment, politics and celebrities | News | Text and Images | 2 |
| BuzzFeedNews | Politics | News | Text | 4 |
| LIAR | Politics statements | News | Text | 6 |
| PHEME | Society and politics | Rumors | Text | 2 |
| CredBank | Society | Rumors, Misinformation | Text | 2 and 5 |
| FakeCovid | Health and Society | News | Text | 11 |
| LIAR | Politics statements | News | Text | 6 |
| Yelp | Reviews | False Information | Text | 2 |
| COCO | Conspiracy Theories about COVID-19 | Misinformation | Text | 12 |
| WICO Text | Conspiracy Theory and 5G-Corona Misinformation Tweets | Misinformation | Text and Graph | 2 and 4 |
| WICO Graph | Conspiracy Theory and 5G-Corona Misinformation Tweets | Misinformation | Graph | 4 |

## 2.5 Summary

This chapter provides a brief overview of fake news detection methods. Then, we study the literature review of these methods, and finally, in section 2.4, we briefly discuss some of the datasets used in this area.

# Chapter 3

# Proposed Method

Graph Neural Network is a cutting-edge technology to use for non-Euclidean data, especially for graphs. Misinformation spreaders are a threat to society and democracy. It is, therefore, very important to pay attention to the Misinformation Spreader detection problem to avoid catastrophic events or losses. The study focuses on the identification of misinformation spreaders by using their social interactions and the graph neural network. For classification or identification of misinformation spreaders, we utilize the user social profile features like the number of followers following, etc., and their propagation structure. This chapter presents a detailed discussion of the proposed method.

## 3.1 Proposed Method

In this section, we present the details of our proposed method as shown in Figure 3.1 to show the flow of our proposed process. Our proposed method consists of multiple steps, including Prepossessing, Neighbourhood Sampling, dividing the data into train-validation sets, and finally, we have trained the proposed model.

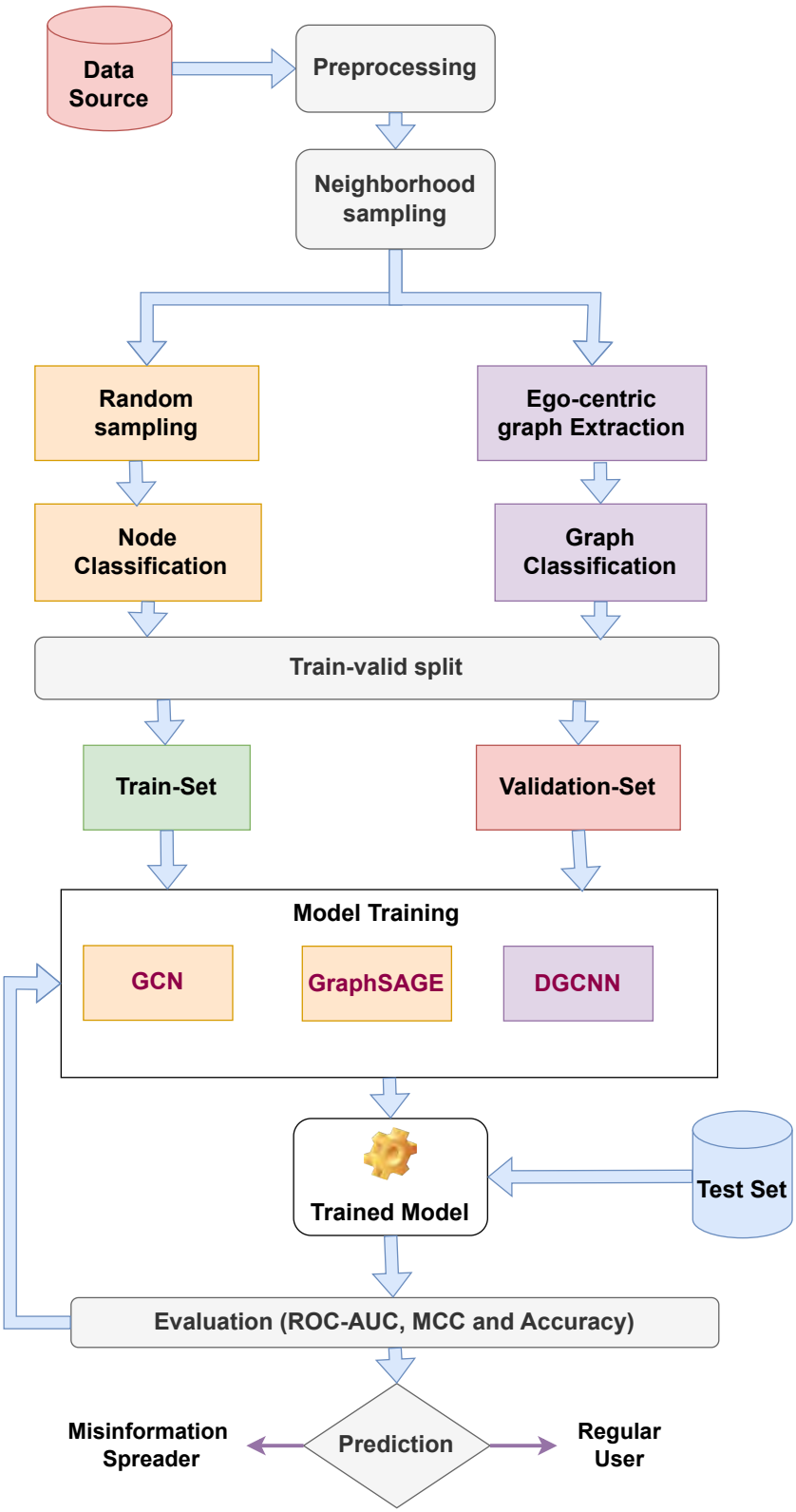



Figure 3.1: Proposed methodology

### 3.1.1 Data-Source

For the evaluation of our model, we use a dataset provided in the MediaEval 2022 Workshop[1]. The data are collected from Twitter through the FACT (Framework for Analysis and Capture of Twitter Graphs). The dataset consists of users who tweet information and their social engagements, like most likely retweet propagation path, their replies, quotes and quoted replies. The dataset consists of a single graph with node labels regular user, misinformation spreader user, or unknown. The dataset creation process is comprised of a multi-stage process. They started with creation of a collection of tweets related to the COVID-19 pandemic from Twitter in a time span between January 17, 2020, and June 30, 2021. Then, they selected and manually labeled 3389 tweets. The final dataset is a graph with 1.6 million nodes and 268 million edges, along with 1913 and 830 node labels distributed in development and test sets, respectively.

### 3.1.2 Preprocessing

While developing a computer model, we go through a series of steps. Which is data collection, Preprocessing, embedding learning, and many more. After the data collection, we perform preprocessing steps on the data. We consider the profile feature of a user (number of followers, numbers of following, description length, account verified, etc.), which consists of empty values and categorical data. The basic method for feature Imputation is to fill the values of the missing features with four possible methods; Feature initialization with a random values from standard Gaussian distribution, feature initialization with 0s. Two more possibilities are global and neighbourhood mean over the graph. The global mean of that feature calculates mean over the complete graph (Global Mean) and the neighbourhood mean is the mean over the features of neighbours of a node [Rossi et al., 2022]. We try the two basic methods: the first

[1] `https://multimediaeval.github.io/editions/2022/`

is setting the missing value with zero, and the second is setting the missing value with global mean. For the categorical data, we use Label Encoding [Hancock and Khoshgoftaar, 2020], which is a technique that involves assigning numerical values to different categories of categorical data. In this method, each unique category is assigned a specific integer value. For example, the dataset has a categorical variable with possible values of {"US", "U", and "Spain"} label encoding would assign the corresponding integer mapped values of {0, 1, 2} to these categories.

## 3.2 Neighbourhood Sampling

Before learning the embeddings, we sample the graph. We use two strategies to sample the graph. These strategies are discussed below.

### 3.2.1 Random Sampling

In the first strategy, we consider the whole graph as one graph and randomly drop nodes and edges, which reduces the graph size. Then, we learn embedding for each node. Algorithm 1 contains the algorithm of Random sampling. The inputs to algorithms are edges_list, which contain the edges of the graph, and the second one is drop_ratio, which means how many edges should be drooped.

**Algorithm 1** Random Sampling

**Require:** *Input* : edge_list , drop_ratio
  updated_edge_list = []
  Iterator = 0
  **while** *edge* ∈ *edge_list* **do**
    **if** Iterator ÷ drop_ratio = 0 **then**
      updated_edge_list.append(edge)
      Iterator = Iterator + 1
    **end if**
  **end while**
  **Return** updated_edge_list

### 3.2.2 Ego-centric graph

In the second strategy, we divided the whole graphs into sub-graphs. Out of 1679011 nodes, we have 1,913 nodes labeled, and the remaining is unknown. We sample the graph into sub-graphs around each labeled node up to 3-hop called ego-centric graphs. Then, we consider a specific number of nodes at each hop. In the 1-hop, we consider ten thousand neighbor nodes; in the 2-hop, we consider one thousand nodes; and in the 3-hop, we consider five hundred neighbor nodes. Figure 3.2 shows an example of an ego-centric graph generated in the proposed methods.

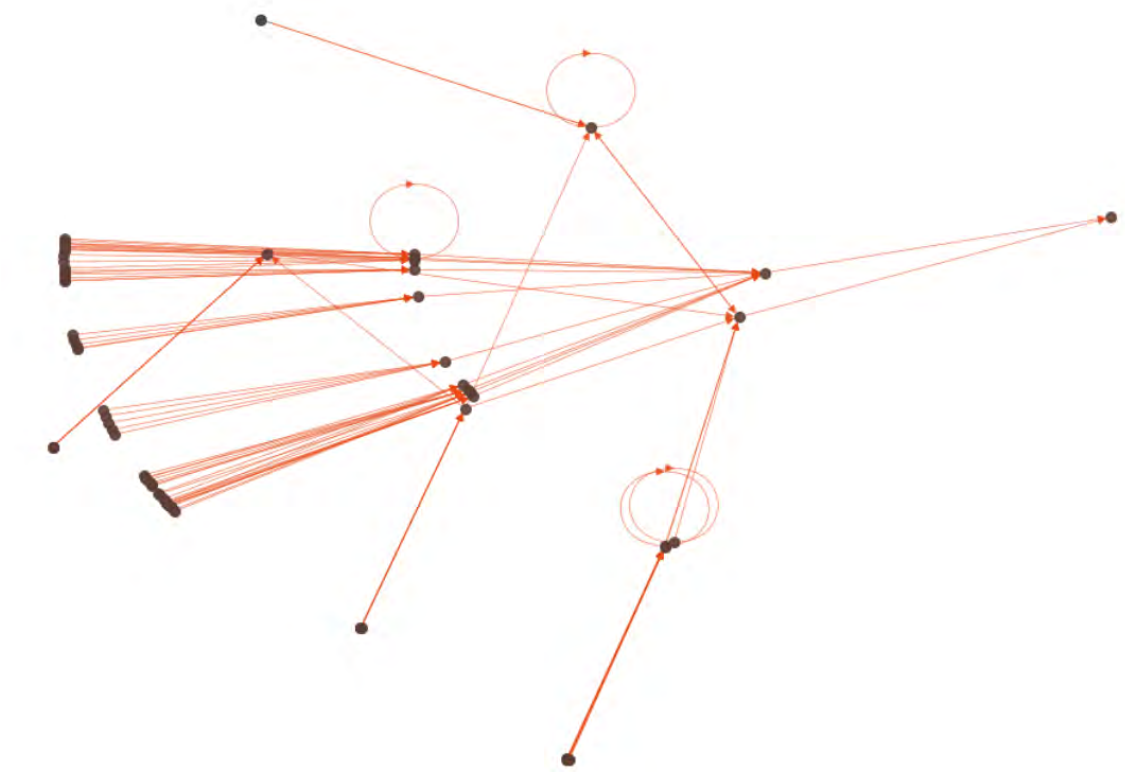


Figure 3.2: Ego-centric graph

Given Algorithm 2 contains the algorithm of the ego-centric graph extraction. Where G represent the whole graph, EN represents A set of Nodes for which an ego-centric graph is required to generate, K denotes the number of hops for each node, and Neighbors denote a set of neighbor to be sampled at each hope Example:{

10000,1000,500 }.

**Algorithm 2** Neighbour Sampling (Ego-centric graphs)

**Require:** $Input : G, EN, k, hop_neighbours, edges\_list$

```
y ← 1
X ← x
Ego ← n
sub_graphs ← array()
while node ∈ EN do
    node_and_neighbours ← [node]
    new_edge_list ← array()
    for hop = 1 , hop <= K , hop++ do
        number_of_edges ← hop_neighbours[k]
        while n ∈ node_and_neighbours do
            new_edge_list.append(select number_of_edges that contain 'n' node)
        end while
        node_and_neighbours ← unique(new_edge_list)
    end for
    sub_graphs.append([new_edge_list])
end while
Return sub_graphs
```

## 3.3 Modelling propagation graph using GNN

Graph Neural Networks (GNN) is a kind of neural network which relies solely on graph structures. Twitter and other social media sites can be modeled as graphs. A GNN provides a practical solution for node level, edge level, and graph level prediction tasks like node classification, Graph classification, Community detection, and Network similarity. The basic architecture of GNN works on neural message passing or neighborhood aggregation in which vectors are exchanged between nodes and updated using a neural network. The input to GNN is a graph G = (V, E) along with a set of node features X $\in R^{|V|\times d}$ with adjacency matrix $A \in \{0,1\}^{|V|\times|V|}$. This information is used to generate embeddings $z_u, \forall_u \in V$. GNN framework is a sequence of message passing Iterations with **UPDATE** and **AGGREGATE** functions. Every single node in the graph aggregates messages from neighbor nodes, and it updates its

node representation. The GNN layers can be described as:

$$h_u^{(0)} = X_u \tag{3.1}$$

where $h_u^{(0)}$ is the initial embedding of node u at layer-0 which is the input feature vector of $X_u$.

$$\begin{aligned} h_u^{(k+1)} &= UPDATE^{(k)}(h_u^{(k)}, AGGREGATE^{(k)})(\{h_v^{(k)}, \forall \in N(u)\})) \\ &= UPDATE^{(k)}(h_u^{(k)}, m_{N(u)}^{(k)}) \end{aligned} \tag{3.2}$$

where $h_u^{(k)}$ is the of node u at layer-k getting information from its neighbor up to k-hop.

$$Z_v = h_v^{(k)} \tag{3.3}$$

Where $Z_v$ is the final representation of node embeddings at the final layer, in contrast, the UPDATE and AGGREGATE are two ordinary functions. The AGGREGATE function actually aggregates the collected node representation from their local neighbor nodes, and the UPDATE function updates the current node feature based on the aggregated features. Figure 3.3 demonstrates how GNN works.

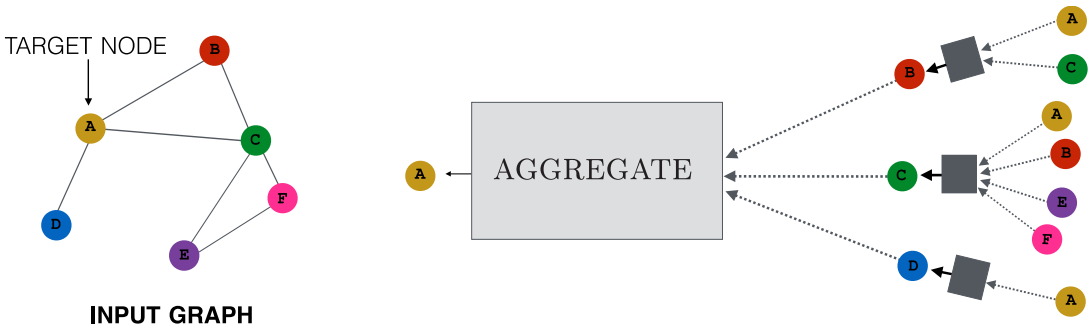



Figure 3.3: Demonstrate how nodes aggregate messages from their neighbor. The model aggregate messages from the neighbors of A, which are (B, C, and D); the message of these neighbors are aggregated based on their prospective neighbor and soon. [Hamilton, 2020]

The purpose of this study is to extract the propagation feature of the misinforma-

tion spreader by using GNN models. The working models can be defined as: Given a set of tweets, retweets, quotes, and replies, how efficient the propagation feature is to classify a tweet. We apply different state-of-the-art GNN models to model the propagation feature of the misinformation spreader. These GNN models are GCN, GraphSAGE, and DGCNN. We will further explain these models below.

### 3.3.1 Node Level Embedding

**Graph convolutional network (GCN)** is a type of GNN that is used for graph data. It is an effective variant of a convolutional neural network and a semi-supervised learning model introduced by [Kipf and Welling, 2016]. In this work, message-passing techniques are used to introduce convolutions in a graph structure, and information from nearby nodes is aggregated using a weighted average function. A localized first-order approximation of the spectral graph convolutions serves as motivation for the selection of the convolutional architecture. The model learns hidden layer representations that encode both local network structure and node attributes and scales linearly as the number of graph edges increases. It is an effective framework to implement convolution on graph data. The author introduced a re-normalization trick with a degree matrix to solve numerical instabilities and exploding/vanishing gradients. The layer-wise propagation of the model is as follows:

$$H^{(k)} = \sigma(\hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}}H^{(k-1))}W^{(k-1)}) \tag{3.4}$$

Where $H^{(0)} = X$, which denotes the representation of node embeddings at layer 0 while $H^{(k)}$ represents the latent representation at layer kth layer and $\hat{A}$ represents an adjacency matrix added with self-loop ($\hat{A} = A + I_N$). Where $l_N$ is the identity

matrix. while $\hat{D}$ is the diagonal degree matrix

$$\hat{D}_{ij} = \begin{cases} \sum_j \hat{A_{ij}}, & \text{if } i = j \\ 0, & \text{otherwise} \end{cases} \tag{3.5}$$

$W^{(k)}$ is a trainable weight matrix at k-th layer and $\sigma$ represent the activation functions i.e ReLU, Tanh etc. GCN performs well in node classification and link prediction, which attracts the researcher's attention.

**GraphSAGE (SAmple and aggreGatE)** is an inductive framework that generates embeddings for unseen nodes. Unlike transductive models, which used matrix-factorization-based objectives to optimize the embeddings, GraphSAGE leverages node features to learn embedding functions for unseen nodes. This work introduced three generalized aggregated functions (mean aggregator, LSTM, and Max pooling for the neighborhood aggregation) used to aggregate neighborhood information. Thus, make it an inductive model. The work is introduced in paper [Hamilton et al., 2017]. The working mechanism of the model is:

$$\acute{x_i} = W_1 x_i + W_2 . mean_{j \in N_{(i)}} x_j \tag{3.6}$$

Where $\acute{x_i}$ updated embeddings of node i, which we get by combining the embeddings of node and its neighbors from previous Iteration. Figure 3.4 shows the GraphSAGE sample and aggregate approach.

**Linear Layer** , also known as a fully connected layer or dense layer, is a fundamental component in neural networks and deep learning models. It performs a linear transformation on input data by applying a set of weights to the input features and then adding a bias term. The result of this transformation is often passed through an activation function to introduce non-linearity into the network.

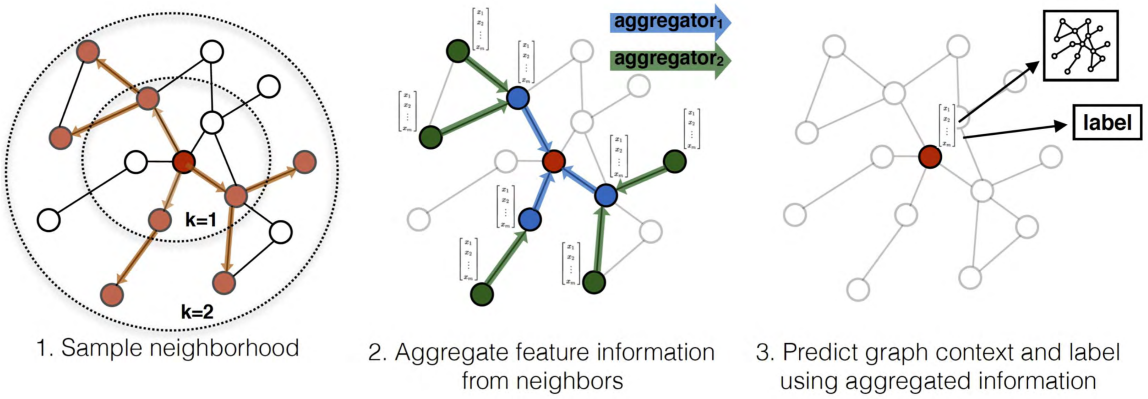



Figure 3.4: GraphSAGE architecture [Hamilton et al., 2017]

$$Y = WX + b \tag{3.7}$$

Where $X$ is a feature matrix of size $(j, )$, $W$ contain trainable weights of size $(k, j)$, in which $k$ is the number of neurons in the layers and finally a bias $b$ with size $(k, )$ is added to each neuron, and $Y$ is the output vector of the linear layer. Furthermore, activation functions like ReLu and Tanh introduce non-linearity in the linear layer.

The output of GCN and GraphSAGE passed to the linear layer for further transformation to produce a final prediction. The linear layer allows further refining of the learn representation in GNN models to enhance model performance. It's something like message passing + Neural Network, which leads to improved model performance for downstream tasks.

### 3.3.2 Graph Level Embedding

**Deep Graph Convolutional Neural Network (DGCNN)** introduced by [Zhang et al., 2018] to generate embeddings on graph-level. It uses a readout or polling layer to learn embeddings on the graph level. DGCNN consists of three parts.

1. **Graph Convolution layers** which generate embeddings for each node at each layer by aggregating local neighborhood information and then concatenating all

the embeddings from all Iteration.

2. **SortPooling layer** is used to sort the vertex features instead of applying aggregation like sum, max, and min to keep much more information and to allow the model to learn the graph-level features.

3. **Traditional Convolution layer and Dense layers** is used to read the sorted graph representation or embeddings and develop a predictive model.

## 3.4 Learning Parameters

To calculate the loss at the node level, we used BCELOSS, which stands for Binary Cross-Entropy Loss. BCELOSS develops a criterion to calculate the Binary Cross Entropy of the input and target probability. The BCELOSS can be described as:

$$l(x, y) = L = \{l_1, ..., l_N\}, l_n = -w_n[y_n log x_n + (1 - y_n) log(1 - x_n)] \tag{3.8}$$

$x$ denotes the actual value or label, and $y$ represents the predicted value, and it must be between 0 and 1.

Where N is a batch size, and $l_n$ is a loss at each batch. To calculate a loss on the whole dataset, we can take a sum or mean of the loss as shown in equation 3.9.

$$l(x, y) = \begin{cases} mean(L), & \text{if } reduction =' mean'; \\ \text{sum(L)}, & \text{if } reduction =' sum'; \end{cases} \tag{3.9}$$

After computing the loss, the gradients decent of the loss with respect to the parameters in each layer are calculated using the chain rule of calculus. The gradients quantify the sensitivity of the loss with respect to changes in the network's parameters. Once the gradient is computed, we apply an Admin Optimizer [Kingma and Ba, 2014],

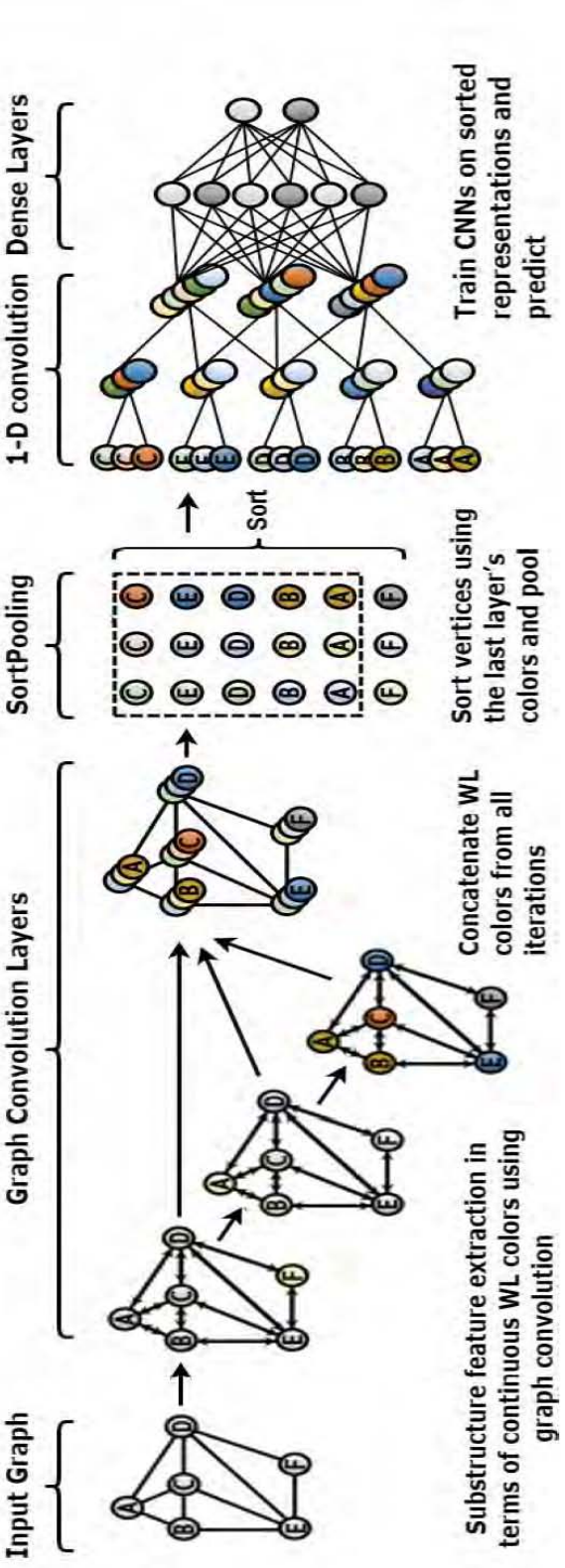



Figure 3.5: DGCNN architecture [Zhang et al., 2018]

a variant of stochastic gradient descent (SGD), to update the trainable weights of the neural network.

## 3.5 Optimum Threshold and Classification

We apply a sigmoid function on embedding to convert it to probability. After finding the target probability for each node, we find an optimum threshold to distinguish the misinformation spreader from the regular user. Keeping that optimum threshold, then we classify the user as a misinformation spreader or Regular user.

$$P = \{P_{U_1}, ...., P_{U_n}\}, label(U_i) = \begin{cases} 1, & \text{if } P_{U_i} >= threshold; \\ 0, & \text{otherwise} \end{cases} \tag{3.10}$$

Where P is a set of probabilities that are predicted by the model for Users while $label(U_i)$ represents a label for i-th user. If the probability is greater or equal to the threshold, then the model will classify as a Misinformation spreader; otherwise, it will be a Regular user.

## 3.6 Summary

In this chapter, we give a brief overview of the proposed method and technique used. The proposed method contains multiple stages of preprocessing, Neighbourhood sampling, and the model used to generate embeddings. These models are GCN, GraphSAGE, and DGCNN.

# Chapter 4

# Experiment and Results

This chapter presents our experimental results obtained through the implemented models. Several experiments have been done, and the results of these experiments were compared.

## 4.1 Platform and Programming Tools

The experiments were performed on a standalone system with OS Microsoft Windows 11 Pro with 64 GB RAM and 12 GB GPU. We used the Python programming language. Anaconda, a Python package manager, is used. We used several libraries during the experiment: Pandas, Numppy, Networkx, Torch, Torch Geometric, Sklearn, etc. For reading and prepossessing of data, we used pandas and numpy libraries; for graph embedding, We used torch and torch geometric libraries. We discussed the use of these libraries in Table 4.1

Table 4.1: Programming libraries used in experiments

| Name | Description |
|---|---|
| Pandas | Analyzing, cleaning, exploring, and manipulating data. |
| Numpy | Perform a wide variety of mathematical operations on arrays |
| Networkx | Graph creation and processing |
| Torch | Creating deep neural networks |
| PyTorch Geometric | deep learning on irregular structures, such as graphs, point clouds, and manifolds |

## 4.2 Dataset

In the experiments, we use the data set provided in the MediaEval 2022 Workshop[1]. There are different tasks in this workshop. We work on the FakeNews Detection dataset. A detailed description of this dataset is followed.

### 4.2.1 MediaEval 2022 FakeNews Detection dataset

The dataset was presented in the MediaEval workshop entitled "FakeNews Detection". It contains various tweets containing COVID-19 and other conspiracy theories with their propagation graph (undirected graphs) in which each node represents a user and edges represent some connection between them. The tweet texts are only in English and contain various long tweets with neutral, positive, negative, and sarcastic phrasing. The graphs also include a set of attributes (user profile attributes such as the number of followers and number of followings) for each node. The graph-based detection contains two classes: misinformation spreaders and Regular users; however, the data are not balanced with respect to the number of samples for each category. The items in the dataset were collected from Twitter between 20-January-2020 to 1-April-2022 using different keywords such as "corona", "COVID-19", related to the COVID-19 pandemic and related conspiracy theories. Then, researchers, postdocs, Ph. D.s, and master students assign labels to each tweet. Then, propagation graphs

[1] `https://multimediaeval.github.io/editions/2022/`

for each of the tweets are extracted from Twitter.

The dataset consists of three sub-tasks. The First Task, Text-Based Misinformation, and Conspiracies Detection is the tweet classification into nine categories based on tweet content. The second task, Graph-Based Conspiracy Source Detection, is to classify the user based on the propagation graph of tweets, their most likely retweet path replies, and quotes and quotes replies. It contains a single undirected graph G with a set of users/vertices and edges between users based on the above-explained criteria. The third sub-task, Graph, and Text-Based Conspiracy Detection, combined the content and propagation pattern. We work on the sub-task two. Our motivation is that the propagation pattern of regular user tweets and misinformation differs from their content, which depends on the specific domain.

Table 4.2: User Profile Features or Attributes

| feature Name | Description |
|---|---|
| Verified | Whether the Twitter account is verified or not |
| Desciption_length | The Length of the profile Bio/Description |
| Num_status | Number of Tweets Post by Users |
| Num_of_Follower | Number of users who followed that user |
| Num_of_Following | Number of users followed by that user |
| Date_creation | When the account was created |
| Location | User Country Location |
| Num_of_Friend | Number of users who followed that user and the user follow back those users |

As discussed above, the dataset consists of a single graph with vertices as a user and edge between them based on the relationship between retweet, reply, quotes, and reply routes. The dataset contains three files. The First file contains the edges between users, the second file contains the user profile features or attributes, and the third file contains the label of users (0 Regular User, 1 Misinformation Spreader). Table 4.2 shows the details of these features.

### 4.2.2 Dataset statistics

The provided graph consists of 1.6 million nodes and 268 million edges. There are 1913, and 830 labeled nodes in training and test set, respectively. Table 4.3 shows the ratio between labeled and unknown users. The amount of labeled users is very small, so we use an exponent of ten to show on the chart.

Table 4.3: Dataset Statistics

| statistics | Description |
|---|---|
| Graph | 1 |
| Number of Nodes | 1, 679, 011 |
| Number of Edges | 268, 694, 698 |
| Development set | 1722 (Unique) |
| Test set | 1822 (Unique) |
| misinformation spreaders (Development set) | 1194 |
| Regular User (Development set) | 528 |
| misinformation spreaders (Test set) | 448 |
| Regular User (Test set) | 334 |

## 4.3 Dataset Preparation

To train our proposed model, we first prepare the data. We split the development set into train-valid with the ratio 80:20. Due to unbalanced data, in the last experiment, we sampled the data to balance it. Table 4.4 shows the total labeled data and how we split it and then random over-sample the data.

## 4.4 Baseline Model

To evaluate our proposed model, we have compared the results obtained with traditional machine learning approaches and geometric deep learning. The baseline of machine learning models is Random Forest (RF), K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Naive Bayes (NB), and deep neural network (DNN)

Table 4.4: Data Preparation for Training the Model; MS = Misinformation spreaders, RU = Regular Users

| Development Data (Total: 1722) | |
|---|---|
| Misinformation Spreaders (MS) | Regular Users (RU) |
| 1146 | 576 |
| split(80:20) | |
| Train-Data | Valid-Data |
| Total: 1055 (MS=460 and RU= 941 ) | Total:667 (MS=116 and RU= 205 ) |
| After Random Oversampling | |
| Train-Data | Valid-Data |
| Total: 1833; (MS= 915 and RU= 918 ) | Total: 459; (MS= 215 and RU= 244 ) |

as shown in Table 4.8 to check weather how much the profile feature contributes in identifying misinformation spreaders. Whereas for the geometric deep learning approach, the baselines are GNN (not mention the specific model) [Maulana et al., 2022], GCN [Akbari, 2022], GraphSAGE [Bocconi et al., 2022], TAG-GCN [Korenčić et al., 2022], node2vec+MLP [Peskine et al., 2023] as shown in table 4.12 whose work on the same dataset and participated in the same workshop.

## 4.5 Experiment 1

In Experiment 1, we consider the problem as node classification and find the embedding on the node level. Based on these embeddings, we classify the nodes/users as regular users or misinformation spreaders. We use GCNConv as a message-passing mechanism to find the embedding of each node. We Use the three GCN layers combined with a single linear layer.

### 4.5.1 Model Configuration

To configure the model, we use stack three GCNConv layers. Within these layers, the ReLU activation function is used As shown in figure 4.1 and then passed to the embeddings to the linear layer, and the output length of the linear layer is passed

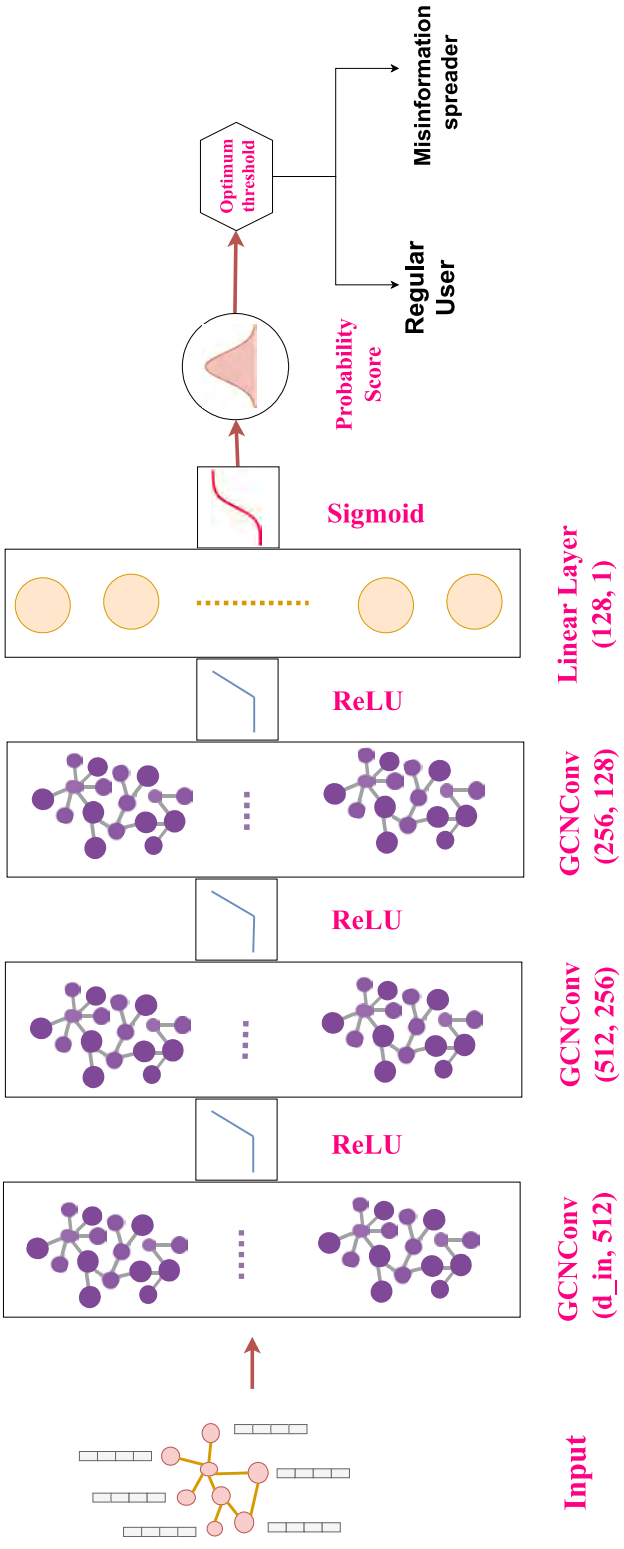



Figure 4.1: Implemented GCN model

through a sigmoid function to convert the vector into probabilities. As the problem is a binary classification, we use a Binary Cross Entropy [2] for the loss calculation. It calculates the loss between the input and the target probabilities. We use Adam [3] which is an algorithm commonly used in gradient-based optimization methods for training deep learning models. It stands for "Adaptive Moment Estimation" and combines ideas from two other popular optimization algorithms: Adaptive Gradient Algorithm (AdaGrad) and Root Mean Square Propagation (RMSProp). We kept the initial learning rate 0.00001, a weight_decay of 1e-5, and the value 0.3 as the dropout value. Table 4.5 shows the model configuration.

Table 4.5: Experiment 1: GCN model Configuration

| Model structure and configuration | | | |
|---|---|---|---|
| Layers | input units | Output Units | Activation Function |
| Input or Layer 1 (GCNConv) | 8 | 512 | ReLU |
| Layer 2 (GCNConv) | 512 | 256 | ReLU |
| Layer 3 (GCNConv) | 256 | 128 | ReLU |
| Linear Layer | 128 | 1 | Sigmoid |
| Hyper Parameter Tuning | | | |
| Loss Fucnction | | Binary Cross Entropy | |
| Optimizer | | Adam Optimizer | |
| Learning Rate | | 0.01 | |
| weight decay | | 1e-5 | |
| dropout | | 0.3 | |
| Epochs | | 100 | |
| callback | | ReduceLROnPlateau | |

## 4.6 Experiment 2

In Experiment 2, we stacked three GraphSAGE layers, the RELU activation function, and a linear layer. Figure 4.2 illustrates the model configuration and hidden dimensions. The motivation behind utilizing GraphSAGE is it is inductive and endeavors

[2] https://pytorch.org/docs/stable/generated/torch.nn.BCELoss.html
[3] https://pytorch.org/docs/stable/generated/torch.optim.Adam.html

to generate embeddings by using sampling and aggregation features from the node's local neighborhood.

### 4.6.1 Model Configuration

We construct the model using three GraphSAGE layers and a linear layer. Between these layers, the ReLU activation function is used. We used the sigmoid function to convert the vector at the last layer (Linear) to probability. After finding the probability of all the nodes, we find the optimum threshold using ROC-AUC. While keeping that threshold, we classify the user as a misinformation spreader or a regular user. We Run the model for 100 epochs. Table 4.6 shows the model construction and configuration.

Table 4.6: Experiment 2: GrapSAGE model Configuration

| Model structure and configuration | | | |
|---|---|---|---|
| Layers | input units | Output Units | Activation Function |
| Input or Layer 1 (SAGEConv) | 8 | 512 | ReLU |
| Layer 2 (SAGEConv) | 512 | 256 | ReLU |
| Layer 3 (SAGEConv) | 256 | 128 | ReLU |
| Linear Layer | 128 | 1 | Sigmoid |
| Hyper Parameter Tuning | | | |
| Loss Fucnction | | Binary Cross Entropy | |
| Optimizer | | Adam Optimizer | |
| Learning Rate | | 0.01 | |
| weight decay | | 1e-5 | |
| dropout | | 0.3 | |
| Epochs | | 100 | |
| callback | | ReduceLROnPlateau | |

## 4.7 Experiment 3

In the first two experiments, we consider the problem as a node classification. In this experiment, we convert the problem from node to graph classification. We generate

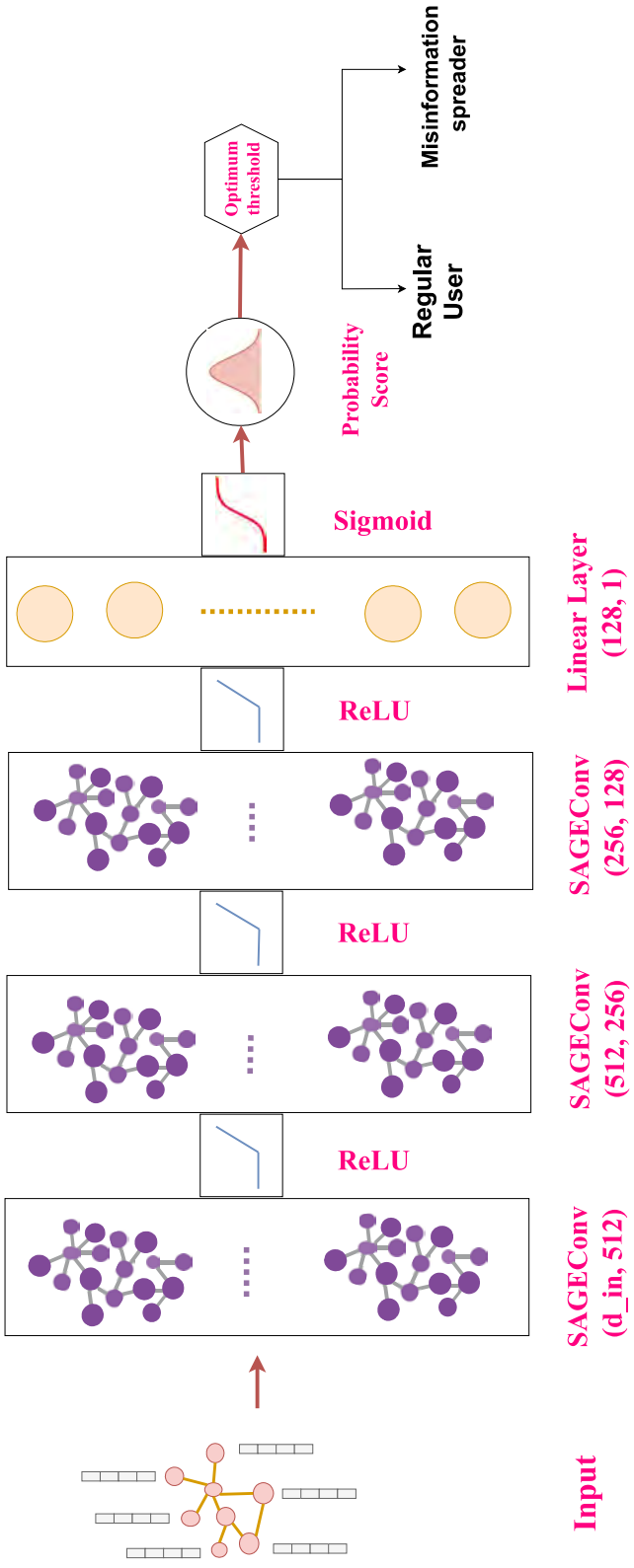



Figure 4.2: Implemented GraphSAGE model

an ego-centric graph against each label node. The subgraphs are constructed by taking the ego network of a node's up-to-3-hop neighborhood against each labeled node. The label is mapped between the node and their ego-centric graph. The label mapping between the node and the matching ego network of the node is formed in this manner. The model's objective is to classify the label of the node's ego network, which is the label of that node. Figure 4.3 represents the implemented model for graph classification.

### 4.7.1 Model Configuration

The model consists of four layers of GCN, which aggregates the feature on the node level. To aggregate the feature on the graph level, we use 1D-MaxPooling in between two 1DConv layers and then combine it with a fully connected layer. Then, we apply a sigmoid function to convert vectors into probability. After finding the probability of all the nodes, we find the optimum threshold using ROC-AUC. While keeping that threshold, we classify the user as a misinformation spreader or a regular user. We Run the model for 100 epochs.

## 4.8 Evaluation Metrics

Evaluation metrics are quantitative measures used to analyze a model's or algorithm's performance and efficiency in various tasks such as classification, regression, clustering, or recommendation systems. These metrics provide information about how well the model is doing and can help us make decisions about model selection, parameter tuning, and comparing multiple models. We utilize the following metrics to evaluate our model.

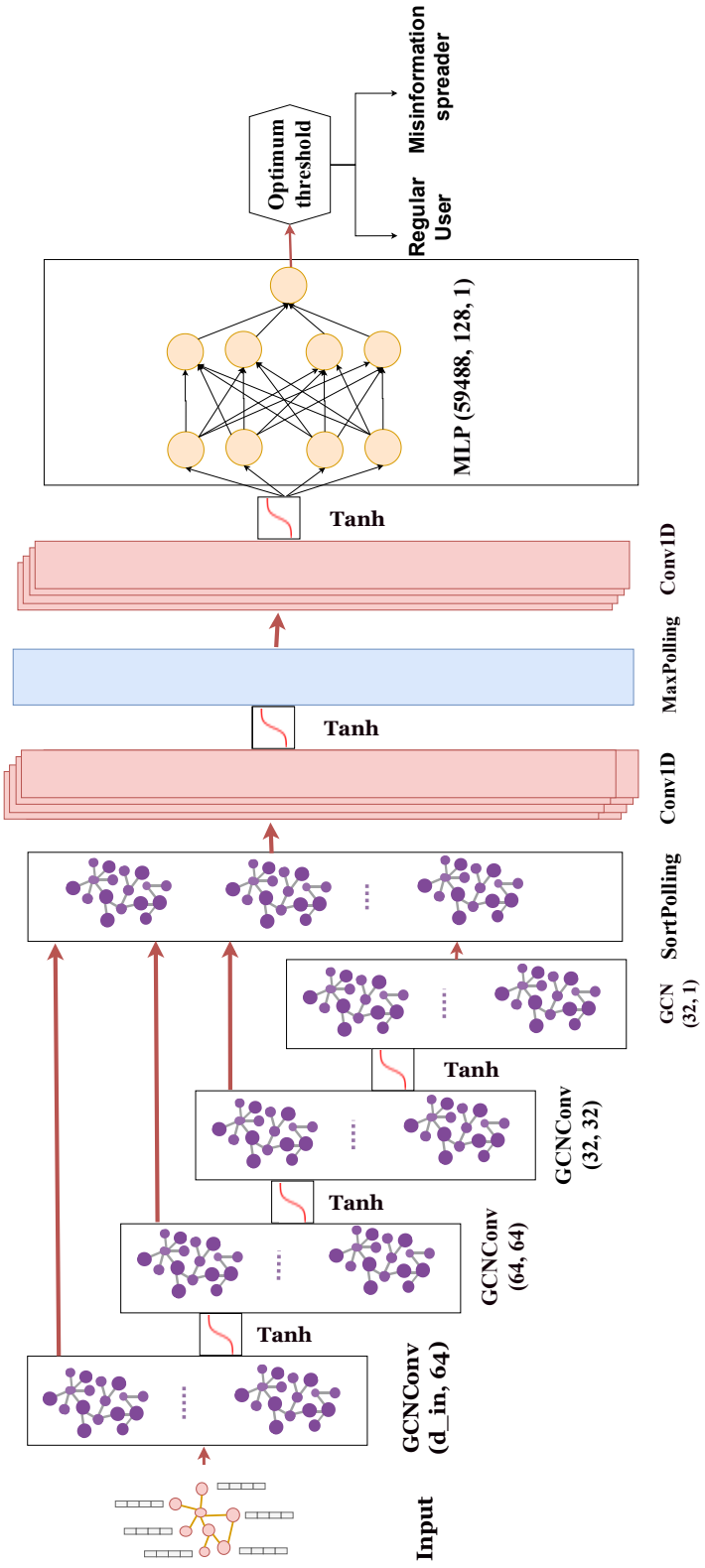



Figure 4.3: Implemented DGCNN model

Table 4.7: Experiment 3: DGCNN model Configuration

| Model | | | |
|---|---|---|---|
| Layers | input units | Output Units | Activation Function |
| Input or Layer 1 (GCN) | 8 | 32 | ReLU |
| Layer 2 (GCN) | 32 | 32 | ReLU |
| Layer 3 (GCN) | 32 | 32 | ReLU |
| Layer 4 (GCN) | 32 | 1 | No |
| Convolution and MLP layers | | | |
| 1D-Conv | (1, 16, kernel_size=(97,), stride=(97,)) | | |
| 1D-MaxPooling | (kernel_size=2, stride=2, padding=0, dilation=1) | | |
| 1D-Conv | (16, 32, kernel_size=(5,), stride=(1,)) | | |
| MLP | (780096, 128, 1) dropout= 0.5 | | |
| Hyper Parameter Tuning | | | |
| Loss Fucnction | | Binary Cross Entropy with logistic regression | |
| Optimizer | | Adam Optimizer | |
| Learning Rate | | 0.00001 | |
| Epochs | | 100 | |

### 4.8.1 Confusion Matrix

A confusion matrix is a way to determine how many predicted categories are correctly classified and how many are not. It is used to evaluate the results of a classification model. A few components of the confusion matrix that are

**True positive (TP):** It represents all objects that belong to a specific class. Our model also predicts them as a category they belong to.

**False Positive (FP):** It represents all those objects that do not belong to a specific class, let as A, but our model categorized them to class A.

**True Negative(TN):** It represents all those objects that are predicted as a negative class, and they are negative as well.

**False Negative (FN):** It represents all those objects that belong to a specific class. However, the model predicts them as another category.

Figure 4.4 shows the confusion matrix on the test set of our model. The test set contains 822 labeled items, of which 488 are misinformation spreaders, and 334 are Regular Users.

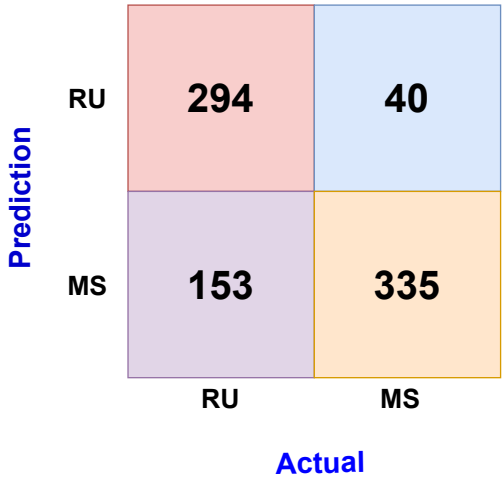



Figure 4.4: Confusion Matrix of our Model

### 4.8.2 Accuracy

Accuracy is one the most widely used evaluation measures, calculated as the ratio of the correctly classified to the total classes. It is a good measure when our datasets are symmetric, where false negative and false positive are approximately the same. In the case of the asymmetric dataset, we need to look at other evaluation measures.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{4.1}$$

### 4.8.3 ROC-AUC

The ROC curve / AUC score metric is useful when evaluating a model that gives us probabilities of positive or negative outcomes. It is beneficial when analyzing model performance at multiple classification thresholds. ROC curve (Receiver Operating Characteristic curve) is a plot between the true positive rate (typically on the x-axis) and the false positive rate (typically on the y-axis). AUC score (Area Under the ROC Curve) is an aggregated measure of model performance at all possible classification thresholds.

**True Positive Rate (TPR)** The true positive rate is a ratio between the number

of true positive outcome predictions and the number of all true positive instances.

$$TPR = \frac{TP}{TP + FN} \tag{4.2}$$

**False Positive Rate (FPR)** The false Positive Rate is a ratio between the number of false positive outcome predictions and the number of all negative outcome predictions.

$$FPR = \frac{FP}{FP + TN} \tag{4.3}$$

### 4.8.4 MCC

The Matthews Correlation Coefficient (MCC) is a measure used to evaluate the performance of binary classification models. MCC is calculated on the basis of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) values to accommodate an overall assessment of the model's predictive ability.

The formula for calculating MCC is as follows:

$$MCC = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP + FP) \times (TP + FN) \times (TN + FP) \times (TN + FN)}} \tag{4.4}$$

The MCC spans a scale from -1 to +1. A score of +1 signifies an accurate prediction, 0 represents a random prediction, and -1 indicates complete inconsistency between predictions and actual labels. MCC is considered a robust metric beneficial when dealing with imbalanced datasets.

## 4.9 Results and Discussion

We perform two types of experiments. The first one is to identify how much the profile feature of the user contributes to identifying misinformation spreaders. We use basic

machine learning approaches such as RF, KNN, SVM, NB, and DNN for that. Table 4.8 show the obtained results by these model. However, the proposed method shows the result by utilizing the propagation feature of the information. From the results, we conclude that the profile features can not contribute to identifying misinformation spreaders because of homogeneity in profile features. Therefore, we can not rely on user profile features to distinguish between misinformation and regular users and need to incorporate the propagation pattern of how information flows in the network by these users. In the upcoming paragraphs, we discuss the impact on the results by adding propagation features and outperforming the deep geometric baseline models.

Table 4.8: Model comparison with Traditional Machine Learning models

| Model | Test-Acc | Test-AUC-ROC | Test-MCC |
|---|---|---|---|
| Random Forest | 0.637 | 0.535 | 0.061 |
| K-Nearest Neighbors | 0.591 | 0.550 | 0.054 |
| Support Vector Machine | 0.66 | 0.50 | 0.050 |
| Naive Bayes | 0.593 | 0.618 | 0.000 |
| Deep Neural Network | 0.585 | 0.570 | 0.146 |
| **Proposed Method** | **0.7477** | **0.7838** | **0.5699** |

We incorporated the propagation features and used different state-of-the-art methods for graph structure data, such as GCN, GraphSAGE, and DGCNN. Table 4.9 shows the performance of the implemented model in terms of accuracy, MCC, and ROC-AUC. Experiments show that DGCNN performs well compared to the other two models, GCN and GrapgSAGE. From the experimental results, we come to the conclusion that the propagation features contribute to identifying misinformation spreaders.

Table 4.9: Result comparison of expirements

| Model | Accuracy-valid | MCC-valid | ROC-AUC-Valid | Test-MCC |
|---|---|---|---|---|
| GCN | 60.54 | 0.24 | 57.54 | 0.22 |
| Graph SAGE | 65.79 | 0.32 | 67.02 | 0.20 |
| **DGCNN** | **69.85** | **0.36** | **73.07** | **0.24** |

We compare the performance of our model with other baseline models that incorporate the propagation features utilized deep geometric learning on the same dataset and participated in the same workshop in Table 4.10. From the results shown in the table, the performance of our model is in the top three models.

Table 4.10: Result comparison with Other Methods

| Author | Model | Test-MCC |
|---|---|---|
| [Maulana et al., 2022] | GNN (not mentioned) | 0.0084 |
| [Akbari, 2022] | GCN | 0.041 |
| [Bocconi et al., 2022] | Graph SAGE | 0.1111 |
| **Our Approach** | **DGCNN** | **0.24** |
| [Korenčić et al., 2022] | TAG-GCN | 0.2831 |
| [Peskine et al., 2023] | node2vec+MLP | 0.35 |

The data was imbalanced. 67% of the data are regular users, and 33% are misinformation spreaders. To tackle the imbalanced problem, we use Random Oversampling and increase the misinformation spreader samples by randomly duplicating them. It is because when the data is imbalanced, we are calculating loss. Then, the loss calculation is biased to the larger sample. To avoid biases in loss calculation, we can assign weight to each class while training, and we can also use Random sampling. Table 4.11 shows the results on DGCNN after random oversampling. The experiment shows that the model performs well after applying Random Oversampling.

Table 4.11: Result After Random Oversampling

| Model | Accuracy-valid | MCC-valid | ROC-AUC-Valid | Test-MCC |
|---|---|---|---|---|
| **DGCNN + Random-Oversampling** | **0.7477** | **0.5593** | **0.7838** | **0.5699** |

After tackling the imbalanced problem, our proposed model outperforms the baseline models. Table 4.12 shows the comparison with other baseline models and the performance of our model placed in the top first position.

Table 4.12: Result comparison with Other Users

| Author | Model | Test-MCC |
|---|---|---|
| [Maulana et al., 2022] | GNN (not mentioned) | 0.0084 |
| [Akbari, 2022] | GCN | 0.041 |
| [Bocconi et al., 2022] | Graph SAGE | 0.1111 |
| [Korenčić et al., 2022] | TAG-GCN | 0.2831 |
| [Peskine et al., 2023] | node2vec+MLP | 0.35 |
| **Our Approach** | **DGCNN+Random Oversampling** | **0.5699** |

## 4.10 Summary

This chapter presents the details of the complete result and analysis of the performed experiments to evaluate the proposed method. The chapter is divided into sections, starting with platform and programming tools. After that, a detailed introduction of the dataset is given, followed by the preparation of the dataset. Finally, we explain different experiments performed (GCN, GraphSAGE, DGCCN) with evaluation metrics and compare the results with other proposed models. From the experiments, we concluded that up to 3 hop neighbors in social media improve the model's performance.

# Chapter 5

# Conclusions and Future Work

## 5.1 Conclusion

A massive amount of data is generated on Social Media. Therefore, checking the authenticity of each piece of information and who is sharing the information is a challenging problem. In this work, we proposed a practical approach for identifying misinformation spreaders. We utilized the propagation pattern of how the information is propagated in a network (twitter). The propagation pattern is generated by retweets, replies, quotes, and their replies. GNNs are beneficial for capturing the propagation pattern of misinformation and real information as they can learn a pattern in structured data such as graphs and outperform benchmark datasets. We employ GNNs such as GCN, GraphSAGE, and DGCNN models to capture propagation features and aggregate neighborhood information.

Different experiments showed that the proposed model achieved very good results from baseline models. The performance of the model in terms of MCC is 0.5669. Graph classification based on an Ego-centric network up to three hops outperform. Therefore, we conclude that the up-to-3-hop neighbor model performs well. The performance of the baseline and our models is considerably low because the feature

distribution of misinformation spreaders and regular users is approximately the same.

## 5.2 Future Work

In the future, we plan to approach the task with more sophisticated GNN models like Graph Attention Network (GAT), which assign high weight to important entities and low weight to less critical entities. A combination of Label Propagation (LPA) and GCN can help improve the performance of classifying misinformation spreaders or regular users. However, a model is required for early detection of misinformation to avoid catastrophic events.

The dataset we used is related to COVID-19 and related conspiracy theories. A diverse and colossal dataset with multiple domains such as political and other conspiracies and misinformation is required. Combining textual features, propagation structure, and profile features will help identify misinformation accurately.

# Bibliography

[Akbari, 2022] Akbari, R. (2022). Evaluating tf-idf and transformers-based models for detecting covid-19 related conspiracies. In *Working Notes Proceedings of the MediaEval 2022 Workshop, Bergen, Norway.*

[Alhindi et al., 2018] Alhindi, T., Petridis, S., and Muresan, S. (2018). Where is your evidence: Improving fact-checking by justification modeling. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 85–90.

[Amer et al., 2022] Amer, E., Kwak, K.-S., and El-Sappagh, S. (2022). Context-based fake news detection model relying on deep learning models. *Electronics*, 11(8):1255.

[Andrey Malakhov, 2020] Andrey Malakhov, Alessandro Patruno, S. B. (2020). Fake news classification with bert. In *MediaEval.*

[Biradar et al., 2023] Biradar, S., Saumya, S., and Chauhan, A. (2023). Combating the infodemic: COVID-19 induced fake news recognition in social media networks. *Complex Intell. Syst.*, 9(3):2879–2891.

[Bocconi et al., 2022] Bocconi, S., Patruno, A., and Malakhov, A. (2022). Transformers and gnns for fake news detection. In *Working Notes Proceedings of the MediaEval 2022 Workshop, Bergen, Norway.*

[Dai et al., 2020] Dai, E., Sun, Y., and Wang, S. (2020). Ginger cannot cure cancer: Battling fake health news with a comprehensive data repository. In *Proceedings of*

*the International AAAI Conference on Web and Social Media*, volume 14, pages 853–862.

[Duffy et al., 2020] Duffy, A., Tandoc, E., and Ling, R. (2020). Too good to be true, too good not to share: the social utility of fake news. *Information, Communication & Society*, 23(13):1965–1979.

[Guimarães et al., 2021] Guimarães, N., Figueira, A., and Torgo, L. T. (2021). Can fake news detection models maintain the performance through time? a longitudinal evaluation of twitter publications. *Mathematics*, 9(22).

[Guo et al., 2020] Guo, B., Ding, Y., Yao, L., Liang, Y., and Yu, Z. (2020). The future of false information detection on social media: New perspectives and trends. *ACM Computing Surveys (CSUR)*, 53(4):1–36.

[Hamilton et al., 2017] Hamilton, W., Ying, Z., and Leskovec, J. (2017). Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30.

[Hamilton, 2020] Hamilton, W. L. (2020). Graph representation learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 14(3):1–159.

[Hancock and Khoshgoftaar, 2020] Hancock, J. T. and Khoshgoftaar, T. M. (2020). Survey on categorical data for neural networks. *Journal of Big Data*, 7(1):1–41.

[Hathnapitiya et al., 2023] Hathnapitiya, S., Ahangama, S., and Adikari, S. (2023). Early detection of sinhala fake news in social media. In *2023 3rd International Conference on Advanced Research in Computing (ICARC)*, pages 130–135.

[Helmstetter and Paulheim, 2018] Helmstetter, S. and Paulheim, H. (2018). Weakly supervised learning for fake news detection on twitter. In *2018 IEEE/ACM*

*International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 274–277. IEEE.

[Horne and Adali, 2017] Horne, B. and Adali, S. (2017). This just in: Fake news packs a lot in title, uses simpler, repetitive content in text body, more similar to satire than real news. In *Proceedings of the international AAAI conference on web and social media*, volume 11, pages 759–766.

[Hoy and Koulouri, 2021] Hoy, N. and Koulouri, T. (2021). A systematic review on the detection of fake news articles.

[Jain and Kasbe, 2018] Jain, A. and Kasbe, A. (2018). Fake news detection. In *2018 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS)*, pages 1–5. IEEE.

[Kaliyar et al., 2021] Kaliyar, R. K., Goswami, A., and Narang, P. (2021). Fakebert: Fake news detection in social media with a bert-based deep learning approach. *Multimedia tools and applications*, 80(8):11765–11788.

[Khan et al., 2023] Khan, I., Sadad, A., Ali, G., ElAffendi, M., Khan, R., and Sadad, T. (2023). Npr-lbn: Next point of interest recommendation using large bipartite networks with edge and cloud computing. *J. Cloud Comput.*, 12(1).

[Kim and Tandoc Jr, 2022] Kim, H. K. and Tandoc Jr, E. C. (2022). Consequences of online misinformation on covid-19: two potential pathways and disparity by ehealth literacy. *Frontiers in psychology*, 13.

[Kingma and Ba, 2014] Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

[Kipf and Welling, 2016] Kipf, T. N. and Welling, M. (2016). Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

[Kochkina et al., 2018] Kochkina, E., Liakata, M., and Zubiaga, A. (2018). All-in-one: Multi-task learning for rumour verification. *arXiv preprint arXiv:1806.03713*.

[Kondamudi et al., 2023] Kondamudi, M. R., Sahoo, S. R., Chouhan, L., and Yadav, N. (2023). A comprehensive survey of fake news in social networks: Attributes, features, and detection approaches. *Journal of King Saud University - Computer and Information Sciences*, 35(6):101571.

[Korenčić et al., 2022] Korenčić, D., Grubišić, I., Sarracén, G. L. D. L. P., Toselli, A. H., Chulvi, B., and Rosso, P. (2022). Tackling covid-19 conspiracies on twitter using bert ensembles, gpt-3 augmentation, and graph nns. In *Working Notes Proceedings of the MediaEval 2022 Workshop, Bergen, Norway*.

[Langguth et al., 2023] Langguth, J., Schroeder, D. T., Filkuková, P., Brenner, S., Phillips, J., and Pogorelov, K. (2023). Coco: an annotated twitter dataset of covid-19 conspiracy theories. *Journal of Computational Social Science*, pages 1–42.

[Liu and Wu, 2018] Liu, Y. and Wu, Y.-F. (2018). Early detection of fake news on social media through propagation path classification with recurrent and convolutional networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

[Liu and Wu, 2020] Liu, Y. and Wu, Y.-F. B. (2020). Fned: A deep network for fake news early detection on social media. *ACM Trans. Inf. Syst.*, 38(3).

[Ma et al., 2017] Ma, J., Gao, W., and Wong, K.-F. (2017). Detect rumors in microblog posts using propagation structure via kernel learning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 708–717, Vancouver, Canada. Association for Computational Linguistics.

[Malhotra and Vishwakarma, 2020] Malhotra, B. and Vishwakarma, D. K. (2020). Classification of propagation path and tweets for rumor detection using graphical convolutional networks and transformer based encodings. In *2020 IEEE Sixth International Conference on Multimedia Big Data (BigMM)*, pages 183–190. IEEE.

[Matsumoto et al., 2021] Matsumoto, H., Yoshida, S., and Muneyasu, M. (2021). Propagation-based fake news detection using graph neural networks with transformer. In *2021 IEEE 10th Global Conference on Consumer Electronics (GCCE)*, pages 19–20.

[Maulana et al., 2022] Maulana, A., Pogorelov, K., Schroeder, D. T., and Langguth, J. (2022). Graph neural network for fake news detection and classification of unlabelled nodes at mediaeval 2022. In *Working Notes Proceedings of the MediaEval 2022 Workshop, Bergen, Norway*.

[Mehta et al., 2021] Mehta, D., Dwivedi, A., Patra, A., and Anand Kumar, M. (2021). A transformer-based architecture for fake news classification. *Social network analysis and mining*, 11:1–12.

[Mitra and Gilbert, 2015] Mitra, T. and Gilbert, E. (2015). Credbank: A large-scale social media corpus with associated credibility annotations. In *Proceedings of the international AAAI conference on web and social media*, volume 9, pages 258–267.

[Monti et al., 2019] Monti, F., Frasca, F., Eynard, D., Mannion, D., and Bronstein, M. M. (2019). Fake news detection on social media using geometric deep learning. *arXiv preprint arXiv:1902.06673*.

[Moosleitner et al., 2020] Moosleitner, M., Murauer, B., and Specht, G. (2020). Detecting conspiracy tweets using support vector machines. In *MediaEval*.

[Nguyen et al., 2020] Nguyen, V.-H., Sugiyama, K., Nakov, P., and Kan, M.-Y. (2020). Fang: Leveraging social context for fake news detection using graph repre-

sentation. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, CIKM '20, page 1165–1174, New York, NY, USA. Association for Computing Machinery.

[Orabi et al., 2020] Orabi, M., Mouheb, D., Al Aghbari, Z., and Kamel, I. (2020). Detection of bots in social media: A systematic review. *Information Processing & Management*, 57(4):102250.

[Peskine et al., 2023] Peskine, Y., Papotti, P., and Troncy, R. (2023). Detection of covid-19-related conpiracy theories in tweets using transformer-based models and node embedding techniques. In *MediaEval 2022, Multimedia Evaluation Workshop, 12-13 January 2023, Bergen, Norway*.

[Pogorelov et al., 2021] Pogorelov, K., Schroeder, D. T., Filkuková, P., Brenner, S., and Langguth, J. (2021). Wico text: a labeled dataset of conspiracy theory and 5g-corona misinformation tweets. In *Proceedings of the 2021 Workshop on Open Challenges in Online Social Networks*, pages 21–25.

[Raj and Meel, 2021] Raj, C. and Meel, P. (2021). Convnet frameworks for multimodal fake news detection.

[Raj and Mehta, 2020] Raj, C. and Mehta, M. (2020). Mediaeval 2020: An ensemble-based multimodal approach for coronavirus and 5g conspiracy tweet detection. In *MediaEval*.

[Rawat et al., 2023] Rawat, G., Pandey, T., Singh, T., Yadav, S., and Aggarwal, P. K. (2023). Fake news detection using machine learning. In *2023 International Conference on Artificial Intelligence and Smart Communication (AISC)*, pages 759–762.

[Raza and Ding, 2022] Raza, S. and Ding, C. (2022). Fake news detection based on news content and social contexts: a transformer-based approach. *International Journal of Data Science and Analytics*, 13(4):335–362.

[Reis et al., 2019] Reis, J. C., Correia, A., Murai, F., Veloso, A., and Benevenuto, F. (2019). Supervised learning for fake news detection. *IEEE Intelligent Systems*, 34(2):76–81.

[Rossi et al., 2022] Rossi, E., Kenlay, H., Gorinova, M. I., Chamberlain, B. P., Dong, X., and Bronstein, M. M. (2022). On the unreasonable effectiveness of feature propagation in learning on graphs with missing node features. In *Learning on Graphs Conference*, pages 11–1. PMLR.

[Saikia et al., 2022] Saikia, P., Gundale, K., Jain, A., Jadeja, D., Patel, H., and Roy, M. (2022). Modelling social context for fake news detection: A graph neural network based approach. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 01–08.

[Santia and Williams, 2018] Santia, G. and Williams, J. (2018). Buzzface: A news veracity dataset with facebook user commentary and egos. In *Proceedings of the international AAAI conference on web and social media*, volume 12, pages 531–540.

[Schaal and Phillips, 2020] Schaal, F. and Phillips, J. (2020). Using a word analysis method and gnns to classify misinformation related to 5g-conspiracy and the covid-19 pandemic. In *MediaEval*.

[Schroeder. et al., 2021] Schroeder., D. T., Schaal., F., Filkukova., P., Pogorelov., K., and Langguth., J. (2021). Wico graph: A labeled dataset of twitter subgraphs based on conspiracy theory and 5g-corona misinformation tweets. In *Proceedings of the 13th International Conference on Agents and Artificial Intelligence - Volume 2: ICAART*, pages 257–266. INSTICC, SciTePress.

[Shu et al., 2020a] Shu, K., Mahudeswaran, D., Wang, S., Lee, D., and Liu, H. (2020a). Fakenewsnet: A data repository with news content, social context, and

spatiotemporal information for studying fake news on social media. *Big data*, 8(3):171–188.

[Shu et al., 2020b] Shu, K., Mahudeswaran, D., Wang, S., and Liu, H. (2020b). Hierarchical propagation networks for fake news detection: Investigation and exploitation. In *Proceedings of the international AAAI conference on web and social media*, volume 14, pages 626–637.

[Shu et al., 2019] Shu, K., Wang, S., and Liu, H. (2019). Beyond news contents: The role of social context for fake news detection. In *Proceedings of the twelfth ACM international conference on web search and data mining*, pages 312–320.

[Silva et al., 2021] Silva, A., Han, Y., Luo, L., Karunasekera, S., and Leckie, C. (2021). Propagation2vec: Embedding partial propagation networks for explainable fake news early detection. *Information Processing and Management*, 58(5):102618.

[Singh et al., 2020] Singh, V. K., Ghosh, I., and Sonagara, D. (2020). Detecting fake news stories via multimodal analysis.

[Singhal et al., 2019] Singhal, S., Shah, R. R., Chakraborty, T., Kumaraguru, P., and Satoh, S. (2019). Spotfake: A multi-modal framework for fake news detection. In *2019 IEEE Fifth International Conference on Multimedia Big Data (BigMM)*, pages 39–47.

[Song et al., 2021] Song, C., Shu, K., and Wu, B. (2021). Temporally evolving graph neural network for fake news detection. *Information Processing and Management*, 58(6):102712.

[Sundararaj and Rejeesh, 2021] Sundararaj, V. and Rejeesh, M. R. (2021). A detailed behavioral analysis on consumer and customer changing behavior with respect to social networking sites. *Journal of Retailing and Consumer Services*, 58:102190.

[Szebeni et al., 2021] Szebeni, Z., Lönnqvist, J.-E., and Jasinskaja-Lahti, I. (2021). Social psychological predictors of belief in fake news in the run-up to the 2019 hungarian elections: the importance of conspiracy mentality supports the notion of ideological symmetry in fake news belief. *Frontiers in psychology*, 12:6255.

[Tuan and Minh, 2020] Tuan, N. M. D. and Minh, P. Q. N. (2020). Fakenews detection using pre-trained language models and graph convolutional networks. In *MediaEval*.

[Ullah et al., 2023] Ullah, A., Abbasi, R. A., Khattak, A. S., and Said, A. (2023). Identifying misinformation spreaders: A graph-based semi-supervised learning approach. *arXiv preprint arXiv:2303.03704*.

[Verma and Agrawal, 2022] Verma, P. K. and Agrawal, P. (2022). Propfnd: Propagation based fake news detection. In Unhelker, B., Pandey, H. M., and Raj, G., editors, *Applications of Artificial Intelligence and Machine Learning*, pages 557–568, Singapore. Springer Nature Singapore.

[Vo and Lee, 2018] Vo, N. and Lee, K. (2018). The rise of guardians: Fact-checking url recommendation to combat fake news. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pages 275–284.

[Vosoughi et al., 2017] Vosoughi, S., Mohsenvand, M. N., and Roy, D. (2017). Rumor gauge: Predicting the veracity of rumors on twitter. *ACM transactions on knowledge discovery from data (TKDD)*, 11(4):1–36.

[Vosoughi et al., 2018] Vosoughi, S., Roy, D., and Aral, S. (2018). The spread of true and false news online. *science*, 359(6380):1146–1151.

[Wang, 2017] Wang, W. Y. (2017). "liar, liar pants on fire": A new benchmark dataset for fake news detection. In *Proceedings of the 55th Annual Meeting of*

*the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 422–426, Vancouver, Canada. Association for Computational Linguistics.

[Wu and Hooi, 2023] Wu, J. and Hooi, B. (2023). Decor: Degree-corrected social graph refinement for fake news detection. *arXiv preprint arXiv:2307.00077*.

[Wu et al., 2019] Wu, L., Morstatter, F., Carley, K. M., and Liu, H. (2019). Misinformation in social media: definition, manipulation, and detection. *ACM SIGKDD explorations newsletter*, 21(2):80–90.

[Yang et al., 2023] Yang, Y., Zheng, L., Zhang, J., Cui, Q., Li, Z., and Yu, P. S. (2023). Ti-cnn: Convolutional neural networks for fake news detection.

[Zhang et al., 2018] Zhang, M., Cui, Z., Neumann, M., and Chen, Y. (2018). An end-to-end deep learning architecture for graph classification. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

[Zhao et al., 2014] Zhao, J., Cao, N., Wen, Z., Song, Y., Lin, Y.-R., and Collins, C. (2014). Fluxflow: Visual analysis of anomalous information spreading on social media. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):1773–1782.

[Zhao et al., 2020] Zhao, Z., Zhao, J., Sano, Y., Levy, O., Takayasu, H., Takayasu, M., Li, D., Wu, J., and Havlin, S. (2020). Fake news propagates differently from real news even at early stages of spreading. *EPJ data science*, 9(1):7.

[Zhou and Zafarani, 2020] Zhou, X. and Zafarani, R. (2020). A survey of fake news: Fundamental theories, detection methods, and opportunities. *ACM Computing Surveys (CSUR)*, 53(5):1–40.

[Álvaro Figueira and Oliveira, 2017] Álvaro Figueira and Oliveira, L. (2017). The current state of fake news: challenges and opportunities. *Procedia Computer Sci-*

*ence*, 121:817–825. CENTERIS 2017 - International Conference on ENTERprise Information Systems / ProjMAN 2017 - International Conference on Project MANagement / HCist 2017 - International Conference on Health and Social Care Information Systems and Technologies, CENTERIS/ProjMAN/HCist 2017.