MODELLING IRRATIONAL AGENT BELIEFS IN ONLINE SOCIAL NETWORKS

Chikadibia Franklin Chukwuma

A thesis submitted in partial fulfilment of the requirement of Staffordshire University for the degree of Doctor of Philosophy

February 2023

Dedication

This thesis is dedicated to my beloved parents, whose unwavering love and support have been my source of strength and inspiration. I am grateful for the sacrifices you have made for me and the endless encouragement you have given me throughout my journey.

To my esteemed supervisors – Dr. David White, Prof. Esther Maccallum-Stewart and Dr. Yvan Cartwright, who have guided me with their expertise and provided me with invaluable opportunities to learn and grow. Your patience, support, and encouragement have been instrumental in my academic journey.

And finally, to my fiancée, whose love and support have been my rock during the difficulties of my research. Your unwavering belief in me has given me the strength to overcome challenges and inspired me to do my best.

This thesis is a testament to the love, support, and guidance of those who have been with me every step of the way.

Abstract

The spread of misinformation through online social network platforms have become a major concern in society

Understanding human behaviour and decision-making in complex systems requires modelling irrational beliefs of actors in social networks. Irrational beliefs can drive people to make decisions that are counter to their own interests or the greater good, producing outcomes that are less than ideal for both the individual and society. This thesis addresses the problem of modelling irrational beliefs in social networks by creating a framework that reflects the impact of such beliefs on agent behaviour. Graph neural networks are increasingly employed to model how beliefs propagate across a network of interconnected agents and to explore how they affect outcomes in a social system.

This research presents a comprehensive review of the latest advancements in the use of graph neural networks for the purpose of modelling irrational agent beliefs in social networks. The approach represents agents and their interactions as nodes and edges in a graph. GNNs' are then used to learn the underlying structure and dynamics of the network, with a focus on understanding how irrational beliefs propagate through the network. The proposed framework incorporates the effects of social influence and biases into a GNN model of agent behaviour and is intended to provide insights into how misinformation and other forms of irrationality can spread within social networks and may have implications for understanding and mitigating the effects of disinformation and other forms of misinformation.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

List of Symbols

Notations	Description
$\{\ \}$	A collection of elements
G	A graph
V	The set of nodes in a graph
E	The set of edges in a graph
W	Weights on node edges
v	A node $v \in V$
e_{ij}	An edge $e_{ij} \in E$
n	The number of nodes, $n = \|V\|$
m	The number of edges, $m = \|E\|$
s	Super user – diffusion initiator
r	Mirror user
a	Informed user – unbiased private beliefs
f	Ignorant user – biased private beliefs
$s \subseteq V$	A set of super nodes
$r \subseteq V$	A set of mirror nodes
$a \subseteq V$	A set of informed nodes
$f \subseteq V$	A set of ignorant nodes

GLOSSARY

Term	Definition
Complex Adaptive System (CAS)	Complex System featuring dynamic network of interactions.
Feedback	Outputs form the network routed back as inputs.
Bias	Weight in favour or against an idea.
Ecology of Opinions	Collection of Differing Opinions.
Sub-Networks	Clusters of Smaller networks within a larger network.
Node level	Level at individual users in the network.
Network Externalities	Circumstances that affect a network.
Real World	Community of individuals (offline or online).
Actors	Participants in an action or process.
User State	Internal state of an agent – private beliefs.
Synthetic Network	Artificial (computer generated) Network.

List of Abbreviations

Abbreviation	Meaning
ABM	Agent-Based Modelling
ANN	Artificial Neural Network
CAS	Complex Adaptive Systems
GNN	Graph Neural Network
GCN	Graph Convolution Network
OSN	Online Social Network
NetT	Network Translation
DL	Deep Learning
RN	Real Network
AI	Artificial Intelligence
SNA	Social Network Analysis
SI	Susceptible Infected
SIS	Susceptible Infected Susceptible
SIR	Susceptible Infected Removed
SIRS	Susceptible Infected Removed Susceptible
SW	Small World network
GUI	Graphic User Interface
NLL	Negative Likelihood Loss

1. Introduction

Our daily lives involve networks like the Internet, the World Wide Web, electricity grids, and transportation systems. These networks are built as networks of interactions. Individuals can be represented as agents in social networks, and relationships or information flows between them can be represented as links (Stokman, 2001). These links can represent diverse types of relations between individuals. An agent with multiple links is an agent with many potential sources and reasons for distinct types of relations.

The activities of communities have tremendous influence across society due to the spread of information, ideas, and opinions (Namatame and Chen, 2016, p. 6). Diffusion is a daily social phenomenon of propagation within a society. From innovation, ideas, technology - to diseases, information, opinion and the latest trends, the study of diffusion examines how these spread through communities. Information diffusion, known as *diffusion of innovations*, is the "study of how information propagates in or between networks" (Rogers, 1995). The process of information diffusion can be compared to a virus spreading through human interaction.

Misinformation, disinformation, and irrational information (identified under the genre of false information) has been the focus of recent research in information diffusion. The spread of misinformation through online social network platforms have become a major concern in society (Luo, Cai and Cui, 2021). Their effects are particularly felt in social decohesion which they often result in (Zhang, Bogle and Wallis, 2021). Using agent-based modelling and simulation (ABM), this research aimed to understand the dynamics of adoption of beliefs at the micro-level within a social network, how it influences user behaviour and the dynamics at the macro-level in such networks.

1.1 Research Overview

Ever improving communication technologies combined with the growing number of social media platforms has rapidly penetrated into every aspect of society, and provided us with new channels and great ease to exchange information, express opinions and receive feedback (Kreindler and Young, 2014). These technologies help disseminate huge volumes of information over different platforms, spreading influence amongst each other. As these platforms become prevalent in human interaction and the diffusion of information, their influence on society has become more evident and significant.

A person's decision to adopt or reject a new belief is frequently influenced by the distribution of comparable decisions they witness among their peers, whether they be friends, co-workers, or acquaintances (Jackson and Yariv, 2006). This decision may also be driven by underlying network externalities - e.g., where such beliefs become more attractive as when more of an individual’s close acquaintances have adopted the beliefs. It could be seen as an artefact of simple learning processes, where the probability that an individual learns about a new belief increases over time as the number of its neighbours who have adopted the belief increases (Jackson and Yariv, 2006).

This research analyses diffusion of information and the resultant beliefs formed on social networks. In this research, a belief formation methodology is proposed for modelling the diffusion process with which information spreads in social networks over an underlying social network interaction topology, with varying user states, user state updates and different user types as some of the parameters. In addition to varying user state, user transitioning between states is made possible by using a message passing system proposed by Gilmer et al (2017) to exchange information about each user’s state between users, which allows the simulation models employing message passing to effectively replicate users' actions under influence of incoming information from neighbours and their effects. The proposed model also assists in predicting the final beliefs adopted of users in the network.

1.2 Research Context

Human behaviour contributes to the occurrence of events in the social environment across all facets of society. Our social environment significantly affects how we behave, think, and speak. (Namatame and Chen, 2016, p. 6). Social network ties form the foundation of interaction amongst individuals. Instead of occurring on a global scale, exchanges take place locally. A person's life is influenced by several things, including their family, tribe, neighbours, peers, friends, co-workers, education, and surroundings (Marsden and Friedkin, 1993). These social connections give people access to knowledge, concepts, and creativity, which influences the choices they make, the things they do, their successes, and the relationships they develop. Social networks form the foundation of these interactions, interactions that develop interdependently – they coevolve as these interactions are often dependent on each other (Namatame and Chen, 2016). Interactions play a leading role in shaping an individual’s private beliefs. The prevalence of Online Social Networks (OSNs) has given rise to the establishment of social links and interactions that are borderless (Chapdelaine and Manzerolle, 2021).

A key feature of modern OSNs is the deluge of false information (Murayama *et al.*, 2021). The interactions amongst individuals in social networks rise and fall making these networks constantly evolving realities, hence the need for a framework to better understand the diffusion processes. Evolving networks produce new emergent properties and ever-changing non-linear dynamic behaviours (Sasahara *et al.*, 2020). These behaviours are of particular importance in recent times with the proliferation of misinformation within networks. This can be seen in the misinformation on the coronavirus pandemic in OSNs - one example of such misinformation on the Twitter OSN platform is the tweet around the origins of the Covid-19 virus being from 5G cellular networks (Douglas, 2021).

The near constant deluge of misinformation has seen early detection of misinformation fast become one of the primary areas of focus of recent research in information diffusion (Zhang *et al.*, 2018; Monti *et al.*, 2019). One such method was proffered by Qian et al. (2018) who proposed a novel model featuring a two-level Convolutional Neural Network with User Response Generator (TCNN-URG). To train a generative model of user response to article text from previous user replies, their model represents semantic information from article text at the sentence and word levels. This allows for the early detection of fake news by capturing semantic information from article text. Table 1 shows some of the common terminologies found in social network research used in this research.

Terminology	Real-World Meaning
Agents	Single individual/user in a network
Clusters	Group of individuals/users in a network
Nodes	Users/agents in a network
Real-World Networks	Community of individuals (online or offline)
Edges	Connections between nodes
Seed users	Initial set of users in a network
Cascade Initiators	Individual users that start a cascade
Viral content	Rapidly shared piece of information

Table 1: Common Terminologies in Social Network Research

This research embraces private beliefs as having considerable influence in determining the links people establish within their social network and that this particularly holds true in OSNs which are not bounded by distance or rationality. The creation of links and relationships in social networks and how this influences information dispersion within such networks are both simulated using an Agent-Based Modelling (ABM) framework. User belief profiles are introduced as a feature of users in the simulation. These profiles are representative of the beliefs of the users within an online community and simulated across network types/topologies representative of such communities. The research establishes OSNs as dynamic networks not fixed in terms of location or geography and show their ability to adapt to changes in both network structure and in the states of users. This is of particular importance when taking into consideration the different regimes of diffusion for a piece of viral information.

1.2.1 What is Information?

The concept of Information is all-encompassing with different meanings in different contexts across all aspects of society (Capurro and Hjorland, 2003). Some definitions for information include - "Information is a measure of the uncertainty or surprise associated with a message" (Shannon, 1949), "Information in science is knowledge that has been derived from empirical observations, is falsifiable, and can be used to make predictions" (Popper, 1979) and "Information is well-formed, meaningful data that contributes to the knowledge and understanding of a system or observer" (Floridi, 2005), Information as defined as a word in English goes as far back to the late fourteenth century. Early usage of the word showed ambiguities that persist to the present times. "Information" describes everything from a precise mathematical property of communication systems to discrete statements of fact or opinion, to a staple of marketing rhetoric (Simpson and Weiner, 1989). Information is widely used by people across all walks of life to describe numerous general and specific aspects of life. This makes it difficult to analyse; there is no single academic discipline or method that can offer sufficient explanation of what information is in terms of the context of its use.

In an age widely regarded as the “Information Age”, defining information goes beyond its multiple meanings. Information can be posited as an ecosystem - Complex Adaptive Systems (CAS) in nature that exist alongside each other and interact with each other (Carmichael and Hadžikadić, 2019). Information as an ecosystem will include the information infrastructure, tools, media, producers, consumers, curators, and sharers (Sabzian, Maleki and Baghaei, 2019). Such ecosystems are “complex organisations of dynamic social relationships through which information moves and transforms while it diffuses”. As a CAS, the complexity of information is organised. In the information system, there are many agents with correlated interactions, and because of this correlation, these interactions can result in emergent, system-wide features - organised complexity. (Carmichael and Hadžikadić, 2019).

A common theme within social relationships in such ecosystems is feedback (Carmichael and Hadžikadić, 2019). Feedback (both endogenous and exogenous) exist between the agents in these clusters and between subnetwork clusters. A social distance between agents can emerge due to private beliefs and bias in such beliefs. This distance as well as changes in agent states after a belief threshold has been crossed are considered features that make information a CAS. The effects of information cannot be understood at its system-level properties and hence is best studied completely by summing up users and their interactions.

1.2.2 The role of Belief

The word Belief from the twelfth century is defined by Algeo (1989) as meaning "conviction of the truth of a proposition or alleged fact without knowledge"; it is also in terms of context "sometimes used to include the absolute conviction or certainty which accompanies knowledge". These convictions differ between individuals or groups of individuals – hence are often considered private beliefs. Private beliefs are often considered as a key factor that determines the social links people establish (Enders *et al.*, 2021). Bias plays a role in the formation and reinforcement of private beliefs.

In real life scenarios - scenarios in day-to-day human interactions (online and offline), when dealing with belief and bias, four conundrums exist with respect to information: what people want to hear, what people want to believe, everything else and then the truth (Ichikawa and Steup, 2014). These conundrums often form the basis of social relationships within society and are highly visible in the online social network space in which user identities can be masked and more extreme beliefs and opinions can be expressed. An overriding question that often arises during the study of private beliefs is – **"*what is the yardstick by which the degree of accuracy of a user's private beliefs can be measured*"?**

Users within an OSN tend to endorse claims that adhere to their system of beliefs (Cinelli *et al.*, 2021). These beliefs can be biased or unbiased. Within a group of people, there often exists an element of bias in interactions pertaining to information. The formation of bias is informed by our associations, neighbourhood, and private beliefs. In information diffusion within OSNs, confirmation bias and polarisation play key roles in the critical and supercritical regime of the diffusion process (the Viral Phenomena) (Li *et al.*, 2017). The near ubiquitous nature of

OSNs is an enabler for polarisation as subnetworks within OSNs are composed of like-minded agents.

The system of beliefs that informs a user's interaction often ensure group polarisation as such a user's interaction will often be limited to like-minded people (Modgil *et al.,* 2021). Users will only maintain interactions with other agents if the distance between their opinions is minimum (Moussaid *et al.,* 2013). By maintaining interactions with like-minded users, communities of users often emerge which are highly polarised and in which there is homogeneity in beliefs amongst the members of such communities. This homogeneity in beliefs often serves as the primary driver for the diffusion of content within such communities (Del Vicario *et al.*, 2017). Echo chambers are a feature of such communities, and they describe a situation where similar often fringe beliefs are amplified or reinforced by communication and repetition inside a closed community.

A feature of real-world networks are sub-network clusters which share the same ecology of beliefs/opinions. OSN sub-networks of different narratives (ecology of beliefs/Opinions), each of which might belong to a different OSN cluster, exists in real-world networks. Such sub-networks will be close in terms of interactions and will often affect each other (at the node level or sub-network level) via a feedback mechanism that allows them to react to changes within the network and in their surroundings (Carmichael and Hadžikadić, 2019). A Red Queen Effect - "the basic concept of which is to continually evolve, adapt and multiply in order to survive within competitive settings" is visible in such settings. In OSNs such an effect will see sub-networks of users in competition with other sub-networks of differing opinions with the aim of shaping the opinion/information narrative in the larger network (Miller and Page, 2019).

1.2.3 Modelling Information Diffusion in Online Social Networks

Diffusion when applied to OSNs can refer to the spread of information/ideas/opinions among interconnected users or entities in a network resulting from the interactions between these entities (Kumar and Sinha, 2021). In a real-life social network, participation in the diffusion process is dependent on several conditions. Some agents exist that autonomously decide to adopt or endorse an idea or piece of information without any external influence in the form of peer pressure from their friends and others that decide not to adopt those ideas (Milli *et al.*, 2018). Spontaneous adoptions by users, blocked users and poorly connected users does influence diffusion within such networks are common features of real-world networks. Such spontaneous adoptions can see certain users have substantial effects in the network - acting as diffusion anchors since their presence, or absence, deeply affect the unfolding of diffusive phenomena (Milli *et al.*, 2018).

In OSNs, hub/super agents often ensure Informational Influence – where the opinion and beliefs of an agent are influenced and shaped by their relationship and the actions of the surrounding agents. In a node-centric model, nodes with high levels of indegree – the number of incoming connections to a node, will often be key nodes and hence exert information influence. These nodes will often occupy key positions in the network and play a role in the diffusion within the network. In an OSN some users are more active than others hence playing a key role in diffusion within the network.

With OSNs being complex networks, complex contagion may be required in which multiple sources of exposure to information are required to initiate the diffusion process. Agent private beliefs and bias levels are complementary and are considered factors that contribute to complex contagion. Modern OSNs allow real-time interaction amongst users often in community clusters, and this can affect social transfer (Cowley *et al.*, 2008). These clusters are found replicated across the different social media platforms. The dynamics of OSNs make them highly susceptible to misinformation as users and the interactions among them rise and fall, hence the need for a framework to better understand the diffusion processes within such evolving networks (Namatame and Chen, 2016). Evolving networks produce new emergent properties and ever-changing non-linear dynamic behaviours.

1.3 Research Statement

A key feature of modern OSNs is the deluge of false information. In social networks, individuals, their beliefs, as well as interactions among them rise and fall hence they exist as constantly evolving realities. This places importance on the need for a framework to better understand the diffusion processes within such evolving networks. Such networks evolve while producing new emergent properties and ever-changing non-linear dynamic behaviours both at the user and network level. **The novelty of this research is a set of assumptions and definitions applied allowing for an end-to-end framework with the intrinsic flexibility to simulate social network interaction under differing belief systems amongst users in diverse social environments with a diverse set of outcomes.**

As a part of the research process and informed by the research background and context, the research problem is defined as:

For trending news in an OSN, a set of highly placed users diffuse misinformation for this trending news in the network whilst a second set diffuse true information. Other users in the network interact with the two sets of information being diffused within the network. All users in the network have a privately held belief which informs their interactions in the network and which information they eventually adopt.

Initially stating the problem statement in a general way with the presence of ambiguities establishes early-on the direction of the research questions. The feasibility of a solution must then be considered before a working formulation of the problem can be set up which then helps define clear research questions. The problem statement informs the three research questions asked:

1. ***How do private beliefs influence the diffusion of information within a social network?***

There is a strong correlation between the opinion of social media users, their beliefs, and their interactions (Arias, 2019). The prevalence of beliefs as the backbone of online social interactions and the ability to anonymise users behind these beliefs, means the public often face the risk of being misled when accessing relevant information as the line between true information and misinformation is not always easy to distinguish (Luo, Cai and Cui, 2021). People with an informational need can interact thanks to the design of social media platforms. While this makes it easier for people to communicate with one another and enhances the likelihood that the general public will have access to the accurate information they require, it also amplifies the detrimental effects of false information (Luo, Cai and Cui, 2021). Using simulations, this research aims to establish the effects of privately held beliefs on the establishment of diffusion links between agents and the roles these agents play in social networks.

2. ***How can instances of irrational information diffusion in OSN interactions be identified?***

The detection of irrational infromation in social networks is a challenging task. There are ways to automate the procedure utilising machine learning and Natural Language Processing (NLP) methods, with a focus on networks (Kumar and Geethakumari, 2014). In this research, simulations are used to establish the internal dynamics of user state interactions when two differing beliefs exist in a network and to check if these interactions inform the connections a user establishes in a network and their internal state.

3. ***How might Artificial Intelligence (AI) algorithms be employed to mitigate the spread of irrational information?***

Artificial Intelligence (AI) has been key to the growth in the reach of social media platforms (Cooper, 2022). Due to its unrestricted availability, which creates difficulties in determining the reliability of material, social media has developed into a significant means of communication and access to pertinent information for the general population. However, identifying and halting the spread of obviously bogus news is only part of the issue in combatting misinformation online (Cooper, 2022). AI algorithms now play a key role in moderating the content on social media platforms. Using simulation, this research aims to establish if AI classifiers can distinguish and classify different agent states at the individual level in a network.

1.4 Research Approach

Firstly, to give research context and to establish a research background, the thesis explores related works done in agent-based modelling, beliefs, information diffusion and Artificial Neural Networks (ANN). This helps establish the state-of-art and methods used in modelling social networks and diffusion as well as defining the areas of focus which align with the research, allowing for an effective project design. The context established helps identify tasks that need to be achieved to ensure successful research is a key requirement. The tasks identified inform the research method and larger methodology and have been well-defined to ensure that they can be accomplished within a specified period.

Secondly, following from the established context, the research aims, and objectives are formulated allowing research questions to be put forward within the background of the problem statement. The research operates under several established research questions and hypotheses to answer the research questions asked. The working hypotheses are tentative assumptions made that serve as the focal point of this research.

Thirdly, with specific objectives stated and research questions asked, the research approach is defined as being inductive research with quantitative methods such as estimation used for analysis. The models defined as part of the framework are implemented in the form of simulations using synthetic network datasets. The areas of interest that were explored to help inform the research method are highlighted in the subsequent sections below.

1.4.1 Background on Information Diffusion

The similarity between epidemic diffusion and information diffusion often sees both being studied using a common framework. However, significant differences exist between the two. Unlike epidemic diffusion, private beliefs, preferences, and bias are factors in information diffusion within a social network. Such diffusion within the social networks we establish does not happen simultaneously due to differing levels of agent suitability.

Information propagation through online social networks has become a facet of modern society. The three regimes (subcritical, critical, and supercritical) of the diffusion process can easily be observed for trending information in an OSN (Namatame and Chen, 2016, p. 152). With the rise of ubiquitous networks and the prevalence of OSNs, misinformation detection and containment remain severely lacking in many OSN platforms. Even in OSNs with some sort of detection, there is an existing time lapse between the detection and labelling of such information across the entire network. Real-time labelling would be a significant step towards broadcasting the truthful information and limiting false information.

The key task is to identify early-on the propagation of misinformation within an OSN cluster and shut down the nodes responsible. Most existing solutions for automatically detecting misinformation utilise machine learning algorithms that incorporate a variety of characteristics such as word count, in the OSN environment (Liu and Wu, 2018). However, these approaches ignore the internal dynamics of OSNs such as characteristics of users**. As a solution, this research addresses these shortcomings through a generalizable approach for the task of misinformation identification using a network's internal dynamics - key users, user interactions, possible early adopters, and propagation paths and to model irrational belief adoption in the specified OSN**.

In an OSN, diffusion within the network can be initiated by exogenous and/or endogenous actors (Koley *et al.*, 2021). Within an OSN sub-network, if the diffusion across all three regimes is initiated by an external node, the properties that allow for a successful diffusion are key to understanding the system. It is expected that actors (exogenous or endogenous) will be interested in feedback from the system and will in most cases react to this feedback. Users are also posited as having self-organisation which allows them to follow their own local rules informed by their private beliefs (Namatame and Chen, 2016, p. 9). These rules are applied based on the relations the agent establishes as well as its own attributes.

1.4.2 Background on Beliefs and Bias

Over a wide range of issues – social, political, and economic issues, differing opinions can be formed by individuals across society. These opinions are based on information obtained both from the established media and word-of-mouth in the form of their real-life social circle such as friends, co-workers, family members, etc (Fernandes, 2020). In the presence of no definite right/wrong or true/false distinction in information or when the information presented is ambiguous and not readily understood by individuals, the process of adopting information becomes heavily reliant on prior established private beliefs.

Bias, particularly confirmation bias, is a common feature of belief systems which according to the psychology literature - "connotes the perception of facts in ways that are consistent with current views" (Del Vicario *et al.*, 2017; Fernandes, 2020). Confirmation bias is often being employed by users in social networks. This means that someone has confirmatory bias if they frequently interpret ambiguous facts as supporting their pre-existing beliefs (Fernandes *et al.*, 2019). Several ways often lead to this which often centres around disregarding information/beliefs contrary to the existing world view and adopting instances that confirm existing bias.

Users in an OSN tend to support assertions that support their worldview and disregard evidence that contradicts it (Del Vicario *et al.*, 2017). They form links with other users with similar beliefs fostering the aggregation of like-minded people where debates tend to enforce group polarisation. Confirmation bias plays a pivotal role in viral phenomena in information diffusion within social networks as it reinforces user beliefs and strengthens existing connections. An Agent-Based Modelling (ABM) approach is used to present a framework for modelling irrational agent beliefs primarily applicable within online social networks.

1.4.3 Artificial Intelligence (AI) in Modelling Diffusion

Diffusion models seek to understand how information/innovation/idea spreads between and within individuals in a network as well as other networks. These models have attracted considerable research attention due to their widespread applications, in areas such as viral marketing, rumour control and technology/process adoption. Such models are often implemented using modelling and simulations which have intersections between both. Systems engineering, software engineering, and computer science all work together in the multidisciplinary discipline of modelling and simulation to create reliable construction methods for computerised models and create tools that can support people in all their endeavours (Zeigler, Muzy and Yilmaz, 2009).

AI is a field of computer science that involves the use of learning algorithms to learn new states of systems and make predictions. When applied to modelling and simulation, AI allows for verification and validation (Balci, 2003), model reuse and composability (Yilmaz and Ören, 2004), as well as distributed simulations (Dahmann, Kuhl and Weatherly, 2016). Using basic AI algorithms for learning which often involves one stage of learning suitable for analysing structured data; for predicting an outcome given a set of inputs; or for clustering items according to their characteristics is now a norm in models as part of data analysis (Strusani and Houngbonon, 2019). More recently, Deep learning AI algorithms are increasingly being used as part of modelling and simulations. They involve several learning stages and are organised like the structure of the brain (DengLi and YuDong, 2014). Such algorithms are suitable for analysing unstructured data such as images, audio recordings, or texts (Thota, 2018). Unlike basic learning AI algorithms, Deep Learning (DL) algorithms have opened new avenues for decision-making inferred from data, as few alternative methodologies exist to process unstructured data.

The growing application of simulation to study both artificial and natural information processes has revealed that the quality and complexity of simulation models will continue to change over the next few decades (Denning, 2007). This has introduced a new paradigm – the use of “Intelligent Agents” in simulation models (Logan, 2013). Using such agents, the idea is that it is possible for active entities in the environment being simulated to have their behaviours fully represented. These behaviours refer to the results of interactions between agents that are operationally autonomous.

1.4.4 The Agent-Based Modelling (ABM) Approach

The research framework can be implemented using an agent-based modelling strategy, which offers the potential to create a flexible model that can be generalised while still being useful in particular situations. In modelling information diffusion, one of the challenges is the state of the agents (nodes). Aymanns et al (2017) in their research on “Fake News in Social Networks” show that trained agents interact differently to untrained agents in diffusion models and that bias influences the links which these agents establish even with trained agents, thus influencing the diffusion process.

In the context of social networks, the game theory is applicable – where the social network structure is a part of the diffusion strategy. Namatame and Chen (2016, pp. 91-134), in their work on ABMs, regard "Agent-based modelling of social networks as the application of the games to a grand large society with various social constructs (social norms, reputations, social punishment, taxes, law)", and then to see how these games played in a decentralised fashion can constantly reshape the topologies of social networks, which may further cause the change of agents' strategies and behaviour. ABM makes it possible to comprehend social diffusion of information, which has a tipping point at which adoption by the population becomes self-sustaining and each additional adopting agent leads to one or more further adoptive agents, until the diffusion permeates a society. ABM is often used in concert with Social Network Analysis (SNA) when modelling social networks.

SNA are ideal for understanding the influences of user interaction in networks while AMBs allow for the exploration of feedback and interdependence between interactions amongst users, their outcomes, and effects on the network. Using SNA within agent-based models permits the simulation tools to reproduce reactions between the behaviour of heterogeneous agents and their surroundings and analyse/understand the effects of these behaviour. In this research, combining social networks and ABM will enable a focus on several important aspects needed to understand/simulate the behaviour of social interactions. Firstly, by having networks not just as a medium for the diffusion of information but also as a medium for social integration, it will be possible to show the relation between a user's network position and power in the network. Secondly, due to the co-evolutionary nature of OSNs, social networks and ABMs will allow for the research model to consider the feedback loop between user states and network topology. Users change their state according to their links in the network and these links according to their state.

Thirdly, with a user's internal state being a key operational assumption of this research, it will be possible to gain insights into relations between network connections and user features using local network metrics. Lastly, using an ABM approach enables network visualisation around the network structure and relationship allowing for a focus on each agent in the network as a unique entity not generalised entity. As a result of the focus on the micro-level in the network for example, the research will be able to establish connections between a user's position in the network and the degree of connectedness relative to users. This should give rise to the behaviour of the system and show its reaction to perturbations caused by false information both at the micro and macro level.

1.4.5 State of the Art in Artificial Neural Networks (ANNs)

Recent advances in ANNs have seen the introduction of Graph Neural Networks (GNNs). GNNs are connectionist models that use message passing between graph nodes to reflect the reliance of graphs (Zhou *et al.*, 2018). In a difference to standard neural networks, graph neural networks retain a state that can represent information from its neighbourhood with much more detail of the network environment. Graph data can be used to implement lots of learning tasks which contain rich relation information among elements. With GNNs, a graph representation of the problem is created and can be implemented as a graph network in one of its variants.

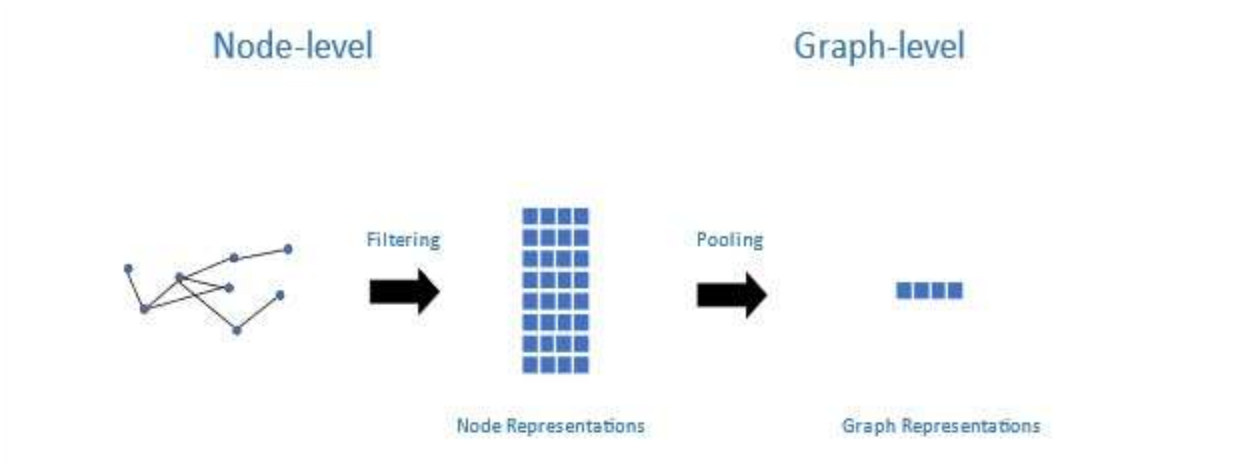

Figure 1. Graph Neural Network (GNN) Data
The figure shows the transformation from node-level to graph-level representations through filtering and pooling processes.

Figure 1 shows the internal workings of a typical GNN, at the node level - individual nodes have features which are filtered to create a representation of nodes in the network. At the graph level, the node representations generated can be used to create smaller graphs as a particular label through pooling whilst preserving its structure and the attributes of nodes within the graph.

With ever-increasing advances in computing power, optimisation techniques, and network architectures, GNNs have become adept at representation learning. Recently, ground-breaking performance on many learning tasks have been demonstrated by systems as variations that extend graph neural networks such as graph convolutional network (GCN), graph attention network (GAT) and gated graph neural network (GGNN). This has seen application in social networks with recent research focused on representation learning within such networks using Graph Neural Networks. One line of studies focuses on node-level representation learning – node embeddings and how this can help in node-link predictions, node classification and graph-level classification (Hamilton, Ying and Leskovec, 2017; Kipf and Welling, 2017). Li et al (2015) in their work, proposed an end-to-end predictor with the purpose of inferring cascade size using Recurrent Neural Network (RNN) and representation learning. A second line of studies focuses on graph representation learning (Ying *et al.*, 2018; Gao and Ji, 2019; Bianchi, Grattarola and Alippi, 2020) with the purposes of classifying entire graphs under a single label.

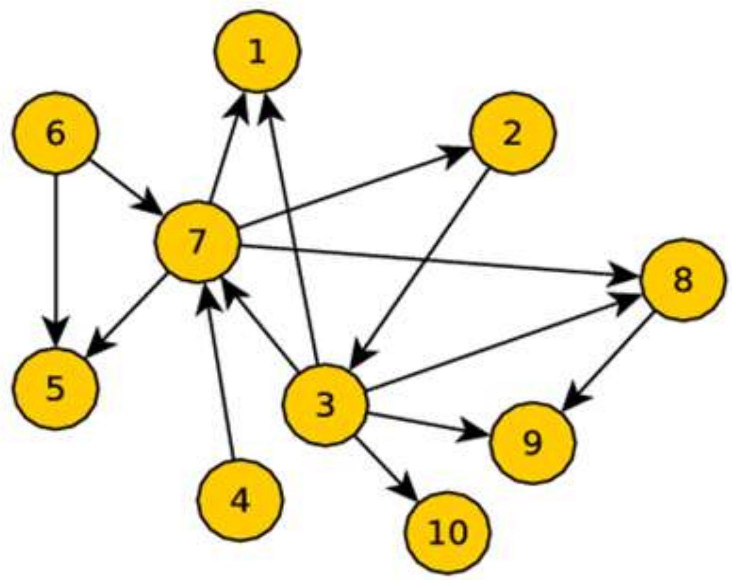

Figure 2. Graph representing a Social Network
The image shows a directed graph depicting interconnected nodes labelled from 1 to 10, where arrows represent the flow or relationship direction between them.

Figure 2 shows a social network represented as a graph network. This research proposes a GNN framework for modelling agent beliefs and their effects on information diffusion within social networks. A variant of the Graph Neural Network (GNN) framework is introduced as part of one of the models using several recent advances in GNNs. By converting social networks into graphs, the aim is to learn the representation of these networks in terms of belief adoptions and diffusion within the networks and the information class type. This should allow for the user states in the network to be predicted and classified accurately.

1.5 Research Aims and Objectives

This research aims to model irrational agent beliefs in online social networks to understand the factors that influence the diffusion and adoption of irrational information. The objectives are to:

1.) **Review and critically analyse the available literature on OSNs, Agent-Based Modelling (ABM), Artificial Neural Networks (ANN) and Artificial Intelligence (AI).**
2.) **Identify key users.**
3.) **Investigate the role of beliefs in the links that users establish in social networks, the state of users and its role in the diffusion process.**
4.) **Develop a method to accurately detect and classify misinformation diffusion within a network.**
5.) **Validate the model by applying the model to real-life applications.**

1.6 Research Motivation

OSNs have transformed communication, influencing cultural trends and public opinion. The spread of irrational information, which frequently take the shape of false information or conspiracy theories, is a serious concern. Understanding how these views spread is crucial given the prevalence of such content on OSNs. Irrational information like those concerning vaccine reluctance, left unchecked can have practical repercussions, especially in fields like public health. Online platforms mix emotional and rational interactions; therefore, belief diffusion and adoption simulation models must take this complexity into account.

Understanding these dynamics can aid with the development of efficient tools to combat irrational information. This can also help to improve digital literacy initiatives by ensuring that people can tell the difference between rational information and irrational information. This research aims to develop a practical framework for modelling belief diffusion and adoption in OSNs.

1.7 Research Framework

In developing the research framework for modelling irrational agent beliefs, several challenges were identified. These are further discussed below.

1.) **What solution can be offered by this research?**

Existing research already carried out had several models and frameworks proffered as solutions to modelling information diffusion and belief adoption. This research proposes a solution that adopts the state of the art in modelling - GNNs, in which graphs are used to represent social networks with the nodes in graphs as users. Using GNNs node structure, features/characteristics are assigned to individual users from which the interactions between users and the emergent behaviours can be captured (Fan *et al.*, 2019).

2.) Model Framework

The framework adopted explains the path of the research done so far and provides truth by grounding it firmly in the theoretical constructs created. The goal is to ensure generalizability and adaptability while making the research findings more relevant and valid to research field conceptions (Davidavičienė, 2018). To ensure a conclusive result, the research framework addresses the research objectives and questions asked. The aim is to develop a framework that supports a model which:

- **Is highly representative of the features of real-world OSNs in terms of misinformation diffusion and their adoptions within the network.**
- **Can abstract its complexity making it as simple as possible to implement while generating valuable data that can be efficiently analysed and presented.**

The nature of the research explores the relationship between users in a network, their private beliefs and information diffused within the network. This is done by positing definitions of terms including the variables relevant to the research and exploring how they may be related. The framework design assists in stimulating research, provides a contribution to knowledge while providing both direction and impetus to the research questions. The framework also serves to reinforce the hypotheses posited in the research.

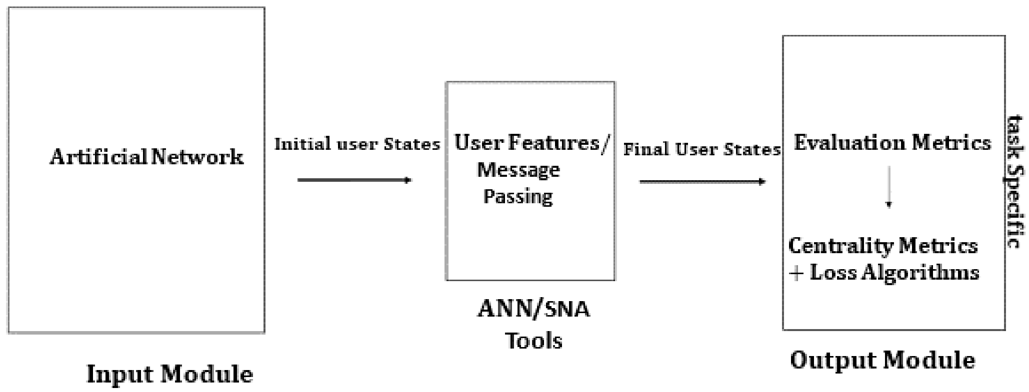

Figure 3. Model Framework Overview
The image illustrates the progression from input module with an artificial network to output module using ANN/SNA tools, encompassing user state transitions and evaluation metrics.

Figure 3 shows the model framework contains the structure/system in a summarised detail that constitutes the model for the realisation of a defined project result. **The model - "Network Translations" consists of three versions - versions one (NetTv1), two (NetTv2) and three (NetTv3) each having three modules and implemented as simulations. Each of the model versions are an extension over the preceding model - both in functionality and components** as seen in Figure 4.

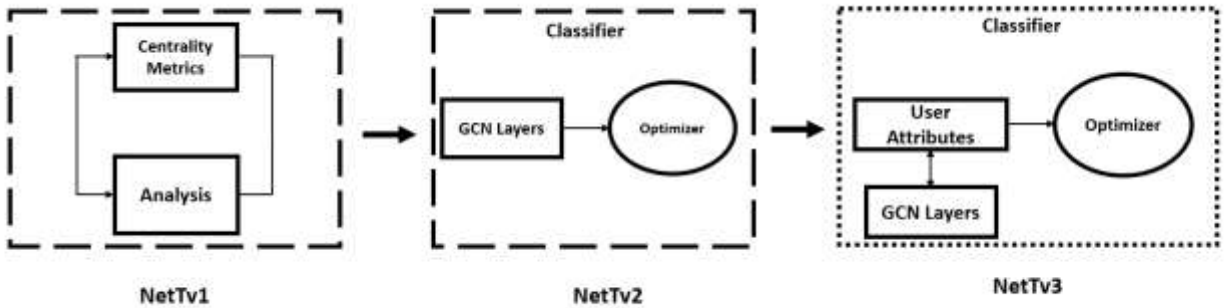

Figure 4. Model Iterations
The figure shows a schematic representation of three network models: NetTv1 focuses on centrality metrics and analysis; NetTv2 employs GCN layers leading to an optimizer and NetTv3 integrates user attributes with GCN layers, which then interact with an optimizer.

3.) Model Simulation and Validation

The research models are implemented as simulations using various tools with the implementation focused on two factors - Model task and Model requirements. Each model addresses a specific task aligned with the research questions and hypotheses. To ease simulation, the model and its components are simplified where possible. The hardware used in implementing the model is common across all three models. A flexible approach is taken towards the simulation software solution as both open-source software and a purpose-built software package are used.

The first model – NetTv1 uses an open-source software for the software implementation with the model tasked with establishing the internal structure of an OSN. The second – NetTv2 and third – NetTv3 are implemented as GNNs, adopting the graph-based approach. All models use synthetic networks dataset for primary implementation and real-world network datasets to validate the results from the synthetic networks. Analysis of primary data collected is done using several evaluation metrics and loss algorithms.

4.) Definitions and Notations

As a part of the framework design architecture, several definitions are introduced that are pertinent to the framework context and are applicable to all model versions in the research. These definitions can be extended as part of unique definitions for the various model iterations during implementation. Several terms and concepts relevant to the framework architecture are summarised in Table 2.

Terms	Definitions
Social network	Structure of interconnected users represented as nodes (users) and links (connections)
Edge	Connection between two users.
Information Diffusion	The spread of information within a network.
User Beliefs	Private beliefs that inform user interactions.
Super Users	User with the largest number of inbound connections from other users.
Mirror Users	Early adopters of information in a social network.
Normal User - Informed Users	Users with an unbiased private belief.
Normal User - Ignorant Users	Users with a biased private belief
Irrational Information	Information lacking factual accuracy.
Weight	Strength of connections between users.

Table 2. Definitions relevant to the Research Models

- **Social network**

A social network is defined as a directed graph $G = (V, E)$ where V represents users called nodes and E called edges represents the links/connections between users. A set of users is denoted by $V = \{ v_1, \ldots\ldots, v_n\}$, where n is the number of users in the network. $E = \{ e_1, \ldots \ldots$, $e_{m\}}$ denotes a set of edges between nodes in the network, where m is the number of edges. Weights W on edges represent the strength of the interaction between users.

The synthetic (artificial) network used in the simulation is created as a scale-free network whose degree distribution follows a power law and uses a preferential attachment mechanism for the establishment of links.

- **Edge**

An edge is defined as a connection between two users in the network. An edge has two sets of edges, (i) in-bound neighbours and (ii) outbound neighbours.

- **Information Diffusion**

The structure of information spread in the social network. Using cascade dynamics provides a precise modelling of information spreading. Users who publish information that trigger the information diffusion process are tagged Cascade initiators.

▪ **Super Users**

Super Users s are defined as seed users who publish information that initiate an information cascade. Supers function as hub agents which are high connectivity agents who initiate the diffusion process and take up a central position in the network. They initiate the information diffusion in the social network. Super nodes within the simulation have no "out-degree" edges and are assumed to have two bias states - positive or negative.

▪ **Mirror Users**

Mirror users r are users who serve to enable the views held by Supers (s) through retweeting the piece of information that they publish. Mirrors can have a unidirectional or bi-directional relationship with super users while having a bi-directional relationship with other mirror users. They also take up key positions within the network. They are early adopters of information in the social network. Mirror nodes in the synthetic network within the model simulation have no more than 50 in-degree node connections and are also connected to the super nodes. They can be positively or negatively biased.

▪ **Normal User - Informed Users**

One of two subgroups of the independent user class. Informed users a are users whose private information about the state of the world is largely correct on average. These users on average, relying only on the state of their private information, should be able to correctly identify such misinformation often - hence being unbiased. Such users are assumed not to update their beliefs based on one sub-network (OSN cluster). Informed agents have a uni-directional with s/r users and bi-directional relationship with other a/f users.

▪ **Normal User - Ignorant Users**

The second subgroup of the independent user class, ignorant users f are users whose private information about the world is biased and so do not check the veracity of the information posted and take it face value. Users classed as f are posited as generally displaying a non-progressive diffusion in their states as part of their interaction within an OSN cluster. f users have a uni-directional with s/r and bi-directional relationship with other f/a users.

▪ **Irrational Information**

Information that is inaccurate or deceptive information which can be deliberate or not. Within this research, false information is classified into 3 classes. *c* denotes the information type. Where for *c* {1 = Satire, 2 = Impostor, 3 = irrational information}.

- **Weight**

The weight w represent the level of interaction between the users for a given post. $W_{s,a,f,r,c}$ represents the level of interaction between users r, a, f and user s with respect to a piece of false information *c*. N represents the number of nodes (users/agents) in the social network. Users r, a, f follows user s. For each interaction (e.g., between f and s) is a set of weights $w_{f,s}$ that represents the strength between users s and f. f which is originally inactive has a uniform random threshold [0,1] as the probability of being influenced and becomes activate by s if:

$$\sum_{s \in Nbor\, f} w_{s,f} = \theta_f \tag{1}$$

were $Nbor\ f$ denotes the set of active neighbours of the target node f.

- **User Beliefs**

User beliefs are defined as the private beliefs of users in the network. Private beliefs are implemented as belief profiles which are created for users in the simulation network. These profiles are posited as node features and are analogous to labels of such nodes. These are created as numerical types of a vector data type. User connections (edges) can also have features assigned to them.

1.8 Research Contribution to Knowledge

The series of experiments conducted, spanning from NetTv1 to NetTv3 advanced the understanding of social network dynamics, user classifications, and information diffusion.

1.) **Novel User Classification:**

The simulations presented a structured evolution in the classification of users within networks. Building upon existing literature, the models innovatively classify users into “super users,” “mirror users,” and “independent users,” detailing the nuanced roles, characteristics, and positions of each user type within the network. This multi-layered understanding is vital to finding out the fundamental principles of information dissemination as it ultimately yields knowledge that can be used to predict and manage information flow in real-world situations.

2.) Role of Network Dynamics

The simulations highlight the intricacies of network dynamics and its notable impact on the diffusion processes. The pivotal role of users' structural positions, their ego neighbourhoods, and their overarching influence on the overall network state, including the efficiency of public signals within it, has been analysed. It has been ascertained that user states and their structural positions aren't monolithic in their influence on diffusion, suggesting a need for targeted interventions and strategies based on user roles in the network.

3.) Ego Networks and Heterogeneous User Exploration

The research models – NetTv2 and NetTv3 delves deep into the understanding of ego networks and their influence on information adoption and diffusion. Recognising that consensus beliefs and mass adoption occur predominantly when neighbouring agents share similar profiles and belief systems, the studies emphasise the need for a targeted approach to disrupting biased opinion and information spread in heterogeneous networks.

4.) Belief Systems

When considered from the view of OSNs in the presence of belief systems, the links users establish can be described being built on an associative basis: two statements put out by two users have similar belief contents hence are endorsed by each other. This reinforces the view that private beliefs determine the social links people establish. NetTv3 established a correlation between links users in the network establish and their belief profiles. The results showed that adoption within a network is eased when neighbouring agents share similar profiles and belief systems.

1.9 Thesis Outline and Summary

This chapter introduced the concepts behind this thesis and an overview of the research questions that are answered with respect to modelling irrational agent beliefs. The chapters consisting of the rest of this thesis are organised as follows:

▪ **Chapter 2 details the research background and related works.**

This chapter reviews existing literature (Chapter 2) allowing for an understanding of the state of the art in information diffusion and belief modelling.

▪ **Chapter 3 presents the project Methodology and Framework for Modelling Irrational agent beliefs in online social networks.**

This chapter discusses the proposed model framework, overall picture of the possible courses of action required to bring to realisation the research aims and objectives.

▪ **Chapter 4 presents the first model iterations (implementation, testing and evaluation) as well as its contribution to knowledge.**

NetTv1 is reviewed, model definitions and assumptions made are described in detail. The model is presented as an extension of the existing LT and IC model and details several practical applications including the model's contribution to knowledge.

▪ **Chapter 5 presents the second initial model iterations (implementation, testing and evaluation) and its contribution to knowledge.**

NetTv2 is reviewed in Chapter 5, with model definitions and the assumptions made described in detail. The second model is presented as an extension of the initial model and practical applications are presented. The contribution to knowledge of the models is detailed. The model is implemented as a synthetic network simulation. The model is evaluated using a real-world dataset.

▪ **Chapter 6 consists of the core model of the thesis, the implementation, evaluation, and data analysis.**

NetTv3 model put forward in the thesis is discussed in Chapter 6. The model exists as an extension of the earlier second model. The model definitions, implementation as a synthetic network and contributions to knowledge are detailed. This model introduces more distinct profile features, and more users are initialised in line with the user classes established within the project. The model is validated using a real-world network dataset with results from the simulation compared and evaluated against those of the synthetic network.

▪ **Chapter 7 provides concluding remarks and identifies future directions.**

A summary of the research will be undertaken along with their contribution to answering the research question and proving the hypotheses. An examination will be made of limitations of the research and potential future research directions are identified and a brief description is given on the several such areas. The contributions to knowledge of this research are also presented and detailed as well as future works building on the existing research done.

2. Literature Review

2.1 Introduction

Information diffusion has been the subject of much research, with the majority of studies focusing on the factors that influence it, the types of information that spread quickly, and the dissemination of information itself (Acemoglu, Ozdaglar and Parandehgheibi, 2010; Pei *et al.*, 2014; Arnaboldi *et al.*, 2017; Allcott, Gentzkow and Yu, 2019). Information diffusion models and other techniques are used to provide answers to these issues, which are crucial for comprehending the dissemination process. Previous research within the context of social networks has established that users in social networks tend to endorse information/ideas that adhere to their world view and ignore dissenting information (Del Vicario *et al.*, 2017; Li *et al.*, 2017; Fu *et al.*, 2019; Cinelli *et al.*, 2021). Users also form links with other users who have similar world views fostering the aggregation of like-minded people where debates tend to enforce group polarisation.

In Chapter 1, this thesis introduced the research giving a brief overview of the context and providing background on the research. The research aims and objectives were discussed as well as the challenges, research methodologies and research approach. Chapter 2 critically analyses literature relevant to this research. Firstly, a review is carried out on related works done in the fields of information diffusion in social networks, misinformation identification and belief adoption, broadening understanding of the existing knowledge on the topic and providing veracity to our research questions and methodology. Secondly, this chapter reviews existing ABM approaches, tools and software that have been used in simulations in previous works and provides details on the approach, tools and software used in the model simulations of this research.

2.2 Research Topic Review

Firstly, a review early models for information flow within a social network is done. Secondly, a review of the diffusion of information/ideas within social networks is carried out. Thirdly, the problems associated with social networks and information diffusion within such networks are identified. Fourthly, a review of previous solutions proffered in modelling contagion within such networks in terms of false information and belief is done. The context of the research in terms of the objectives and approach is also contrasted with existing works.

2.3 Information Flow in Social Networks

Early studies on information flow in social networks was done by Mark Granovetter. In his papers “The Strength of Weak Ties” (Granovetter, 1973) and “The strength of weak ties: a network theory revisited” (Granovetter, 1983), Granovetter looked at information flow in a small-world social network. According to Granovetter, in a small world social network, information sharing mostly occurs through loose links between members, or so-called weak ties, whereas strong connections, or "friendship-style" ties, oversee decision-making, knowledge creation, and knowledge preservation (Zinoviev, 2015). The theory of weak ties and strong ties has been extended in other works and has served as a foundation for contemporary information diffusion theories.

An effect of the theory of weak and strong ties is that social networks and individual ties within such networks develop interdependently – they coevolve. This often results in opinions and behaviours over a wide range of issues coevolving according to the social network and the individual characteristics possessed by the users within the network. These networks are dependent on the success of social relationships which shape the relationship between individuals in the network. The social environment of an individual influences much of what we do, and our behaviour contributes to the social environment of which we are a part of – creating Social Interdependence (Ball, 2004). Over time individuals come to closely resemble the peers to which they are connected to within their social networks.

One area of research models the propagation of an epidemic as the information flow through a sizable social network (Castellano, 2009). OSNs, now the modern prevalent form of social networks has relegated real-world social network communities to the background. Most individuals today engage with one another through online social media platforms, and information frequently spreads virally in these networks. Social networks are essential for the spread of knowledge, ideas, opinions, suggestions, and new goods. When people learn about new information, concepts, or products, they inform their friends, co-workers, and other people in their networks. Information diffusion within these networks is related to time, the strength of the relationship and levels of interaction between users, information content, social factors and network structures as rightly stated in (Liu and Zhang, 2014).

2.4 Information Diffusion in Social Networks

Much research has been done in terms of modelling the process around the diffusion of information/ideas/opinions. Most current efforts on diffusion modelling can be divided into two groups: graph-based approaches and non-graph-based approaches (GuilleAdrien *et al.*, 2013). Graph based modelling approaches treat social networks as a graph of relationships and interactions within a group of individuals allowing them to take a micro view of such networks while non-graph modelling looks at the network not concerned about the dynamics of the network.

2.4.1 The Graph Based Approach

In graph-based models, social networks are represented as graphs where members are represented by nodes and relationships between users are represented by links linking two nodes. The linear threshold (LT) model (Granovetter, 1978) and the independent cascades (IC) model (Goldenberg, Libai and Muller, 2001) are two early contextual models of the graph-based approach. In both models, it is possible to distinguish between users in the network actively involved in information spreading – seed users and those users who are inactive in the network.

The IC model works by classifying the edge between users with two classifications – “weak connection”, indicating a common relationship and “strong connection” indicating a stronger relationship. In the IC model, a seed user tries to influence one of its inactive neighbours in the network with the success of such an influence operation dependent on the propagation probability of the links between both users (Angelo, Severini and Velaj, 2016). The LT model takes into consideration the collective behaviour of users into consideration and introduces a threshold in which an assumption is made that a user would adopt information when the percentage of neighbouring users who had adopted the information had crossed the said threshold (Li, Zhao and Chen, 2019).

These two early approaches have been adopted and used in a lot of diffusion research. Two of the most fundamental and extensively researched diffusion models are the linear threshold and independent cascade models, with many more models frequently existing as expansions of these two.

2.4.2 Non-Graph Based Approach

Non-graph-based approaches model the information diffusion process in a macro view ignoring the effects of the network structure in their analysis. These models form the backbone of the epidemic models from epidemiology (Daley and Kendall, 1964). The "Susceptible-Infectious-Recovered" (SIR) model and the "Susceptible-Infectious-Susceptible" (SIS) model, which were developed to simulate the spread of information and computer viruses through online networks, are two common epidemiological models (Daley and Kendall, 1964; Pastor-Satorras and Vespignani, 2001; Abdullah and Wu, 2011). In terms of modelling the spread of information on a social network, these epidemic models classified users into different groups: those who had not heard and never spread the information (susceptible), those who had received and passed on the information (infectious) and those who had stopped passing on (recovered) the information within the network.

2.5 Models of Information Diffusion in Social Networks

Individual interaction with the rest of the world happens at the local rather than at the global level. With OSNs, individuals that were once passive receivers of information are now its active publishers and communicators having global reach thanks to a combination of ubiquitous networks and ever improving social network formations. An individual's behaviour is partly determined by their environment and the neighbours within their social network. With OSNs come connected behaviours which can result in complex properties at the aggregate level and these in turn influence individual behaviours. Early gent-based modelling (ABM) of diffusion and social networks focused on network formation using the payoff cost. Whether an agent (an individual) will choose to form a network or not depends on the payoff (Abramson and Kuperman, 2001).

Vanin's (2002) connection model, the earliest work on network-game experiment using an ABM approach, focused on the role the agent plays in the formation of their networks. By allowing open group discussions between agents – a quite common social behaviour, resulting in egalitarian rules with fairness in allocation of payoff. If considered from an information diffusion perspective this model offers a balance, as agents weigh the information presented and from multiple sources before taking a decision. While the open group discussion makes this model unique, the assumption is that all agents participating are rational, have equal influence, complete information and only a small number of links, an assumption that is not representative of modern OSNs. The model's use of a single focal point (fair resource allocation) with respect to its open discussion which indicates homogeneity amongst agents' needs is also an issue, as most agents are heterogeneous in their associations.

Li et al (2017) in their survey of Information diffusion Models and Methods in OSNs classified models into two main groups of "explanatory" and "predictive" models. "Explanatory" models extend to influence and epidemic models, and the later extends to game theory models.

They also classified basic information diffusion issues into “what”, “why” and “where” based on these models. The “what” and “why” questions relate to the explanatory models of information diffusion and the “where” question refers to the predictive models of information diffusion. Epidemic diffusion models form the foundation for explanatory models which look at the factors affecting information diffusion, what influences information diffusion, and why information diffuses in a manner like epidemics. Predictive models are posited as being used to predict future information diffusion processes in social networks.

Qian et al (2013) in their work on information contend that the spread of information differs from the spread of epidemics. While there is no threshold for the spread of illness, they contend there is a threshold that will have an impact on the diffusion of information. This implies that while in epidemic diffusion every person encountering an infected individual has the same probability of being infected, information diffusion in contrast, a user with many links may not share a held opinion if only a few of the links agree with the opinion. The result is an agent flooded with an excess of information, may not spread the information better or even spread it at all. They proposed a Fractional Susceptible Infected Removed (FSIR) model to consider the effect of neighbours on an individual in the diffusion of information.

Explanatory models like that of Xia et al (2015) and Rui et al (2018) which are extensions of the base epidemic models (SI (Susceptible Infected) model, SIS (Susceptible Infected Susceptible) model, SIR (Susceptible Infected Removed) model and SIRS (Susceptible Infected Removed Susceptible) model) consider the information diffusion process as being similar to the epidemic spread process. They seek to provide explanations to information diffusion by asking the questions: What are the main factors that affect information diffusion? Which node has the most influence? and why does the information diffuse the way it does? (Luu *et al.*, 2011). While they take an integrated approach to modelling diffusion in social networks, they offer no insight into the role of network structure and topology, dependencies between individuals in the networks as well as network externalities – all of which affect the diffusion process.

A good performance model will likely cut across the two main groups identified as it needs to understand, influence, and predict information diffusion. Such a model would have significant reference value to various applications (Qian *et al.*, 2013). The "SIAR" model, developed by et al (2015) introduced a new group that represented these authorities into the classic SIR model, which is called the “SIAR” model. Through the simulation over synthetic networks, the authors showed that the "SIAR" model could realistically characterise the evolution of the rumour propagation. Their model considers a scenario where the facts could be clarified by a few authorities enabling the ability to confirm or refute the content of network rumours. Xia et al (2015) added a new group to the traditional SIR model that reflected these authorities. The authors demonstrated that the "SIAR" model could accurately characterise the evolution of rumour dissemination through simulation over artificial networks. Their model considers a scenario where the facts could be clarified by a few authorities enabling the ability to confirm or refute the content of network rumours.

Rui et al (2018) proposed a model - Susceptible-Potential-Infective-Removed (SPIR) introducing a new group - the "Potential" groups into the classic SIR model. The "Potential" group was created to represent those who had heard about the information but did not spread it to others. By including this new group, the SPIR model outperformed the traditional SIR model in its ability to replicate information diffusion processes over synthetic and actual networks.

As was already mentioned, the process of information dispersion might be compared to a virus spreading through human interaction. The mechanism by which one individual spread an infection to another and the characteristics unique to information are what distinguish biological from information dispersion. Information dissemination is affected by how people handle information, and because the process is sufficiently complicated and imperceptible at the individual level, it cannot be observed (Luu *et al.*, 2011). Information diffusion also has little or no lag time between the first instance of an information and its widespread adoption. The condition under which an information spreads depends on the diffusion rate and network structure/topologies (Wu and Huberman, 2007).

The description of information diffusion by Wu and Huberman (2007) corresponds to the diffusion of information in OSNs. At the core of OSNs are several features: their near ubiquitous presence, their very decentralised structure and network topologies that are an evolution of the scale-free network all of which ensures a very high diffusion rate and a globally connected network.

2.5.1 The Role of Network Topologies and Structure

The topology and structure of a network greatly influences information diffusion within such networks with some network structures favouring the rapid spread of information over others. Network topologies determine a basic and important form of social interactions among agents, and behaviours of agents at the individual level (microscopic level) determine the diffusion patterns observed at the network level - macro level (Namatame and Chen, 2016, p. 35). Although people may be concerned about the outcome of such decisions on a macro level, most human social interactions are not centrally managed, and people frequently act in ways that are in their own best interests. This holds true in OSNs where there is no central management and the connections that individuals establish are done based on preference which is informed by their private beliefs.

From an information/idea adoption and diffusion viewpoint Fogli and Veldkamp (2013) argued that network topologies can matter for growth in a society since different network topologies have different implications for the spread of new information and ideas and they, in turn, can serve to contribute to or be disruptive to growth to a different extent. Using a small-world network in their model, their research showed that a highly clustered social network and hence a highly fragmental social network may inhibit the spread of information/ideas required, whereas a social network with weak ties may help promote the diffusion because it can facilitate the use of such information/ideas and the generation of new ones. The topology of OSNs offer an insight into this as their decentralised structures enable with ease the diffusion of information/ideas.

Watts (2002) proposed that some common characteristics of contagion phenomena can be explained in terms of the connectivity of the network by which influence is transmitted between individuals. Specifically, Watts' model addressed the set of qualitative observations that an exceptionally large cascade (global cascade) can be triggered by some initiators (seeds) that are small relative to the social network size. This can be observed in OSNs where very popular users with super connectivity (many links) can efficiently allow for drastic diffusion of information. Watts presented an explanation of this phenomenon as dependent on interacting agents whose actions are determined by those of their neighbours. The network topology is also essential for characterising the cascade dynamics. The cascade process can have a critical value at both the agent and network level.

Research by Cao et al (2016) explored using the evolutionary dynamics of the natural ecological systems to model information diffusion over the social networks. The authors, using different network structures and diverse types of information, modelled the diffusion process with evolutionary game theory over synthetic and real networks, analysing the evolutionary stable states. One might characterise the diffusion processes in the micro perspective using graph-based methodologies by concentrating on how each rational user in a network is influenced by neighbours and whether it affects the decision to embrace the knowledge. These models, however, were frequently constrained by the difficulties of obtaining the whole social network structure, making it impossible to use them to precisely represent the adoptions of views among users because of the diffusion process (De *et al.*, 2010).

2.6 Misinformation

Bala and Goyal (2000) showed that decentralised agents via a series of interactions and decision-making, have their network structure organised and stabilised as a Nash network configuration. Their findings indicate that even with the presence of self-interested and boundedly rational agents within a social network, the network evolves towards equilibrium rapidly. Their model is however based on the assumption that information in question is complete, thus making it unrealistic in real-world OSN scenarios as incomplete information is a feature of the information diffused in such networks.

Song and van der Schaar (2013) in their model, assume incomplete information and produce a more realistic network formation that relates most closely with OSNs. They show that when information is incomplete, network topology is of a much wider type and strength. This network can be a set within a set having multiple high-value agents. This is particularly of importance in social networks where a single user (high-value agent) can be the anchor point for thousands of links from other users who might also have lots of links on their own, having enormous social influence.

Consensus reached by users in social networks which can have real-world effects - positively or negatively, can serve to inform policy makers and reinforce or weaken societal institutions. Understanding how a consensus is reached also enables us to understand the dynamics of diffusion for a piece of information. Beutel et al (2012) and (Prakash et al (2012) modelled the propagation of different information and how they compete for users' attention. They also modelled how different information propagation interacts with each other, e.g., different information can also promote the propagation of each other.

Following the works of Myers and Leskovec (2012), Sun, Zhou and Guan (2016) and (Fu et al (2019), this research studies how correlated information informed by similar views influences each other's propagation within social networks. This leads to the conclusion that for two different pieces of information, some people who have heard about and helped spread the first information may already have some preconceived notions and knowledge about the second information. Thus, the probability for such individuals to spread the second information should be different from those individuals who have not heard the prior information and hence did not partake in spreading the first one.

Zhao et al (2013) presented the forgetting and remembering mechanisms and posited a new category of users known as "Hibernators"— people who had changed from spreaders due to the forgetting process and could change back to spreaders. Using simulations with homogeneous and heterogeneous networks, their research found that this new group of agents reduced the maximum of rumour influence and postponed the terminal time of the diffusion. Y et al (2016) considered the existence of the “super spreaders” in the networks, who can spread information much faster, and introduced a corresponding new group into the classic SIR model. Their proposed model was validated on real-world Weibo dataset and results showed that the model was able to better detect a superspreading diffusion event by identifying and focusing on the super spreader present in the network. This offered a significant improvement over the classic SIR model in detecting and characterising a superspreading event.

Most existing works focused on the diffusion process of only one information. However, in a typical OSN multiple pieces of information from several sources could be in competition with each other for diffusion and adoption. Using the Barabasi-Albert Scale free network, a network in which the distribution of links to nodes follows a power law - with preferential hubs/attachments at its core (Barabási, 2013), Wang et al (2015) proposed an Emotion-based SIS model based on the SIS model. Their research focuses on the effects of emotions on information diffusion and adoption, with seven classes of emotions used in their work. Using datasets from a real network (Sina Weibo, (http://www.weibo.com)) to perform simulations, they considered that when information is transmitted between individuals, it also expresses a kind of emotional attachment from such individuals to that information. The proportion of forwarded information within the data set that has an emotional quality was used as the weight on links in their model. They proved that information diffusion is related to propagation probability and transmission intensity. This conforms to the real-life social network characteristic of so-called super users (celebrities, influencers etc) observed in OSNs. Their work also confirms that emotions inform personal preferences in terms of adopting a particular information/idea which plays a major role in the links that an individual will establish within a social network hence also a role in the spread of misinformation.

A strategy for modelling and evaluating the influence of microblog opinion leaders based on information transfer was proposed by Chen et al (2014) as a method for modelling and measuring the influence of micro-blog opinion leaders based on information transmission. This method is based on network structure only. It mines the opinion leader by finding out who the tipping point node is in the information diffusion process. The process of information diffusion is described by the dynamic direct graph. It shows that information dissemination is weakly correlated with the number of opinion leaders. Their model shows that the propagation of a message/idea raised by opinion leaders in a social network does have substantial influence power within their followers. Using estimated parameters, they evaluated quantitatively the initial influence, influence decay rate and influence consistency of opinion leaders. The results of their experiment which was based on the data collected from Sina microblog (http://www.weibo.com), shows the total retweeted messages is weakly correlated to the number of opinion leaders in the spread process.

Understanding how misinformation could affect public opinion and the effects it might have on policy makers could be expanded by a model that provides a comprehensible narrative of how, for instance, some plausible micro-level behaviour might give birth to a surprising higher-level result. Nowak, Matthews and Parker (2017) proposed a framework for a general ABM for studying the ways in which micro-level social influence gives rise to population-level dynamics at the macro-level. They posit their model as a simple standalone model able to examine how OSNs influence generic classes of behaviour or tailored using datasets to examine specific behaviours. Their approach, which considers beliefs, norms, self-efficacy, intention, and a few outside influences, measures the "propensity" to engage in a specific behaviour. The likelihood that the person will decide and act in accordance with it is determined by this predisposition. They show within their research that when presented with information within a social network, users will either seek to conform to other users around them or do the opposite of what other agents around them do. This explains the adoption phenomena – users in a social network will more likely than not adopt/endorse the information put out by highly placed users in the network with similar beliefs. Previous research places users and diffusion in social networks in one of two states – either active or passive during the diffusion cycle. But this is often not reflective of social networks as user states can be time varying while also dependent on several factors including the information type.

A user's beliefs, the worldview they generate and the possible bias in such views are now considered key to the diffusion process and often informs the interactions of such users. Models such as the Threshold model view diffusion only in terms of the passive diffusion which presents several limitations including not considering agent interest (shaped by private beliefs) towards the information as well as diffusion is initiated by only one source of perturbation within the network. In such models' users are held not to play an active role in the diffusion process. A diffusion model that considers a user's worldview and the beliefs that shape those views as well as peer-pressure effects on such views should offer a semblance of real-life interactions found in social networks.

2.6.1 The Issue of Private Beliefs and Bias

A feature of the spread of misinformation is a reliance on differing belief systems which are found in a large OSN. The formation and evolution of beliefs has always been a focus of cognitive research. In real life scenarios when dealing with bias, **four conundrums exist** with respect to information: **"what people want to hear", "what people want to believe", "everything else" and then "the truth"**. These conundrums often form the basis of social relationships within society and are highly visible in the online social network space.

Private beliefs are often considered as a key factor that determines the social links people establish. This particularly holds true in OSNs in which user identities can be masked and more extreme beliefs and opinions can be expressed. Belief systems in social networks are known to be self-organising in terms of structure showing a scale-free degree distribution (Antal and Balogh, 2009). A question that often arises with respect to private beliefs is by what yardstick do we measure the degree of accuracy of an agent's private beliefs.

As prior stated, users within an OSN tend to endorse claims that adhere to their system of beliefs. These beliefs can be biased or unbiased and form part of the foundation that informs a user's interaction with other users. Group polarisation is a feature of such one-sided interactions as such a user's interaction will often be limited to interacting with likeminded people. It is possible to find sub-networks of even more polarised users within already polarised groups. It can be posited that users in a social network will only maintain interactions with other users in the network if the distance between their opinions and beliefs is at a minimum.

The pattern by which information spreads from one person to another is determined by the inherent properties of the medium carrying it, the internal state of the individuals who encounter the information and the information content (Crane and Sornette, 2008). Additionally, how knowledge is likely to disseminate depends on the interaction network topology within a population. This shows that to fully understand the diffusion dynamics, it is imperative to precisely represent the interaction network and user states in such networks. This is particularly important in the epidemic level spread of “Misinformation”. A recent feature amongst information diffusion via new online social media platforms is the effect of so-called forceful agents. Models of information diffusion in which Forceful agents play a role combine both predictive and explanatory models explained earlier in (Li et al (2017).

Forceful agents (users) can play to the strength or weaknesses of other agents (users) around them. In a social network, a user’s private information informs their view about the world and meeting a regular user will see both agents update their beliefs to be equal to the average of the pre-meeting beliefs (Acemoglu, Ozdaglar and Parandehgheibi, 2010). In contrast when a forceful agent meets a regular agent, this may result in the forceful agent influencing the beliefs (private information) of the regular agent so that the agent inherits the forceful agent’s belief and has a change in its original state. It may also result in the forceful agent reinforcing the private beliefs held by the regular agent. These scenarios often happen when the goal of the agents is to purposefully influence others with their opinion. This is seen with influencers and social bots on social media platforms which are agents often paid/engineered to influence the beliefs of other agents.

2.7 Solutions in Previous Works

Information propagation through online social networks has become a facet of modern society. The three regimes of the diffusion process can easily be observed for trending information in an OSN. With the rise of ubiquitous networks and the prevalence of OSNs, false information detection and containment remains severely lacking in many OSN platforms. Even in OSNs with some sort of detection, a significant time lapse between the detection and labelling of such information across the entire network of its label is a significant step towards broadcasting the real information and limiting false information (Ramezani *et al.*, 2019). **The key task is to identify early-on the propagation of false information within an OSN cluster and shut down the nodes responsible**.

When considered within social, economic, and political context, information can be posited as a complex adaptive system (CAS) as it possesses all the characteristics

(Carmichael and Hadžikadić, 2019). Like **complex systems**, information has properties that influence the way decisions, policymaking, and management for them should be properly approached. Their behaviour cannot be precisely predicted; such systems are their own best model. It is this complexity that makes irrational beliefs particularly threatening to social stability. Irrational beliefs diffusion draws its strength from the heterogeneity in our beliefs as humans as well as from poor vision and preference of people (Pancs and Vriend, 2003).

One approach to detect misinformation is based on the text content of messages (Castillo, Mendoza and Poblete (2011); Qazvinian *et al.*, 2011; Gupta *et al.*, 2014; Popat, 2017). These approaches have several limitations. Firstly, the messages sent on popular social media platforms such as Twitter are limited in word count and hence short. This makes the linguistic features extracted from them inadequate to make accurate predictions as there is insufficient data for machine learning algorithms. Secondly, these approaches are only applicable to messages that contain only text. Messages that include an image or a video are not applicable to these approaches.

Castillo, Mendoza and Poblete (2011) focused on using the characteristics of the source users (users who first put out a piece of information) for misinformation as a way to detect misinformation. With a focus on the Twitter platform, they hypothesised that it is possible to automatically determine the degree of credibility of information shared through social media by using several variables that can be seen in social media platforms themselves and that are helpful to determine information credibility. They defined a set of features to characterise each information topic in their collections and identified four types of features depending on their scope key of which was the user-based features considering characteristics of the users which post messages, such as: registration age, number of followers, number of followers. Whilst a novel approach its drawback is it ignores the characteristics of spreaders of the concerned news story, which also ensures the longevity of such news stories within the social network.

Another line of studies explored using temporal-structure features extracted from the propagation paths or networks in the early detection of misinformation (Jin *et al.*, 2013; Ma, Gao and Wong, 2017). These approaches, while being more effective at fake news detection than preliminary approaches that only adopt text, are inadequate in the early stage of news propagation and fail at spotting manipulated posts from users. More recent approaches for detecting misinformation treats information as a product and focuses on the effects of network externalities in the spread of misinformation. A particular focus is given to the role forceful agents play in creating the enabling conditions for misinformation adoption.

Aymanns, Foerster and Georg (2017) focused on the role external actors play in misinformation in the model to combat the spread of misinformation on a social learning game on a network. They base their approach on the premise that, on average, users' private information about the status of the world and, consequently, the validity of a claim, is accurate. Their model features agents within a network with an outside adversary playing the role of forceful agents (users). In their work, Aymanns, Foerster and Georg (2017) they show two key things. First, if agents are educated with knowledge of the existence of an enemy, they have a different optimal strategy. When agents are aware of an adversary's presence, the adversary's ability to spread false information is significantly reduced. This emphasises the value of informing readers that assertions made even in news sources that seem reliable could be wrong. Second, if the adversary is aware of an agent's location in the social network and their private signals, the adversary's chances of success in an attack increase. This approach is limited in its view of forceful agents having a profound effect on the private beliefs and the internal states of users.

Study Reference	**Solution Summary**
Castillo, Mendoza and Poblete (2011)	Using features of seed users that initiate the diffusion process in a network to detect misinformation.
Qazvinian et al. (2011)	Framework that employs statistical models and maximizes a linear function of log-likelihood ratios to retrieve rumours relying on different categories of features, including content-based, user-based, and network-based features, in capturing tweets that show user endorsement.
Jin et al. (2013)	Using an epidemiological modelling approach by adopting a the SEIZ (susceptible, exposed, infected, skeptic) model to capture the adoption of news and rumours on Twitter.
Gupta et al. (2014)	Machine learning algorithms to automatically evaluate the credibility of a tweet based on various Twitter feature including user features.
Popat (2017)	A Natural-language text-based model for assessing credibility of textual claims in OSNs.
Ma, Gao and Wong (2017)	A kernel-based approach that leverages the propagation paths information take in a of microblog posts to detect rumours.
Aymanns, Foerster and Georg (2017)	A social learning game model on a network to explore the spread of news and strategies for identifying false information as well as a multi-agent deep reinforcement learning approach to model the behaviour of social and economic agents.

Table 3: Summary of Solution Presented in Prior Works

Table 3 summarises some solutions presented in prior works to detect irrational information in OSNs. To address the limitations of existing works, there is a need for an approach that takes into consideration the individuality of users in a social network. Understanding that user behaviours in a social network will differ due to differing belief systems allowing these users to play different roles in the diffusion process. Differing belief systems will also see clustering amongst likewise users adopting similar beliefs in the network producing emergent properties that will influence the network.

2.7.1 Solutions in Recent Works

Recently, **Graph Neural Networks (GNNs) have been adopted in modelling information diffusion in social networks as they have demonstrated to be powerful in learning on graph data (Fan *et al.*, 2019).** GNNs are a deep learning-based method that operate in the graph domain and can naturally integrate node information and topological structure in their operation. As a data structure which models a set of objects (nodes/users) and their relationships (edges/connections), graphs are rich in relational information among elements and are ideal for simulating OSNs. Increasing interests in extending the DL framework to graph operations has led to the new definitions to handle the complexity of graph data.

By converting social networks into graphs, it is possible to model with high fidelity the interactions between users in the network while embedding each user with private beliefs as node embeddings. Graph Neural Networks have been posited as being able to learn and classify user states in social networks. As a connectionist model that captures the dependence of graphs via message passing between the nodes of graphs, they are ideal for simulating social networks. Such models also allow for the simulation of heterogeneous networks where there are several kinds of users, each capable of having different attributes (Zhang *et al.*, 2020). Figure 5 shows a social network of 80 users displayed as a graph.

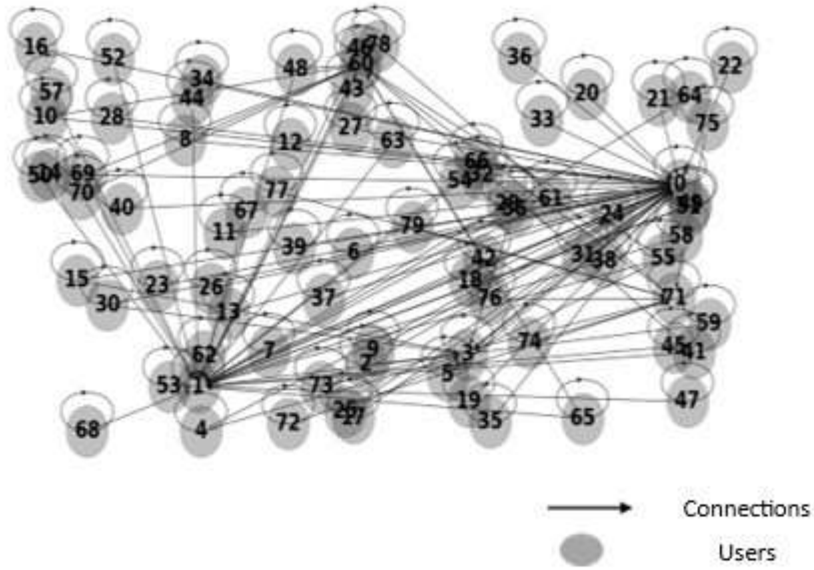

Figure 5. Online Social Network displayed as a Graph Network
The figure shows an OSN of 80 users generated as a graph network where each node in the network represents a user.

These advantages of GNNs provide great potential to advance modelling the internal states of OSNs which is key to understanding information diffusion within such networks. Most recently, research on GNNs has focused on representation learning**. One line of studies focuses on node-level representation learning – node embeddings and how this can help in node-link predictions, node classification and graph-level classification (Li *et al.*, 2016; Hamilton, Ying and Leskovec, 2017; Kipf and Welling, 2017a)**. Such research is particularly important as interactions that occur at the node (user) level form the fundamental building blocks of social networks. Li et al (2016) proposed an end-to-end predictor for inferring cascade size by incorporating recurrent neural network (RNN) and representation learning.

To detect fake news early, Qian et al (2018) proposed a novel model featuring a Two-Level Convolutional Neural Network with User Response Generator (TCNN-URG) which captures semantic information from article text by representing it at the sentence and word level and learns a generative model of user response to article text from historical user responses for early detection of fake news. This model captures semantic information from article text by representing it at the sentence and word level.

A second line of studies focuses on graph representation learning (Ying *et al.*, 2018; Gao and Ji, 2019; Bianchi, Grattarola and Alippi, 2020). Cangea et al (2018) in their work demonstrate that competitive hierarchical graph classification results are possible without sacrificing sparsity. Their model combines several recent advances in graph neural network design. This is particularly important in terms of graph classification as most existing approaches are poor at predicting a single label for an entire graph. However, predicting a single label for an entire graph in social networks will not be applicable in the presence of heterogeneous users.

Using an approach that combines ABMs, graphs and neural networks to model information diffusion in OSNs, this research presents models that allow for the internal states of users within such networks to be learnt and environmental states to be considered offering an advantage over prior models. One such advantage are user adaptive behaviours which can be implemented using this approach highlighting the dynamic nature of the system. This enables models that are more informed of the state of the social environment and/or users, whilst being able to react to changes from internal and external actors/sources.

2.8 Review of Simulation Tools

When dealing with ABMs, the common strategy is to model and simulate actual scenarios using several self-governing agents that are either represented as simple entities inside the computational code snippets or as highly sophisticated objects (Abar *et al.*, 2017). To provide specific capabilities such as intelligence/attributes for agents and their environments, general **programming languages such as Python, Java, and C++, and C** also can be used outside of those already provided by **specialised agent modelling tools** (Macal and North, 2010). Several ABM tools have been developed over the years and features common to these tools include:

▪ Agent-based Centric System: The modelling tool always adopts an agent-based approach to modelling and simulation. Autonomy is at the heart of the agent being simulated in these tools and the system should be able to interact with agents individually as well as collectively.

▪ Interaction Network and Parameters: Agent interaction will vary both in terms of strength and space hence the need to define the interaction network (Railsback and Grimm, 2011).

▪ Data representation/output: The results from the simulations would have to be easily displayable most commonly in 2D format using graphs and charts in a manner that is easily representable and understandable (Railsback and Grimm, 2011).

▪ Effective Environment implementation: The environment should support multiple scenarios under which agents can be simulated. Scenario setup should be external and independent, to the simulation engine which would prevent a change in a scenario should not require re-compilation of the code (Abar *et al.*, 2017).

2.8.1 ABM Simulation Tools

An extensive list of ABM tools has been developed allowing for the agent-based simulation across different agent interaction behaviours, programming languages, complexities, and scalabilities. To implement large-scale models, a range of services is required. Project specification services, agent specification services, input data specification and storage services, model execution services, results storage and analysis services, and model packaging and distribution services are just a few of these frequently requested services (Macal and North, 2010). ABMs are mainly classified based on two factors:

▪ Agent Development Environment

Non-dedicated agent-based environments such as spreadsheets like Microsoft Excel are generally considered to be the simplest approach to modelling as they are much easier to use to develop agent-based models than with many of the other ABM tools. However, these are limited by the model results which include limited agent diversity, restricted agent behaviours, and poor scalability (Macal and North, 2010).

Dedicated Agent-based Prototyping environments are special-purpose ABM tools that provide special facilities focused on ABM. One such popular free ABM tool is NetLogo, developed at North-Western University's Centre for Connected Learning and Computer-Based Modelling (Wilensky, 1999).

▪ Scalability

The issue of scalability arises when the model is to simulate tens of thousands of agents in a virtual environment. Such simulations will have such high computational requirements that may exceed the computational power of simple computers used in small scale simulations. One popular, free and open-source ABM toolkit for large scale simulation is Repast (Recursive Porous Agent Simulation Toolkit) developed as a pure Java implementation (North, Collier and Vos, 2006).

One of the key advantages of ABM is the agent specification feature, which gives modellers a way to specify the properties and behaviours of agents. General-purpose programming languages like C++, Python, or Java as well as textual domain-specific languages (DSLs) like MATLAB can be used to implement these concepts (Macal and North, 2009). Most implementation environments also provide special support for features such as adaptation, optimization and learning using neural networks useful to applications in genetic algorithms and social networks (Macal and North, 2010). Table 4 lists some ABM simulation tools highlighting some of their basic features and attributes.

ABM Simulation Tool	Programming Language	Development Environment	Scalability
AnyLogic	Java	Dedicated	Large-scale
Breve	C++, OpenGL	Dedicated	Medium scale
Echo	C++	Non-dedicated	Large-scale
FlexSim	.Net, OpenGL	Dedicated	Small-Medium scale
Java Enterprise Simulator (JES)	Java	Dedicated	Small-scale
Mesa	Python	Non-dedicated	Small-scale

Table 4: Comparison of various ABM simulation tools.

2.8.2 Research Simulation Tools

A new paradigm for ABM tools has also been introduced by GNNs which has seen a transition in ABM systems from "data-poor systems" to "data-rich systems". By adopting a GNN based ABM simulation approach in simulating a social network, this research can present a detailed description of the system's components, including users, their links, and the information network. The model's framework design architecture and implementation were done across three model simulations. The model named Network Translation (NetT) has three versions namely, Network Translation Version One (NetTv1), Network Translation Version Two (NetTv2) and Network Translation Version Three (NetTv3). Each version exists as an iteration over the previous version.

Model	Integrated Development Environment (IDE)/Libraries	Programming Language	Implementation Platform
Network Translation Version One (NetTv1)	Social Network Visualiser (SocnetV - Dedicated environment)	C++	Laptop (Windows Operating System (OS))
Network Translation Version Two (NetTv2)	Visual Studio, NetworkX, Deep Graph Library (DGL), Matplotlib, NumPy (Numeric Python)	Python	Laptop (Windows Operating System (OS))
Network Translation Version Three (NetTv3)	Visual Studio, NetworkX, Deep Graph Library (DGL), Matplotlib, NumPy (Numeric Python), Pandas	Python	Laptop (Windows Operating System (OS))

Table 5: Research Simulation Models - Features and Attributes

Table 5 shows the hardware, software's, libraries, and programming languages used to implement the various models. The hardware component was a laptop (Lenovo ThinkBook 14) while the software component consists of several open-source libraries and tools put together and built in an integrated development environment (IDE).

- Social Network Visualiser (SocnetV) is an open-source platform for social network analysis and visualisation platform (Kalamaras, 2015). SocnetV performs matrix operations, centrality, and prestige indices, as well as common graph and network cohesion measures (such as density, eccentricity, clustering coefficient, etc.). It supports quick algorithms for network generating models, structural equivalence analysis, loading and editing of multi-relational networks, and community discovery. There are several other open-source SNA tools easily available online such as Gephi, NetworkX and Graphviz all of which are very powerful tools for creating and analysing the structure and dynamics of social networks. SocnetV was chosen for its user-friendly and easy to use Graphic User Interface (GUI).

- NetworkX is a python programming-based package for the visualisation, creation, manipulation and study of the structure, dynamics and complex networks such as graphs (*SourceForge.net: Project Info - NetworkX*, 2005). An alternative to NetworkX is iGraph. Originally written in the C programming language, igraph is suitable for very large graphs, or complex computations. The choice of NetworkX was informed by its flexibility and API which enables easy interface with other libraries used in the model.
- DGL is a graph-oriented library with GNN and its variants built for deep learning on graphs (Wang *et al.*, 2019). DGL has features like that of other deep learning graph libraries like Pytorch Geometric (PyG) and Spektral. All libraries offer high-performance and are scalable python libraries for deep learning on graph The research adopted the use of DGL due to its higher levels of abstraction and a similar graph interface to the NetworkX tool adopted. Using DGL also allows for the combination of graph-based modules (pytorch) and tensor-based modules (TensorFlow) - unlike PyG which only works on Pytorch and Spektral which uses keras and TensorFlow.
- Matplotlib is a cross-platform library that allows for creating static, animated, and interactive visualisations in python from data arrays (*Matplotlib 3.4.3*, 2003). While matplotlib is the oldest python plotting library, there are other libraries which offer similar functionality. Plotly is one such library that offers analytics and visualisation for graphs.
- NumPy (Numeric Python) is a python library tool used for working with large, multi-dimensional arrays and matrices. It provides a collection of high-level mathematical functions to operate on these arrays (Harris *et al.*, 2020). SciPy library is a popular alternative to NumPy, both of which offer tools for powerful data analysis. The choice of using NumPy was due to its support for arrays which are used within the programming language implementation in the project.
- Microsoft Visual Studio - a suite of component-based software development tools and other technologies for building powerful, high-performance applications was used as the IDE.

2.8.2 Research Simulation Datasets

The various models also use different datasets as a part of the simulation. Two groups of datasets were used for each model – a synthetic (artificial) dataset and a real-world dataset. These datasets were representative of social networks used in the model simulations. Table 6 provides details on the dataset used in the research model simulations.

Model	Synthetic Dataset	Real-World Dataset	Number of Users (Nodes): Synthetic Dataset	Number of Users (Nodes): Real-World Dataset
NetTv1	Scale-free Network	YouTube Network and Weibo Network	200	12694 (YouTube), 9727 (Weibo)
NetTv2	Custom Generated Network	Zachary’s Karate Club	16	34
NetTv3	Custom Network	Zachary’s Karate Club	80	34

Table 6: Datasets used in Research Models

▪ **NetTv1:** The synthetic dataset was a social network generated as a scale-free network in the SocnetV platform. The choice of using the scale-free network as the primary data sample over other network types was based on their use of a growth and preferential attachment mechanism for the establishment of connections between nodes which is representative of real-world social networks. The Small World (SW) network is an alternative to the SF network as they are representative of offline social networks (Watts and Strogatz, 1998). Like SF networks, SW networks are close structurally to many social networks in that they have higher clustering between nodes and inhomogeneous distribution of degree. Their limitations in their growth mechanism and lack of preferential attachment limits their use to small sized networks and are not representative of OSNs hence they are not used.

The real-word datasets in the model were used to validate the results from the synthetic network. The first dataset which is representative of a social network growth focuses on the ways in which new user-user links are created in such networks. It is an anonymized YouTube dataset of nodes within a directed network. The second dataset is an anonymized Tencent Weibo social network dataset showing nodes and their edges also within a directed network. It includes an hourly instance of the Weibo system's suggestion events. The edges reflect items recommended to users and follow-relationships between users, whereas the nodes represent users and items.

▪ **NetTv2:** uses a custom dataset for the synthetic network featuring a social network created as a graph network. The nodes in the created network were initialised with specifically defined connections The network was implemented as a graph convolution neural network setup (Kipf and Welling, 2017a). The dataset used for validation was the real-world network Zachary's Karate club dataset, a well-known dataset that describes the intricate relationships representative of that of a social network (Zachary, 1977). The Zachary's network's modest size, coupled with its density, presents both computational feasibility and the complexities intrinsic to genuine human interactions. This network's display of social influence—which emphasises the significant significance of interpersonal ties on people's ideas and affiliations—is one of its most important features. Despite its seemingly simple structure, the network encapsulates embedded communities, offering insights into the spread of beliefs within specific subgroups. Consequently, lessons derived from this network often resonate with larger social structures, underscoring its value as a microcosm of broader social dynamics.

▪ **NetTv3:** A custom dataset serves as for the synthetic network. As with NetTv2, the network was implemented as a graph convolution neural network setup (Kipf and Welling, 2017). A custom Zachary's Karate Club dataset serves as the real-world network validation dataset (Zachary, 1977).

For each model in using the datasets, simulations are done with both the synthetic datasets (artificial network) and the real-world datasets (real network). The real-world datasets are anonymised, comply with the usage polices of both social network platforms and meet the university's research ethics as approved and seen in the research ethics disclaimer form in **Appendix E**.

2.9 Conclusion

This chapter analysed studies in the field social networks with a focus on the diffusion of information and belief adoption within OSNs. The review provided a ground truth for the research problem as identified in misinformation. Reviewing existing literature, it is established that adopting an ABM approach in combination with social networks in modelling is a promising approach to simulate interactions between individuals, their environments as well as the effects of their individual state in such interactions.

Whilst studies have been done on modelling misinformation in social networks with a view to proffering a solution via early detection. However, with the exception of a few works (Usó-Doménech and Nescolarde-Selva, 2015; Seo, Raman and Varshney, 2020; Enders *et al.*, 2021), there is a distinct lack of value placed on the role of belief systems and the internal states of users. This research addresses this gap by replicating much more realistic user characteristics found in OSNs. This line of study sees the introduction of distinct profile features aligned with the user classes established as part of the definitions posited in this project. These profiles allow for the research to model differing belief systems amongst users enabling the different roles that these users play in the diffusion process to be established. Heterogeneous graphs where users and their connections can have independent features are also used in this research. User states have been identified in prior works as being key to diffusion in OSNs and this is further explored positing that the roles the various user classes play during the diffusion regimes differ and allowing for the identification of key users that are enablers of misinformation diffusion in the network.

3. Methodology

3.1 Introduction

Much like epidemic diffusion, information diffusion has similar characteristics – (i) the diffusion content and the need for interacting agents to diffuse, (ii) the degree of activeness of the agents involved. The diffusion process can be broken down into three components namely: the population on which they unfold (target group), the diffusion mechanisms/ medium and the contents of the diffusion (Milli *et al.*, 2018). These components are important to effectively model, understand and simulate a diffusion process.

This chapter presents the research methodology and proposed model framework that provides the context for irrational agent beliefs modelling. The design of the framework enables iterative and scalable developments both in terms of the model definitions and the simulation mechanisms. Each of the iterative models complements and extends the previous models providing a level of abstraction and separation for earlier simulations to be re-run. Using the framework, the following are presented:

▪ The interconnectedness, dependency and the structure amongst the problem statement, literature, methodology, simulation, data collection and analysis.

▪ The selection process employed for choosing an environment suitable for social network simulation.

▪ A justification of the tools used in design and implementations of the model and the relationships between them if any.

3.2 Research Methodology

This research incorporates both qualitative and quantitative research methods, hence adopts a mixed-methods methodology. Qualitative methods offer depth, context, and a nuanced understanding of the complexities of irrational beliefs in OSNs. This is achieved through a comprehensive review of existing research, articles, and studies related to irrational beliefs in OSNs. This will provide a theoretical foundation for the research and highlight gaps in current knowledge that the research aims to address. The quantitative methods provide hard data and measurable outcomes and is achieved through:

- Simulation & Modelling: Developing a computational model to simulate the spread and dynamics of irrational beliefs in OSNs. This involves defining agents, constructing the network, and running the model to understand its behaviour.
- Data Collection: Collecting data from online social networks (OSNs) through publicly available datasets.
- Network Analysis: The structure of the network will be examined. Centrality measures, and other evaluation metrics would be used to get insights into influential nodes and information flow patterns.
- Statistical Analysis: This will involve analysing the data to identify patterns of behaviour, correlations, and other statistically significant phenomena related to irrational beliefs.

3.2.1 Research Approach

The descriptive and practical nature of the research informs the research approach of the research. For the quantitative research approach, simulations are used as the preferred method to implement the model. Such a strategy entails creating an artificial environment in which pertinent information and data can be produced, enabling the observation of a system's (or a system's subsystem's) dynamic behaviour under controlled circumstances (Davidavičienė, 2018). The characteristics of OSNs of being able to evolve and adapt to changes make the use of simulations ideal for their modelling.

Using simulations to implement the models created offers the best way to define and create scenarios relating to the research objectives (Bellinger, 2004). Complex interactions of data and the violation of key assumptions made in the research is possible while allowing for iterative development of the model and for the behaviour and performance of the model to be studied while testing the hypotheses.

3.3 Research Method

The research adopts computational modelling and simulation as the most appropriate research method. Computational modelling involves the use of computers to simulate and study the behaviour of complex systems using mathematics, physics, and computer science (Imbert, 2017, pp. 735 – 781). In the context of social networks, this method can be particularly useful to understand and predict the spread of beliefs, information, or behaviours across a network**.** Model construction, where 'agents', representing network users, are defined is the first step. Their interactions, belief systems, and how they modify their beliefs based on these interactions are crucial components. The next step is parameterisation, where the research assumptions help set the model's parameters, like the probability of a user altering their belief upon encountering differing information.

To ensure that the model's behaviour mirrors real-world situations validation is needed. Being controlled simulations will aid with this as the simulations can be run under varied scenarios, to comprehend all possible factors affecting belief propagation. Lastly, outcomes from the simulations are analysed to yield deep insights into how irrational beliefs proliferate. For this research, computational modelling and simulation would allow for:

- Understanding the Dynamics: By simulating various scenarios, one can gain insights into how certain structures or parameters influence the spread of irrational beliefs.
- Predictive Analysis: Once the model is validated, it can be used to predict the spread of beliefs under certain conditions or interventions.
- Testing Interventions: The model can be utilised to simulate the impact of different interventions, like fact-checking, on the spread of irrational beliefs.
- Iterative Refinement: Based on the results from the simulations, the model can be continually refined to improve accuracy.

3.4 Research Questions

3.4.1 Question One

How do private beliefs influence the diffusion of information within a social network?

Belief systems are opinion structures that normalise a personal sense of reality. The formation of private beliefs is influenced by the interrelation between several beliefs. Individuals have a system of beliefs that is utilised as part of social interactions and is the mechanism through which individuals make sense of events happening around them. One feature of belief systems is that they vary almost infinitely in substantive transport. Different belief systems are frequently in play in social networks, and these systems frequently draw on episodic content from personal experience, cultural belief systems, folklore, or political doctrines, or both (Usó-Doménech and Nescolarde-Selva, 2015).

▪ **Hypothesis:**

1. ***Users within a social network have differing roles and belief strengths. This can mean different genre of users in the network with differing roles and level of belief strengths This influences interactions between users in the network.***

As a solution to the question asked, this research presents the novel concept of User classes to classify users based on their roles, belief structure and positions in the network. Using this class structure - user roles within the network and user attributes are defined for users as grouped into various classes. The class structure would allow for the simulation of user interactions in the presence of differing belief strengths. The effects of the beliefs and interactions of other users in a network on a target user is also explored.

3.4.2 Question Two

How can instances of irrational information diffusion in OSN interactions be identified?

Fake news, false information, misinformation, and disinformation – are terms that are often used interchangeably and hence tagged under the same umbrella. Previous research has shown that a person's level of education and information literacy, or media literacy, are the two main elements that influence their ability to identify misleading information (Busselle, 2017). This means that if a person is stronger at critically evaluating information from any source or more knowledgeable with the subject matter and methodology of information study and presentation, they are more likely to spot disinformation.

Online Social networks thanks to their ubiquity and network structure often serve as a ground for the spread of misinformation. There are still questions as to what exactly is behind the sharing of false information as well as the motivation behind why such information easily spreads through social media platforms (Chen *et al.*, 2015). There are two major reasons why combating the spread of false information is difficult: the plethora of information sources, and the presence of echo chambers. First, the abundance of information sources makes it more difficult for people to evaluate the veracity of the information they encounter. The unreliable social cues that come with such knowledge serve to emphasise this (Meserole, 2018). People tend to follow or support others who share their opinions, which results in the creation of echo chambers and filter bubbles.

Second, users in an OSN also play a role in the spread of false information. According to Chen et al., (2015) study of Facebook users, distributing false information was most frequently done for social reasons rather than because the user believed the material to be true. Despite the possibility that users are not deliberately disseminating erroneous information, it is nonetheless happening.

▪ **Hypothesis:**

2. ***It is posited that private beliefs and the beliefs systems which inform them play a major role in the connections users establish in a social network and hence significantly influences information diffusion within the network.***

The role of belief is observed in interactions in OSNs - users interacting with others and/or contents that align with their private beliefs. The means that users will likely interact with users who have similar beliefs with them in the network (Jimenez-Martınez, 2015; Usó-Doménech and Nescolarde-Selva, 2015; Xia *et al.*, 2015; Namatame and Chen, 2016; Sasahara *et al.*, 2020). A result of these interactions are clusters of users of identical beliefs. The second hypothesis considers that these clusters of users which results in highly polarised interactions between users in the network can contribute to the diffusion of misinformation hence the need to understand the dynamics behind these belief systems which are informed by the internal state of users in a network.

3. ***It is posited that a user can be affected by neighbouring users in the network in terms of the spread of misinformation and belief adoption.***

The effects of peer pressure have been captured by previous models (Granovetter, 1978; Moussaïd *et al.*, 2009; Myers and Leskovec, 2012; Chen *et al.*, 2014; Sun, Zhou and Guan, 2016). One drawback of these models is they fail to capture the effects of private beliefs in the presence of peer pressure. The neighbourhood effect encompasses the effects of peer pressure in OSNs and is underpinned by the assumption that users will adopt a belief because others in their vicinity have adopted such beliefs. The third hypothesis considers how this influences the spread of misinformation in networks by determining the effects a user's neighbours will have on its state during the diffusion process.

3.4.3 Question Three

How might Artificial Intelligence (AI) algorithms be employed to mitigate the spread of irrational information?

A piece of information becomes false information if the information presented is in an incorrect format or if the information doesn’t represent or contain the facts that it is expected to carry. A dilemma about information is that when it comes to information, it is not just about the type but also about assuring the veracity of information as it is about diffusing, processing, and securing information. Combating false information diffusion in social networks is also a big data veracity issue as information sources are vast, can easily be created and fabricated (Cassauwers, 2019).

The spread of false information across OSNs is not only about accidental inaccuracies, there also exists a more intentional dynamic attempt to spread misinformation (Allcott, Gentzkow and Yu, 2019). AI algorithms have increasingly been used to detect false information using a whole range of approaches. With the bulk of this information coming in the form of opinions that target certain belief systems, in-depth and sophisticated AI centred approaches are required to establish information authenticity as one of the most important aspects of combating false information diffusion.

▪ **Hypothesis:**

4. ***Classifying information propagation paths in a social network can allow for control in terms of Information diffusion cycle, allowing for false information to be identified early-on. This however will be dependent on the availability of user characteristics as well as on the classification method used.***

The fourth hypothesis considers how diverse user characteristics can be used in detecting and classifying misinformation. Users play differing roles during the diffusion of information in the network. Due to differing private beliefs and levels of bias, some users are more active than others in the diffusion process. The roles, beliefs and levels of bias can be used as a part of user characteristics to discriminate amongst users in the network in terms of the belief adoption and misinformation diffusion.

3.5 Research Framework

The benefits offered by AI systems in social networks make possible the development of a huge number of applications, which are increasingly becoming available to our society. From recommendation systems to user verification systems, many are the domains in social networks in which new AI applications would likely improve the quality of interaction and service. In general, architectures for AI applications relating to information diffusion in social networks will share some similarities such as having essential building blocks (input datasets), the approach (graph or non-graph), network type (artificial or real social network) and evaluation metrics.

To meet the research aims, a framework that addresses the research objectives and questions asked was put forward. This ensured that the challenges encountered could easily be addressed, key of which was the challenge of multiple entities which are typically encountered in a social network and its environment and that the environments use and serve. A social network can be viewed as a smart environment which often exists as a multi-agent system - agents can have differing roles (Cook, 2009). The word "agent" used in the context of a social network can be looked at from two perspectives.

▪ First perspective, an agent can refer to the users that make up the social networks.

▪ Second perspective, it is argued that these users who typically form clusters based on their interactions and beliefs of sub-networks is a natural multi-agent setting.

In the second case, social network environments themselves can be viewed as intelligent agents able to adapt to changes around them. These environments often contain sub-networks of users organised along various lines such as interests, beliefs, and level of engagement. In addition, software needs to be designed to reason about and interact with each node, thus transforming the software itself into a multi-agent system.

In designing and implementing the framework architecture for the research, two factors are taken into consideration:

1). Definition of task to be performed by the model.

In defining a task, each model design is based on an idea or a problem that requires a solution. These are in the form of questions on what exactly was required to be achieved and the feasibility of the implementation.

2). Requirements for success of the tasks identified.

Requirements for realising the design architecture was the next step once the idea had been established. The need to determine the hardware and software requirements for the project. As an ABM design, the requirement choices must satisfy the tasks identified while being agent focused.

3.6 First Model - Network Translation Version One (NetTv1)

The model extends the classic information cascade (IC) diffusion model where nodes (agents) in the social network are in one of two states – “Active” or “Inactive” for each time step. Validation was achieved by comparing the results from the simulation with the results from a real network simulated under the same conditions.

3.6.1 Design Architecture and Implementation

▪ ***Task***

The task of this model was to establish the internal structure of a social network examining the relationships and dependencies between users in the network.

The architecture was based on scale-free (SF) network representation of an OSN in the context of information spread with social network analysis (SNA) performed on the network created. This allowed for the initial veracity of the research hypotheses to be established. The synthetic network was created using the Barabasi-Albert’s model algorithm of preferential attachment where the more connected user is more likely to receive new connections (Barabási, 2013). Figure 6 shows the component modules of the model. The model is made up of three modules: an input module, a SNA module, and an output module.

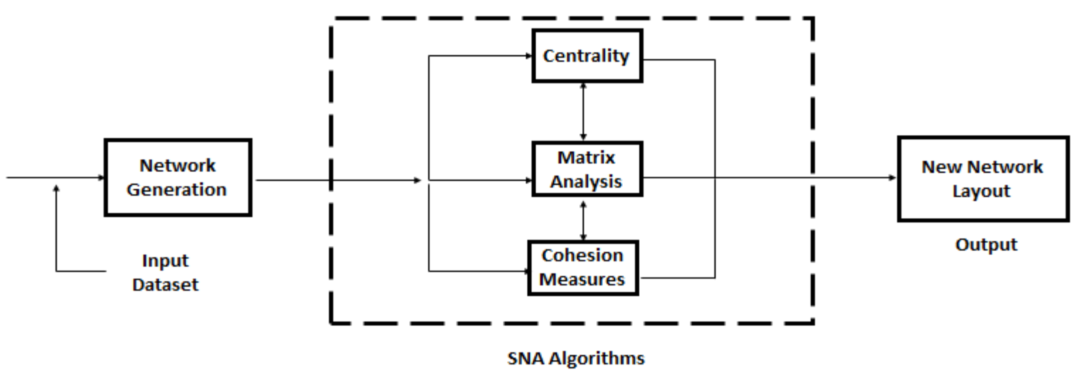

Figure 6. Architecture of NetTv1
A flowchart depicting the simulation process of NetTv1. An input dataset used for network generation. This generated network undergoes analysis to highlight its internal dynamics.

▪ ***Model Simulation Requirements***

The hardware requirement for the NetTv1 simulation was a laptop. The model was not hardware specific, so it is possible to use any laptop or desktop but the capabilities of the hardware in terms of computing power affects the model operations hence informing on the type of software module to be used. A minimum RAM size of four gigabytes is recommended for any laptop/desktop used. The software used in the model implementation was Social Network Visualiser (SocnetV), an open-source platform for social network analysis and visualisation platform (Kalamaras, 2015).

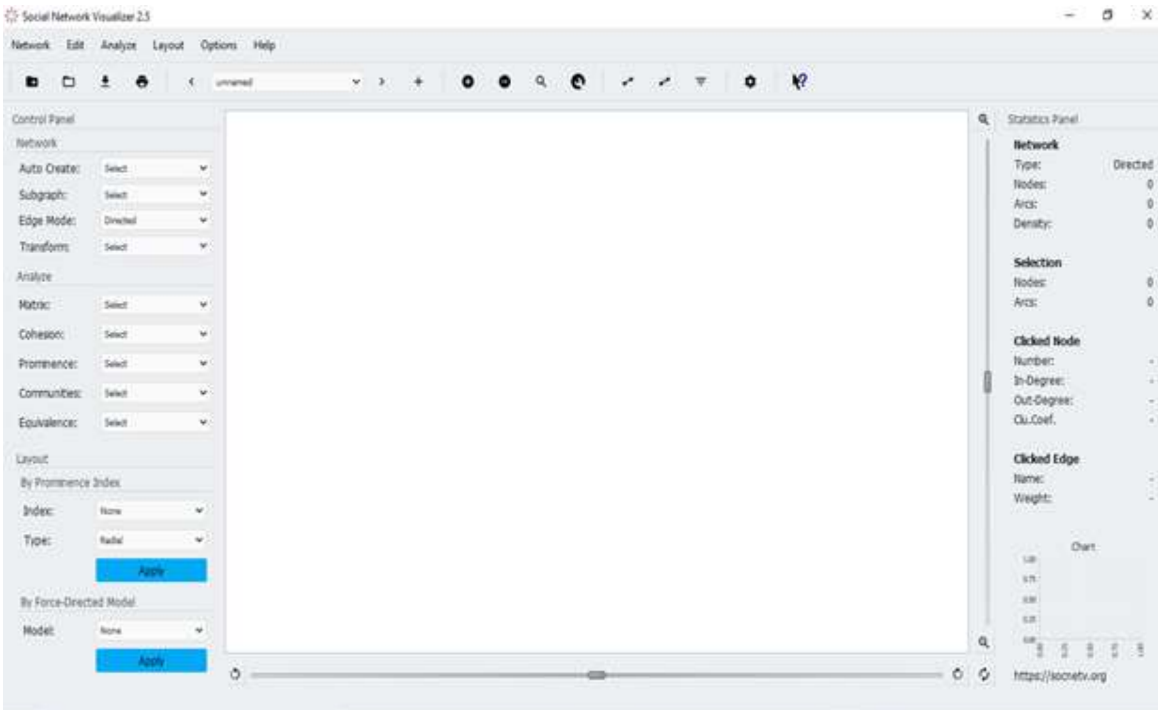

Figure 7. SocnetV Graphic User Interface (GUI) (Kalamaras, 2015)
A screenshot of the Social Network Visualizer 2.5 interface. The control panel on the left offers various options for network creation, analysis, and layout adjustments, such as edge mode selection, metric analysis, and layout configurations.

SocnetV offers an easy to use Graphic user interface (GUI) as seen in Figure 7, as well as the ability to create online reports in HTML format and to plot the network in layouts in rich visual content based on the data analysis metrics (Faysal and Arifuzzaman, 2018).

▪ ***Research Question and Hypothesis***

The task of this model relates it to the first research question and the hypothesis put forward as a possible answer. The research question seeks to establish the role user beliefs play in the diffusion of information in a social network. The first hypothesis describes a user's private belief as central to their interactions with other users in an OSN. To answer this question by proving the hypothesis, it is necessary to understand the internal structure of a social network. This is important to know how belief systems are formed and roles they play in the networks. NetTv1 is used to delve into this internal structure using synthetic and real-world social networks as contained in the datasets used to establish the nature of user relations in a social network, examine the dependencies between these relations if any and understand how this affects diffusion/belief adoption in the network.

3.6.2 Data Sample, Analysis and Validation

A scale-free network was used as the artificial network providing the synthetic dataset. Centrality metrics were used to perform data analysis. Centrality concepts which were first developed in social network analysis are popularly used in graph theory and network analysis; they assign numbers or rankings to nodes within a graph corresponding to their network position with varying applications (Newman, 2010). Datasets from a real network were used to validate the results from the synthetic network with the centrality metrics used in the SNA of the synthetic network also used in the analysis of the real network comparing results obtained.

3.6.3 Simulation Summary

The SocnetV platform was used to simulate both the synthetic and real networks. The synthetic network is simulated as a self-generating network. The real network was a dataset loaded into the platform via an input file and run. The simulation allowed for the visualisation of the internal structure of the networks. Using several centrality metrics allowed for the examination of connections between nodes in the network by identifying key nodes in the networks accomplishing the primary task of the simulation.

3.7 Second Model - Network Translation Version Two (NetTv2)

This model extends the first model and features user interactions that are simulated at the micro-level. Adopting a graph-based approach, the model allowed for users in the network to be classified based on their beliefs.

3.7.1 Design Architecture and Implementation

- ***Task***

The primary task of this model was user classification via User Representation Learning, where the model predicted the ground truth category of each user based on their internal belief structures.

The design architecture centres around the Graph Neural Network (GNN) framework. Using this framework allowed for the simulation of user interactions in the presence of user labels analogous to beliefs in this research. Termed as user-level classification, using graph filtering to generate node representations for each node in the network. This allows for a better understanding and characterization of user beliefs and their roles in diffusion within an OSN. Using the GNN paradigm, a graph representation of the problem was created and implemented as a graph convolution network (GCN), a variant of GNN. The choice of using a GNN was informed by its ability to learn from graph-structured data including that of neighbour nodes. An alternative to GNN is the Convolution Neural Network (CNN). CNNs are very useful in tasks like image classification and like GNNs use deep learning algorithms. However, unlike GNNs, CNNs lack the ability to learn graph data and can't be used for representation learning operations.

The model is made up of four modules: an input module, a labelling module, a neural network (NN) module and an output module. Figure 8 shows the internal structure of the model.

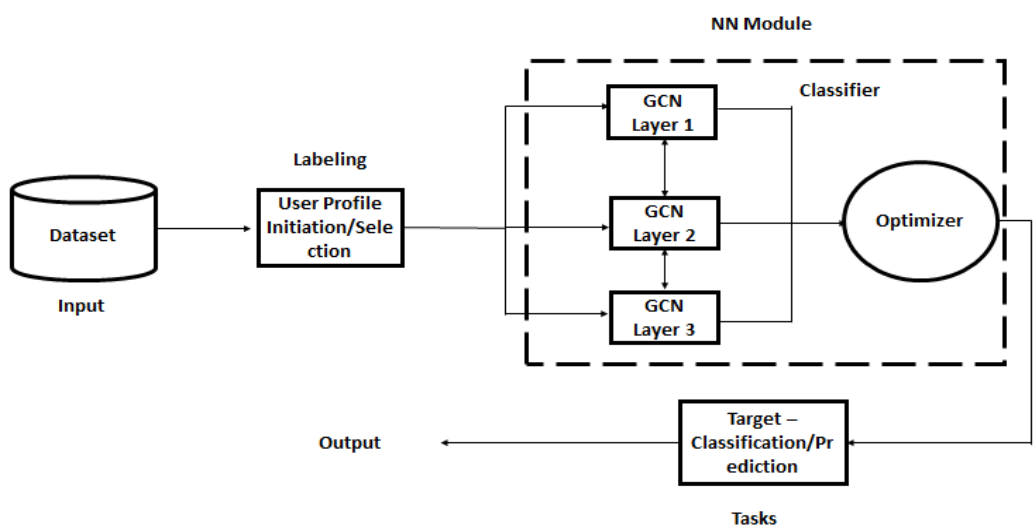

Figure 8. Architecture of NetTv2
A diagram illustrating the process of user profile classification using neural networks. The labelled input dataset feeds into a neural network (NN) module consisting of three Graph Convolutional Network (GCN) layers. Post-processing through a classifier, the results are optimized, leading to the final output: target classification or prediction of user profiles.

▪ ***Model Simulation Requirements***

Like the previous model, the hardware component used for the model implementation is generic - a laptop (Lenovo ThinkBook 14) with an intel core-i5 10th generation central processing unit (CPU) running at 1.60ghz. The software component was purpose built, combining several open-source libraries and tools together including NetworkX, DGL, Matplotlib and NumPy and built in an integrated development environment (IDE) – Microsoft Visual Studio.

▪ ***Research Question and Hypothesis***

This model answers the second research question and the hypotheses put forward as possible answers. The research question seeks to establish if early detection of misinformation in OSNs is possible. The hypotheses describe the effects of a user's private belief in the connections they establish as well as the effects of their neighbourhood on whether they adopt a belief.

The task of NetTv2 is achieved by simulating the network internal dynamics using GNNs in an ABM approach. User interaction in the presence of differing opinions in a network is simulated. By having users learn two representations of a given belief, the effects of a user's connections on their final belief state in the network is explored. This is made possible by the embedding feature of GNNs, which allows a feature produced as a vector description to be learnt by all users in the network. Once the network dynamics are known including the state of users, it would be possible to predict the state users would adopt for a given piece of misinformation diffusion in the network.

3.7.2 Data Sample, Analysis and Validation

The synthetic network was a social network created as a graph in a custom dataset. Two sets of labels (user profiles) were created for the network and a few users were initiated with the profiles. Using this enabled the testing of several model definitions and assumptions made. The Zachary's Karate club dataset used for validation as the real-world network (Zachary, 1977). The links in this network indicate the 78 various encounters between pairs of club members outside the club, whereas the nodes in this network represent the 34 students that participated in the club. There are two different categories of labels for users in the Zachary's network.

With representation learning for User classification being the aim of this model, the Negative Likelihood loss was used as the evaluation metric in the synthetic network and real network. This function is useful when training classification problems with several classes and shows how well the model performed across both datasets. Further data analysis was performed on the loss using a T-Test and an F-Test to check the variance of the datasets.

3.7.3 Simulation Summary

A graph network representative of an OSN was created with a number of nodes using the input module which was used to train the model. SNA was used to establish key nodes in the network for each dataset run. Each dataset was trained over 100 epochs and featured a three-layer neural network as part of the NN module component. The results of the evaluation metric from the synthetic network were compared against that of the real-world network.

The idea was to capture the network information on user interactions based on their created profiles which was implemented using an array of numbers which are called low-dimensional embeddings.

3.8 Third Model - Network Translation Version Three (NetTv3)

An extension over the second model, this model introduced more advanced concepts for simulating user interaction at the micro-level as well as at the macro-level and mirrors the graph-based approach to social network simulation.

3.8.1 Design Architecture and Implementation

▪ ***Task***

The task of the model was multi-class user classification introducing heterogeneous users and connections. A User-Representation learning framework is used.

The design architecture like the previous model is based on the Graph Neural Network (GNN) framework. The previous model framework allowed for the simulation of user interactions with a set of users labelled. Heterogeneous users and connections are introduced allowing for different classes of users and varying connection relationships. The model design architecture is like that of NetTv2 sharing the same type and number of modules with differences in how the module components are structured. Figure 9 shows the internal structure of NetTv3.

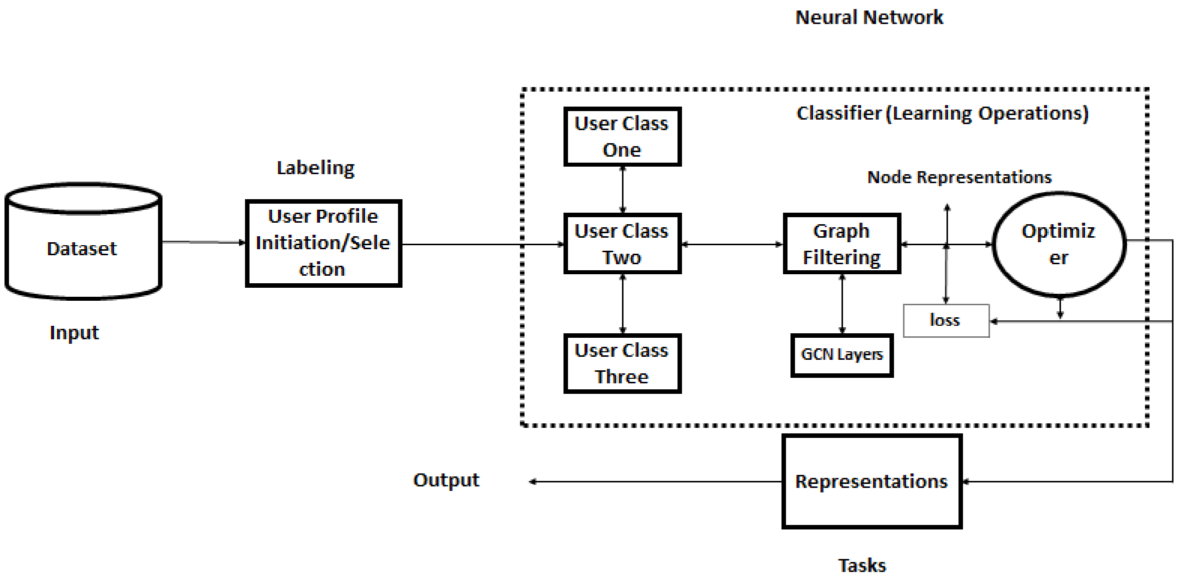

Figure 9. Architecture of NetTv3
A diagram showcasing the workflow of user profile classification using a neural network. Starting with an input dataset, user profiles are initialized and labelled.

The model components of NetTv3 are identical to those of NetTv2 with four modules making up the model. The major difference is in the internal structures and programming implementation of the modules.

▪ ***Model Simulation Requirements***

The hardware and software requirements of NetTv3 mirrors that of NetTv2. NetTv3 features all the libraries used in the previous model in addition to the Pandas library with the model built in the Visual Studio IDE using Python programming language. Pandas is a python based library for data manipulation and analysis (McKinney, 2011). Using Pandas allowed for the importation of custom datasets created in CSV (comma-separated values) format.

▪ ***Research Question and Hypothesis***

This model serves to answer the third research question and the associated hypotheses. The research question seeks to establish the role AI can play in combating misinformation diffusion. To answer this question and prove the hypotheses, it is necessary to simulate the conditions found in a real-world OSN. These networks are heterogeneous in nature, characterised by differing user types and user beliefs. The task of NetTv3 achieves this by implementing multiple user attributes in the GNN model still using an Agent-Based Modelling (ABM) approach. This allows for individuality users/groups of users to have unique attributes assigned to them in the simulation enabling a heterogeneous network in terms of user types. This replicates the diverse user characteristics found in real-world OSNs. Enabling the individuality of users in the network also allows for assumptions unique to this model to be tested.

3.8.2 Data Sample, Analysis and Validation

A custom dataset serves as the input for the synthetic network. Four sets of labels (user profiles) were created for the network and several users were initiated with the profiles. As with NetTv2, the network was implemented as a graph convolution neural network setup (Kipf and Welling, 2017). The dataset is split into a test and training set.

A custom Zachary's Karate Club dataset serves as the first real-world network validation dataset (Zachary, 1977). The cross-entropy loss and classification accuracy are introduced as the primary evaluation metrics. The loss was done for the models at large and between the selected test and training datasets. Accuracy of the test and training datasets was also calculated. The results are evaluated against the baseline semi-supervised classification model (Kipf and Welling, 2017).

3.8.3 Simulation Summary

The model was trained over 100 epochs and featured a three-layer GraphSAGE as part of the NN module component. The results of the evaluation metrics from the synthetic network were compared against that of the real-world network.

3.9 Framework Criteria

The criteria under which all model version's function and definitions of terms and concert relevant to the framework is detailed.

3.9.1 Functional Context of Models

A key factor that was taken into consideration when implementing the research models of social networks was the context of the social network environment and what level of user interactions should exist. To make it simpler for model designers to create context-aware applications, input handling approaches that take context into account are required. However, given the distinctions in characteristics between environmental context and user engagement, this is insufficient. There are two main differences:

▪ The source of user interaction is often single sourced, but the context of a social network environment can have varying definitions that evolve depending on the requirement.

▪ Both user interactions and context call for abstractions to clarify the specifics of internal systems, but context calls for more abstractions because it frequently does not take the form that the model demands for execution.

To overcome these differences, several requirements were identified as being important to the framework to accurately capture and replicate the features of an OSN - environmental features and user features. To summarise, these requirements were:

1). Network Growth

The social network should display a power-law like degree distribution and preferential attachment in its growth. Meeting this requirement would see the synthetic networks created as part of the model simulations as a Barabasi-Albert (BA) scale-free network (Barabási, 2013). This structure of the BA scale-free network is identical to that of OSNs where users in the establishment of links will prefer users with a higher number of existing links.

2). Network Connectedness

The users in the network should have links with other users in the networks. In real-world settings, OSN networks are considered connected as users in the network will establish a connection with other users. The networks used in the model simulations will need to have similar connectedness to that found in OSNs.

3). Network Clustering

The network should demonstrate the ability of users to cluster at the local level. With private beliefs being a focus of the research, it is posited that users in a network will likely form clusters based on their private beliefs. The local clustering is an indication of the embeddedness of single nodes, and it is also used as an indication of the network transitivity. In networks such as social networks, Transitivity reveals the existence of tightly connected clusters of users. Complex networks such as OSNs and notably small-world networks often have a high transitivity and a low diameter.

4). Network Assortativity

Assortative mixing, also known as network assortativity, is the tendency of nodes in a network to attach to other nodes that are like them. It is a description of the relationships between two nodes in the network. Network theorists frequently look at assortativity in terms of a node's degree, however the precise measure of similarity may differ (Newman, 2002). The addition of this requirement to the framework will allow for the creation of models that can simulate a more realistic approximation of the behaviours found in real-world networks. (Newman, 2002). The addition of this requirement to the framework will allow for the creation of models that can simulate a more realistic approximation of the behaviours found in real-world networks. Correlations between nodes of similar degree are often found in the links establishment patterns of many observable networks. In OSNs for example, highly placed users will establish links with other highly placed users. Even common users will prefer to associate with other users that have not just similar beliefs but are also highly placed in the network.

3.9.2 Research Framework Practicality and Assumptions

The research takes the graph-based approach in modelling false information diffusion. The foundation of our model framework is presented as an extension of the IC and LT model and aims to show several practical applications.

1.) Predictability:

The framework aims to develop a model that can accurately make predictions on user connections within an OSN using a user's belief profile and the propagation paths present as means to classify belief systems and detect false information. Identifying a node's belief profile and the possible propagation path of information should allow for early detection of false information diffusion within the network.

2.) Role of Beliefs:

This research aims to show that a node's (user's) belief profile will have an impact on the diffusion within the network across all diffusion stages. It is assumed that if most of the users within an OSN sub-network are of similar belief profile, then the diffusion of information with such a belief profile will most likely transcend the three diffusion regimes - subcritical, critical and supercritical. This highlights the possibility of nodes having selective exposure to information leading to the echo-chamber phenomenon (Sasahara *et al.*, 2020).

3.) Network Externalities:

The framework aims to show that network externalities such as external actors can have influence within an OSN sub-network. If the beliefs of these actors are not distant from an in-group OSN sub-network, such beliefs can be adopted by the sub-network and diffused within the network.

4.) Influence of Bias:

Using the framework, the model aims to show the role of bias in information diffusion. Node (user) interaction under bias can result in nodes being resistant to new information (anchoring bias). This can result in predictable and consistent interaction patterns amongst groups of nodes (stability bias) leading to emergent patterns at the macro level that are inconsistent with the fundamentals. This can be observed in OSNs where subgroups anchored around an influential user with similar worldviews are resistant in adopting differing views.

As part of the research framework context, several assumptions are made with respect to the overall criteria of the models. These assumptions seen in Table 7 provide functional context useful to both user interactions within the network and the network environment.

Assumptions	Descriptions
Active Users	Users that are actively involved in the spread of information in the network.
Passive Users	Users that are not actively involved in the spread of information but adopt the information due to their beliefs.
Phased Users	Users that are neither active nor passive users and are typically found on the periphery of the community clusters
Profile State	An assumption is made that individual belief acts as the preferential schema for adoptions in the model.
Diffusion Threshold	An assumption is made that agents have a personal threshold which influences diffusion within the network and that this threshold is also influenced by the actions of likewise (beliefs) neighbouring nodes in the network.
Belief Adoption	A belief adoption is introduced. It is assumed that mass adoption by users will only occur when neighbouring users have the same belief profile and confirmation bias has already been achieved by the said users
User Bias	A node bias is posited, and it is assumed as being dependent on the internal state (private beliefs) of the user.
Network Dynamics	It is posited that node level changes and system level changes within the network are correlated.
Micro Level Changes	Different agent behaviours are assumed in line with the classes of agents established in the earlier model which in turn will lead to different emergent behaviours at the network level.

Table 7: Assumptions made for Model Frameworks .

3.10 Summary

This chapter presented the research methodology - including the research questions and hypotheses. Informed by the prior background investigations, the framework design architecture described delineates the solutions needed to achieve the research aims and objectives by answering the research questions asked. The procedures for the model frameworks that span the steps from broad assumptions to detailed methods of data collection, analysis, and implementation are described in detail. The various model designs are based on specific tasks in relation to the research questions and requirements to fulfil such tasks.

4. First Model – Network Translation Version One (NetTv1)

4.1 Introduction

Chapter Four presents the first model of this research. Social networks are known to be capable of evolving and adapting to internal changes and network externalities making them dynamic systems. Using simulations to implement the model design framework allows for the observation of the internal structure that are representative of such dynamic systems addressing the first research question together with the associated hypothesis and serves as an explanatory model that seeks to provide a ground truth for the research's model definitions of the user classes and several assumptions made.

Using the model created and simulated, the aim is to understand and visualise the information diffusion process. The model's novel concept of user classes posited will be tested and established using information generated from publicly available social network datasets as well as from an artificially generated network representative of an OSN. The model's details, methodology and simulation results are presented all while testing/proving the hypothesis associated with it as an answer to the first research question.

4.2 Model Overview

The individual interactions of users in a social network often aggregates to collective behaviours. This is because of the interplay between network dynamics and network structure (Namatame and Chen, 2016, p. 9). It has already been established that nodes (individual users) play a social role in the diffusion process (Yang, Tang and Leung, 2015) and that the impact of individual agents in the network performance depends on the network topology and specificities of the dynamics (Namatame and Chen, 2016, p. 9).

The model draws from the Independent Cascade model (IC) (Goldenberg, Libai and Muller, 2001) and the Linear Threshold model (LT) (Granovetter, 1978) with model simulation performed using a synthetic network and real-world networks for result validation. The three classes of agents defined are introduced in this model in a functional context with three assumptions made regarding the functional context of the model.

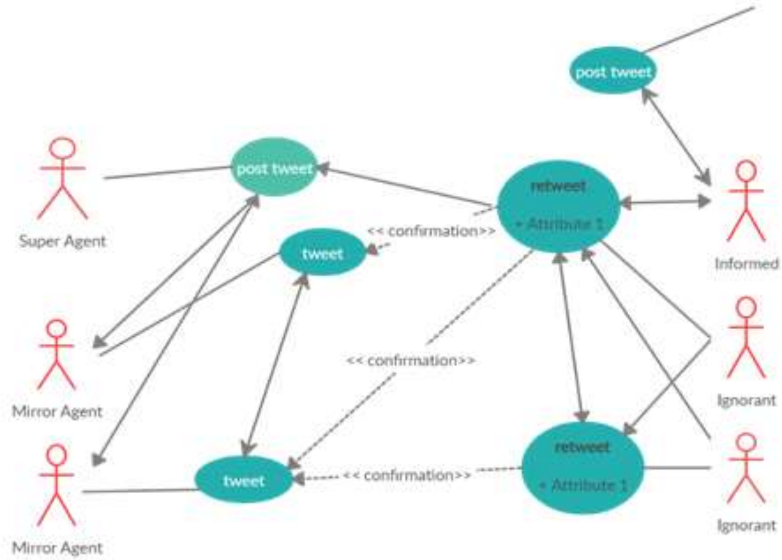

Figure 10. Model Use case
The model use case describes the interaction between user classes in a social network.

Figure 10 shows the use case of an example interaction representative of the Twitter OSN platform. A user posts a tweet on the platform, the tweet is seen and referred (retweeted with or without comments and or liked) to by other high placed users. The followers of users retweet the post if it aligns with their private information with only users whose private beliefs are unbiased checking the veracity of the post.

For interactions at the micro-level of the network, two major features are described with respect to a user's state, namely: user bias and user belief profile. Assumptions are made in terms of the operational context of both features.

▪ The User bias - considers the state of its private information. A node's bias profile can be in only one of two states. The user bias reinforces the user's belief profile.

▪ The user belief profile – the state of a user's belief profile is posited either as being positive or negative. A positive belief profile indicates that a user's private information is highly correct and hence should be able to correctly identify false information often using this private information. A negative belief profile indicates the opposite, and such a user will often endorse false information as their private information is incorrect and can easily be influenced by others.

4.2.1 Problem Statement

With interaction between the different classes of agents being the focus of the model, the LT model's weight parameter is included in the model as a feature of each node. This ensures that in an information spreading scenario both the sender nodes and receiver nodes are considered. OSNs are often made up of sub-networks that cluster together to form one large network.

Within an OSN network, a user - s posts false information j. The information which is visible to the user's followers initiates the diffusion process. A set of agents - r, serve as early adopters of the information and shares j ensuring the longevity of the post allowing the false information to reach the critical stage of diffusion. Users who are classed as ignorants, adopt this false information using repost from other high placed users r of whom they follow as a confirmation. This mass adoption sends the diffusion process into the super-critical stage. The flow of information within the sub-network is represented by a set of edges.

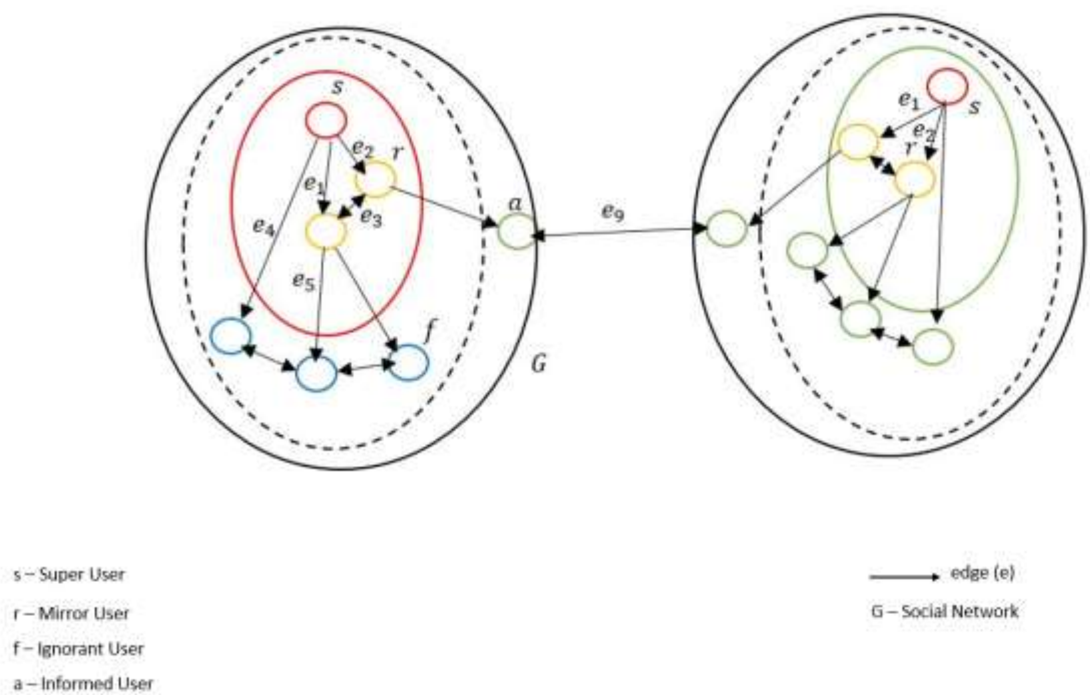

Figure 11. Information Cascade within Networks

Figure 11 shows an information cascade within two sub-networks in an OSN. The classes of users (s, r, f, a) are shown as well as the edges (e) (links) between the agents.

Figure 11 shows an information cascade within two sub-networks in an OSN. The classes of users (s, r, f, a) are shown as well as the edges (e) (links) between the agents. Using a diffusion model of information spreading, the model simulation's goal is to simulate the information diffusion within a network. Considering diffusion within the context of an information cascade, the model seeks to understand the information diffusion life cycle, identify the nodes/agents in the network part of the diffusion cycle and establish if these nodes conform to the model definitions put forward as part of the framework.

4.2.2 Model Assumptions

The model (NetTv1) operates on several assumptions which are shown in Table 8. Using the assumptions described, as the functional context of this model, the model simulation will seek to prove and test the validity of the three-user class - Super, Mirror, Independent (informed and ignorant) types defined in Chapter 3 as part of the research framework as well as if the definitions as put forward regarding these user types are valid.

Assumptions	Description
Assumption 1	Users (agents) will endorse or not endorse false information irrespective of information validity. Agents particularly in the case of the super agents and mirror agents share news to get other agents to adopt their beliefs about the state of the world.
Assumption 2	Only one class of agents have an informative private signal, which is not biased. Without this informative private signal, agents are by default expected to have homogeneity (bias) in the views about the state of the world and would make decisions that conform to these views.
Assumption 3	Mirror agents defined earlier are assumed as agents whose private signal mirrors that of Super agents but are not opinion leaders, hence would be early adopters of the information which the super agents put out.

Table 8: NetTv1 Model Assumptions

Assumption 1 highlights the tendency of Individual Agents (ignorant) to endorse misinformation as a confirmation of their private beliefs. This is in line with research from (Acemoglu, Ozdaglar and Parandehgheibi, 2010) who characterised the evolution of beliefs and quantified the effects of forceful agents on regular agents particularly in terms of consensus information (public opinions). **Assumption 2** is supported by the work done by Marsella et al.

(2004) who show that agents with heterogeneity in their private beliefs have better vision and more diverse global connections than agents with homogeneity in their private beliefs.

Assumption 3 supports definitions made regarding the Mirror user class with agents in this class assumed as agents whose private signal mirrors that of Super agents but are not opinion leaders, hence would be early adopters of the information which the super agents put out. It is also assumed that these agents have mid-level connectivity, mid-influence and occupy a moderate social role and position within the network.

4.3 Model Methodology

The model - Network Translation version one (NetTv1) is composed of three modules that together handle network generation, network analysis and network output. These modules are implemented using an open-source Social Network Analysis (SNA) platform - Social Network Visualizer (SocnetV) which performs all three module functions on its platform. SocNetV has a simple Graphical User Interface (GUI) composed of; the menu, the toolbar, the panels (control and statistics) and the canvas (Kalamaras, 2015) interfaces. The canvas serves as the main area of interaction of the SocnetV platform. Table 9 presents an overview of the model.

Model	Components	Description
NetTv1	Datasets	Synthetic network and real-world (YouTube and Weibo) network.
	Model Components	Input module for network generation.
		Output module for network display.
	Evaluation Metric	Centrality metrics – indegree, information and eigenvector.

Table 9. Model Summary

4.3.1 Input Module – Synthetic Network Generation

The input module handles the network generation routine for the synthetic network using its generation algorithm. The model simulation experiment is done using SocnetV platform, (Kalamaras, 2015). The simulation is performed using a self-generating synthetic network representative of an OSN. A graph network generating algorithm is used to create the scale-free network. The algorithm consists of two growth processes: (1) implicit preferred attachment resulting from following edges from the randomly selected initial contacts, and (2) random attachment. (Toivonen *et al.*, 2006). The local nature of the second process gives rise to high clustering, assortativity and community structure observed in most social networks.

A Barabasi-Albert (BA) network (200 nodes) using a preferential attachment mechanism is generated and serves as the synthetic network in the model simulation. The algorithm starts with the given m^0 connected nodes and is as follows (Barabási, 2013):

▪ **Step 1:** Start with $G_1^{(0)}$, corresponding to an empty graph with no nodes.

▪ **Step 2:** Given $G_1^{(t-1)}$ generate $G_1^{(t)}$ by adding the node $v_t$and a single link between $v_t$and v_i.

In each step a single new node is added, along with m edges to existing nodes. The procedure for the generation of a network in SocnetV using the network generator is as follows:

▪ Enter number of nodes v

A network size of 200 nodes is selected for the synthetic network, $v = 200$.

▪ Select power of preferential attachment p

A new node has the option to join to any other node in the network because preferential attachment is a probabilistic method. However, the likelihood that a new node will connect to a degree-four node is twice as high as that of connecting to a degree-two node. In the simulation, $p = 1$ means a linear regime of preferential attachment (Barabási, 2013). This corresponds to the Barabasi-Albert model as the degree distribution follows the power law.

▪ Select number of initial connected nodes m_0

Two nodes are selected as seed (super users) nodes in the network.

▪ Select number of Edges to add in each step m

Two edges are added for each time step in the network. The network starts as two ordinary nodes connected by an edge. A new node is added for each time step which randomly picks an existing node to connect to, but with some bias in terms of the number of connections of the already existing nodes.

▪ Select the Zero appeal

The zero appeal deals with the initial attractiveness of a node depending on its degree, $a = 1$.

▪ Choose network type - Graph Mode

The graph is chosen as a directed graph. A directed graph is a set of vertices (nodes) connected by edges, with each node having a direction associated with it (Wilson, 2009). Using a directed graph allows for the connection type (incoming or outgoing) to a node to be identified.

Figure 12 shows the network generator of SocnetV which is created from the platform's toolbar. The created network is displayed in the platform's canvas. Using the synthetic network, the aim is to initially test the model specific assumptions as well as the novel user class definitions made.

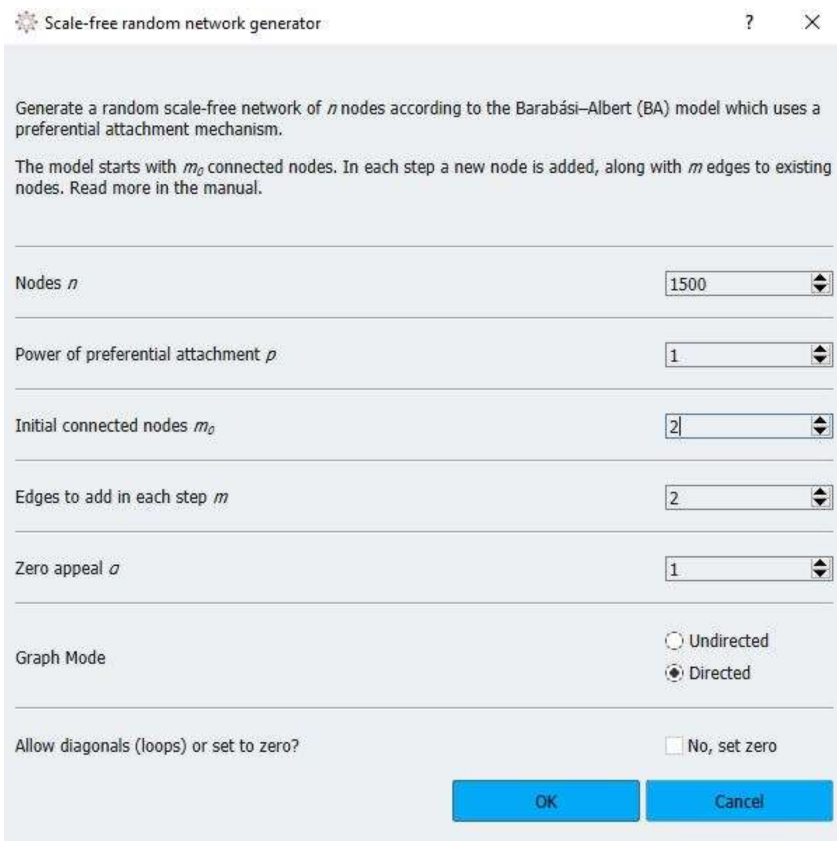

Figure 12. SocnetV Network Generation routine

Figure 12 shows the interface for generating synthetic networks on the SocNetV platform.

4.3.2 Input Module – Real Network

For the real-world network input module handles the network generation routine for the synthetic network using its generation algorithm and for the real-world network using externally loaded datasets. SocnetV supports creating networks from network data loaded in several supported formats. Some of the supported formats include; GraphML (XML for graphs), GML (Geography Markup Language), Pajek and Adjacency Matrix files (*Social Network Visualizer: SocNetV Manual*, 2015). Two datasets (Streamed Graph Datasets, 2014) are used as part of the real-world network - a **YouTube dataset** and a **Weibo dataset**.

The **YouTube dataset** features a directed network with $12647\ users$ (nodes) and 11920 connections (edges). User interactions in the network take place in the form of sharing, comments, and reposts, and user interaction also takes the same form. Sharing and reposts are the diffusion links in the network that differentiate social links amongst users in that they do not require a follow relationship. The **Weibo social network dataset** features 9727 users and 10005 connections with relationships defined in a directional manner hence being a directed network. Likes, comments and shares are the diffusion links in the network. Using this dataset, the aim is to prove the model definitions and validate the findings from the synthetic network with comparisons made between the internal structures of all networks.

4.3.2 Social Network Analysis Module

Analysing graph networks involves using several metrics and tools depending on the data requirements. SocnetV supports SNA using a wealth of tools for analysis and this is used to implement the SNA module. The data analysis focus for this model is on analysis tools that analyse the related connections between nodes in the network. Centrality Metrics which attempt to quantify how central each node is inside the social network satisfy this focus. Using this focus method allows for the simulation results to be observed through the lens of the local graph structure.

The choice of centrality metrics used was informed by the model’s task and the type of graph network used - Directed networks. Using centrality metrics, it is possible to characterise the important local interactions in a graph network. This yields insight into the growth properties and functional properties of the nodes in the network and helps to identify key nodes and relationships in the network.

In terms of the functional context of this model, a centrality-based approach for analysing the internal structure of the networks allows for the consideration of the sequence of contacts in terms of users in the networks and their roles and positions. This will serve to prove or disprove the definitions made with respect to the novel user classes. Three centrality metrics are used to achieve this – “Indegree Centrality”, “Information Centrality” and “Eigenvector Centrality”.

4.3.3 Output Module

The output module handles the display of the networks and other analysis data generated. SocnetV’s canvas is used to realise this module. Networks can be displayed in various layouts and data generated can be saved in easy to access formats like PDFs. With different layout methods, SocNetV provides two different types of network visualisations: By Prominence index and Force-Directed (*Social Network Visualizer: SocNetV Manual*, 2015). With centrality metrics used for analysis, a prominence-based placement is used to visualise the network so that each node takes a position that reflects its centrality status inside the network.

4.4 Results

Rationale: As an explanatory model, the results of the simulations are reviewed to establish a strong foundation for the research project.

The results of the synthetic network simulation and that of the real networks are detailed and analysed. Data analysis focuses on three centrality metrics for the synthetic and real-world networks - the Indegree Centrality (DP) on the nodes, the Information centrality (IC) and Eigenvector Centrality (EC). The metric results are compared against each other for both networks which are further described below.

4.4.1 Synthetic Network

A Barabasi-Albert (BA) network is generated using the preferential attachment mechanism. The network starts with two nodes which serve as the information cascade initiators and features $200\ nodes$ with $355\ edges$. Figure 13 shows the generated network in the canvas. Identical weight values (1) are assigned to all edges in the network, allowing each edge on the nodes to have similar diffusion capacities giving no edge any more importance than the others.

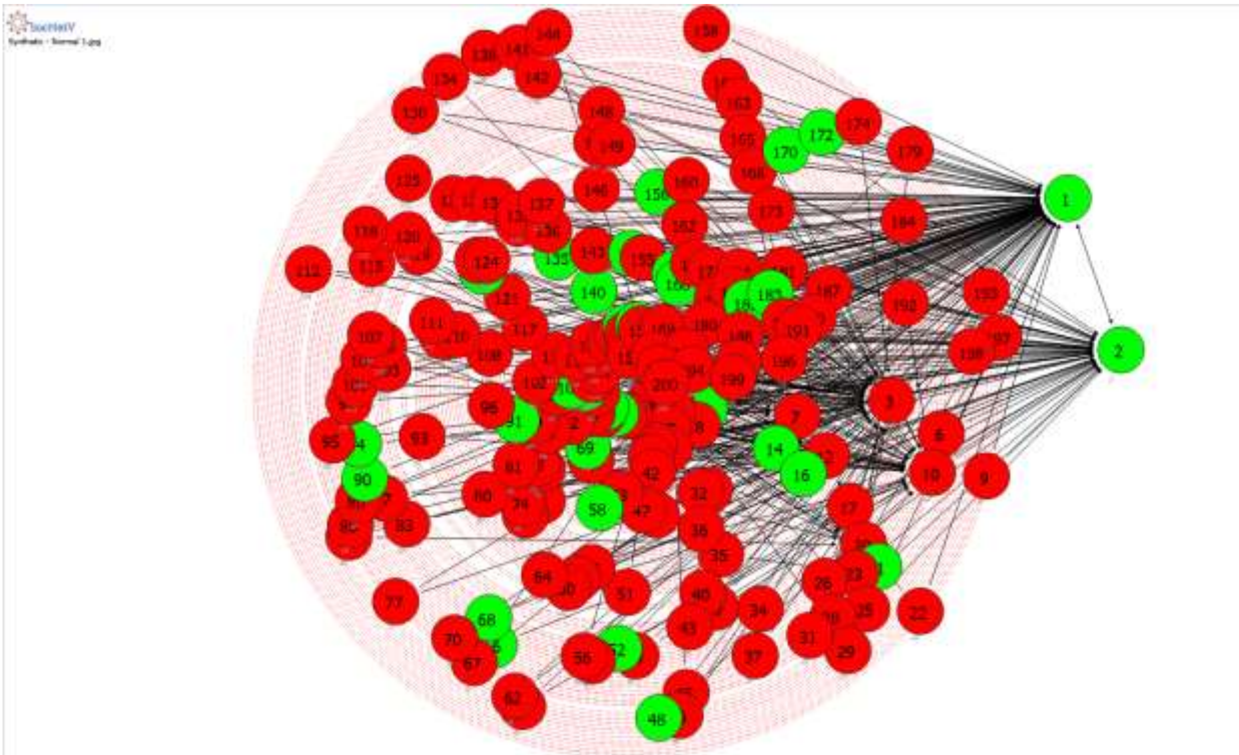

Figure 13. Synthetic Network

Figure 13 showing the generated synthetic network used in the simulation generated by the SocNetV platform.

- **Synthetic Network - Indegree Centrality**

For each node u, the indegree centrality (DP) metric counts the number of inbound arcs at the node (Borgatti and Everett, 2006). The metrics is particularly meaningful in directed graphs as a measure of the status of each node within the network. Nodes with higher DP are considered more prominent among others because they receive more attention from other nodes due to their already existing number of links. The node with the largest index is considered the most prestigious node in the network. Using the Indegree Centrality values for each node, the node classes are identified within the networks based on their earlier stated definitions.

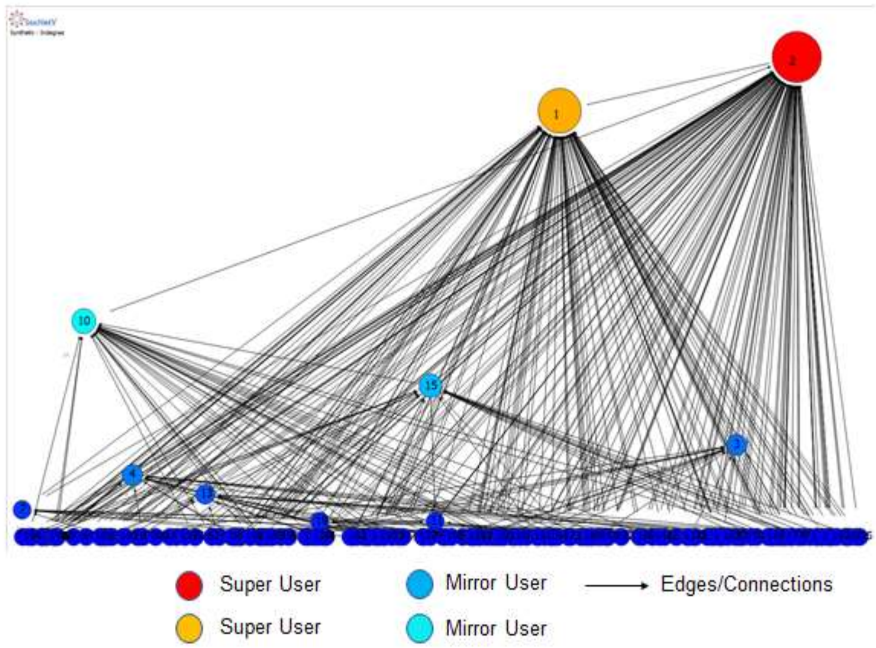

Figure 14. Synthetic Network Indegree Centrality

Figure 14 shows the synthetic network displayed on a layout based on the indegree centrality. Nodes with the largest indegree are seen at positional levels that are higher than that of other nodes in the network.

Two super users (S_1 and S_2) are identified and three mirror users (R_3, R_{10} and R_{15}) are identified as seen in Figure 14. All other users in the network are assumed to be users of the normal user class. It was observed from the edges on the nodes that S_1 and S_2 establish a bi-directional relationship with each other and nodes R_3, R_{10} and R_{15} establish a relationship either both with each other or with S_1 and S_2. This conforms to the definitions put forward with respect to the class of users. Table 10 gives a list of the super and mirror nodes in the network and their number of connections (inbound and outbound). See Appendix A for the Indegree report detailing key nodes by their values.

Nodes	Number of Inbound Edges	Number of Outbound Edges
S_2	133	1
S_1	110	1
R_{10}	32	1
R_{15}	24	2
R_3	14	2

Table 10: List of Identified Key nodes

- **Synthetic Network - Information Centrality**

The Information Centrality (IC) metric measures the information flow through the paths considering strength of edges and distance (*Social Network Visualizer: SocNetV Manual*, 2015). Using this metric, the spread of information is modelled using shortest paths identifying the nodes central to information flow in the network as well as nodes who control the information flows in the network. These are nodes through which information flow through them will result in faster diffusion across the network.

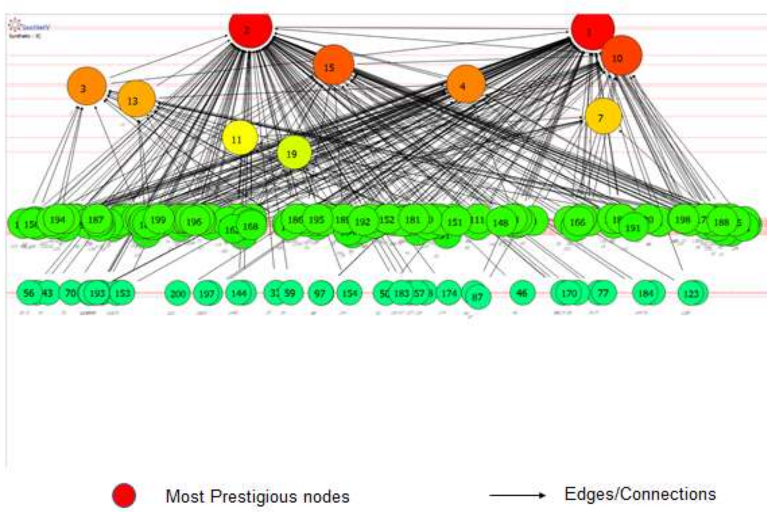

Figure 15. Synthetic network based on Information Centrality

Figure 15 shows the synthetic network's layout based on the information centrality metric. Nodes with that are central to information flow in the network are seen at the top of the network.

Nodes S_2 and S_1 have the largest ICindex values amongst all connected nodes in the network.

Nodes	$\boldsymbol{IC}$ Index
S_2	1.65
S_1	1.64
R_{10}	1.54
R_{15}	1.50
R_3	1.43

Table 11: Most Prestigious nodes based on their IC index

This indicates that these nodes are central to information flow within the network. From Figure 15, the nodes at the top of the graph have the most influence on information flow within the network. With nodes at the bottom having no influence in the network. Table 11 shows a list of the nodes with the highest IC index. These nodes are important to the diffusion of information in the network. See Appendix A for Information Centrality report detailing key nodes by their values.

- **Synthetic Network - Eigenvector Centrality (EC)**

The Eigenvector Centrality (EC) evaluates a node's importance while considering the importance of its neighbours. It is an extension of the Degree Centrality. (*Social Network Visualizer: SocNetV Manual*, 2015). All nodes in the network are given relative ratings based on the idea that connections to high-scoring nodes increase the node's score more than similar connections to low-scoring nodes.

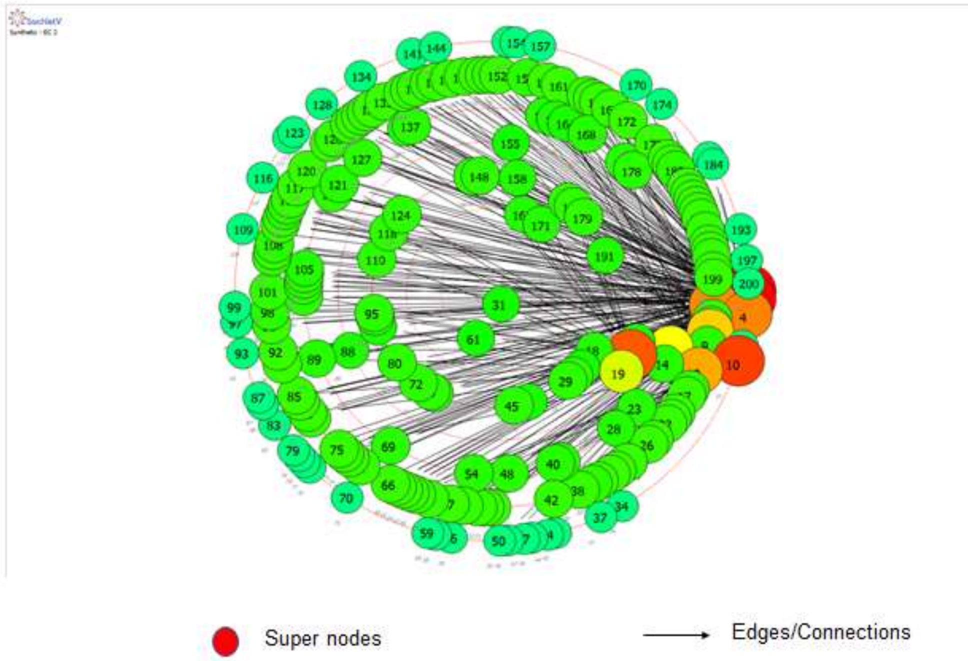

Figure 16. Nodes Based on their EC Index

Figure 16 shows nodes in the network by the prominence index of the Eigenvector Centrality metric.

Figure 16 shows the network displayed by the EC metric and Table 12 shows the top four nodes in terms of the EC index. See Appendix A for Eigenvector Centrality report detailing key nodes by their values.

Nodes	**EC Index**
31	0.21
61	0.18
163	0.16
171	0.13

Table 12: Nodes with the highest EC values

From the index report generated, nodes $31, 61$ and 163 have the highest EC values. This indicates that these nodes are connected to other nodes who themselves are highly connected with high EC values as well and would feature prominently in the flow of information in the network. The transitive influence of nodes is identified showing that node connections in the network are related - nodes will most likely be connected to friendly nodes. The eigenvector also shows that connections can have varying benefits - some connections will be of more value than others.

4.4.2 Real Network - Dataset One

In a Real OSN platform like YouTube, active information spread is observed in the form of sharing and reposts, and user interaction also takes the same form. The first dataset used detailed interactions amongst users in the formation of new user-user links.

- **Real Network – Dataset One Indegree Centrality**

The most connected nodes in the network are identified by the Indegree centrality values. Nodes 993, 2249, 2565, 2609 and 5056 are identified as mirror nodes based on the model definitions. The rest of the nodes in the network are classified as independent users. No nodes are identified as super users.

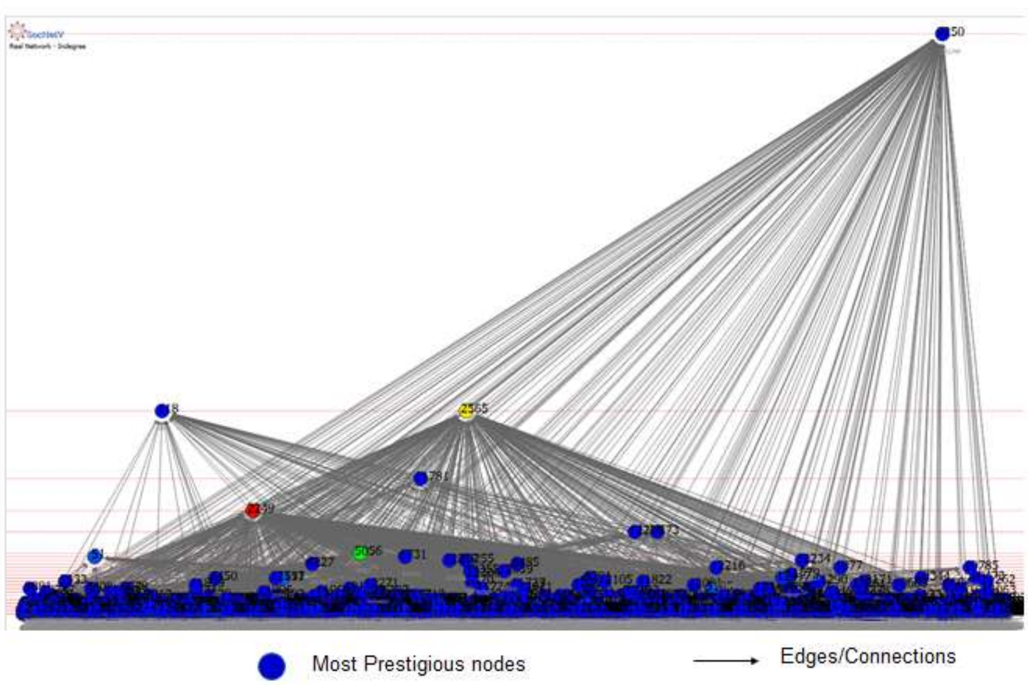

Figure 17. YouTube Network – Indegree

Figure 17 shows the nodes in dataset one network displayed on a layout based on the indegree centrality.

Nodes with the largest indegree are seen at positional levels that are higher than that of other nodes in the network - seen at the top of the network as shown in Figure 17. No super users are identified, however, five mirror users $(R_{6749}, R_{18143}, R_{21265}, R_{21939}\ and\ R_{60921})$ are identified. Table 13 gives a list of the mirror nodes in the network and their number of connections (inbound) (See Appendix A).

Nodes	Class type	Inbound Connections
$993(R_{6749})$	Mirror	164
$2249(R_{18143})$	Mirror	434
$2565(R_{21265})$	Mirror	332
$2609(R_{21939})$	Mirror	130
$5056(R_{60921})$	Mirror	210

Table 13: YouTube Dataset - Indegree Values

- **Real Network – Dataset One Information Centrality**

Used to establish nodes that are central to information flow within the network, Figure 18 shows the users seen at the top of the graph who have the most influence on information flow within the network and such are central to effective information diffusion within the network. Nodes 2249, 2565 and 5056 (R_{18143}, R_{21265} and R_{60921}) have the most influence within the network with node R_{21265} having the highest value. These nodes can be targeted for severing diffusion links within the network.

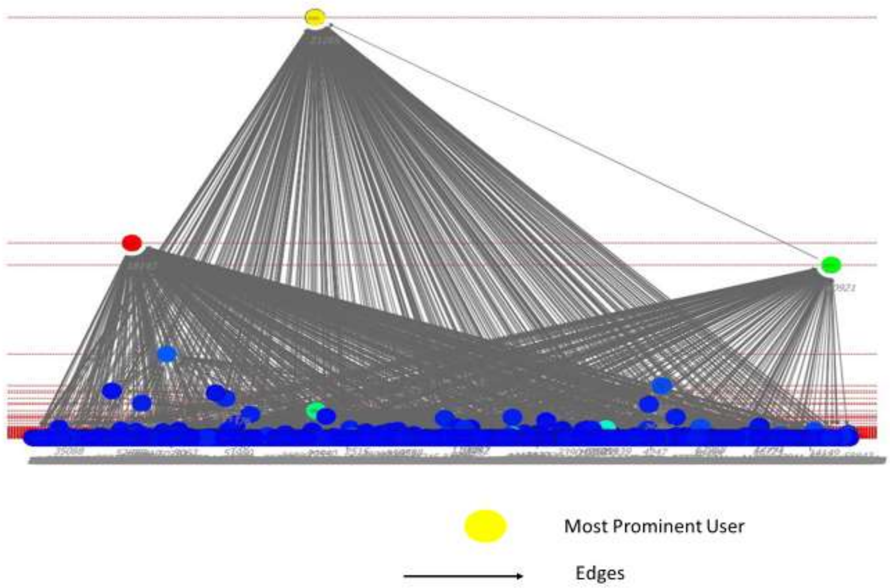

Figure 18. Users based on their IC Index

Figure 18 shows nodes in dataset one displayed on a layout based on their information centrality index.

Nodes	Class type	IC Values
$2249(R_{18143})$	Mirror	1.03
$2565(R_{21265})$	Mirror	1.98
$5056(R_{60921})$	Mirror	1.14

Table 14: YouTube Dataset - Information Centrality Values

Table 14 shows the nodes with the highest IC index in the dataset. Node 2565 is identified as having the highest IC index.

The nodes identified in the Table 14 played a key role in diffusion of information in the network.

- **Real Network – Dataset One Eigenvector Centrality (EC)**

The EC metric of all users in the network is determined. Figure 19 shows nodes in the network on levels by the prominence index of the EC metric. From the index report generated, user 2565 has the highest EC value - 0.899.

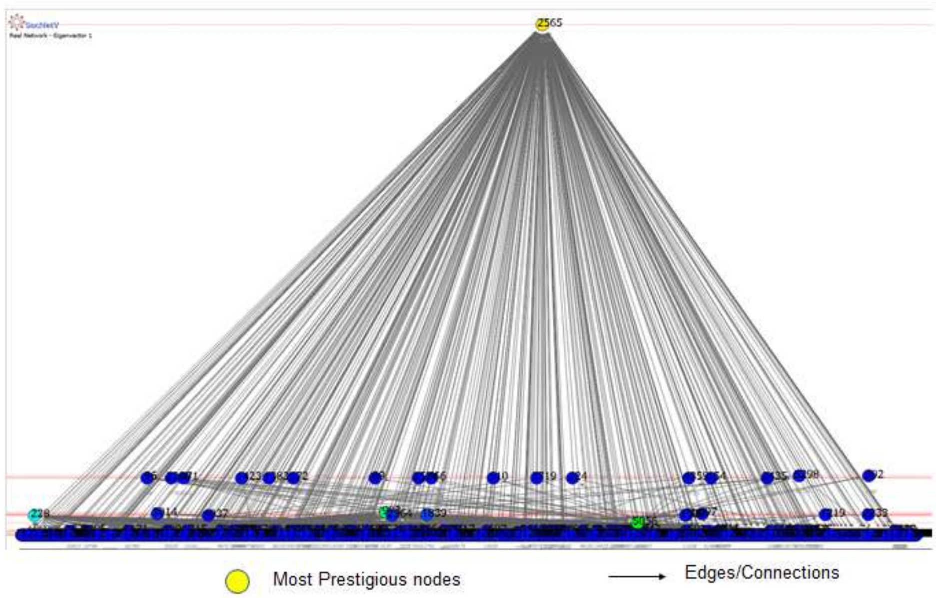

Figure 19. Network Users Based on their EC index.

Figure 19 shows nodes in dataset one displayed on a layout based on their EC index. The node with the highest value is seen at the top.

As with the same scenario for the users identified in the synthetic network, this indicates that this user is connected to other users who themselves are highly connected with high EC values as well. Users that are connected to user 2565 would be related - friendly users and would depict the node as being influential in diffusion in the network.

4.4.3 Real Network - Dataset Two

Dataset two is an anonymized dataset of the Weibo social network platform. The network shows a follow-relation between users in the network based on recommendations for items by highly connected users in the network. Figure 20 shows the generated network which features 9727 users and 10005 connections with relationships defined in a directional manner hence being a directed graph. There is an identical weight value of 1.0 on all edges between users in the network.

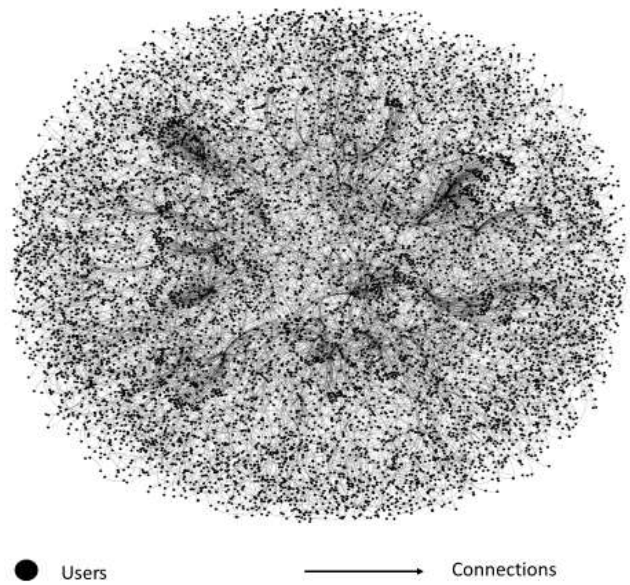

Figure 20. Weibo User Network

A visual representation of the Weibo network, illustrating the intricate web of connections between users. Each dot represents an individual user, while the lines denote their interactions or relationships, highlighting the interconnected nature of the network.

- **Real Network – Dataset Two Indegree Centrality**

One super user is identified S_{663306}, several mirror users are identified from which the top five are selected $(R_{682140}, R_{663931}, R_{1014440}, R_{115241}\ and\ R_{665990})$. The rest of the nodes in the network are classified as normal users.

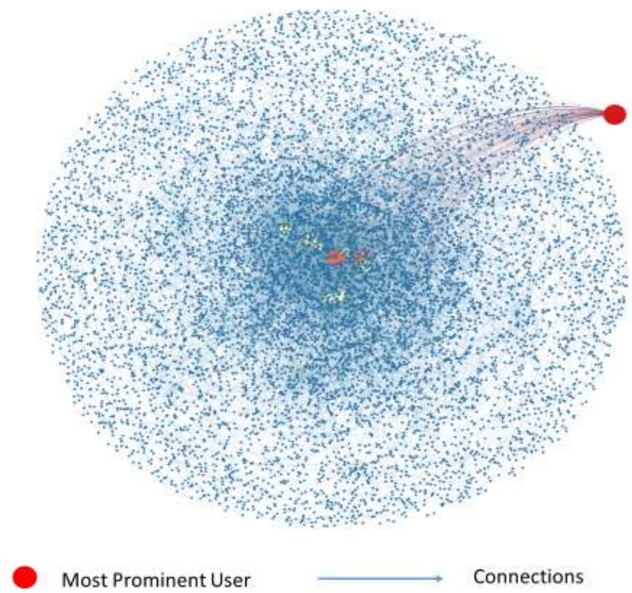

Figure 21. Network Users by In-Degree

Figure 21 shows the Weibo network displayed on a layout based on the in-degree centrality

The user with the largest inbound connection is seen at the top of the network shown in Figure 21 with other highly placed users seen with differing colours in the centre of the network. Table 15 gives a list of the users in the network and their number of connections (inbound).

Nodes	Class Type	Inbound Connections
S_{663306}	Super	889
R_{682140}	Mirror	80
R_{663931}	Mirror	74
$R_{1014440}$	Mirror	54
R_{115241}	Mirror	48
R_{665990}	Mirror	46

Table 15: Weibo Dataset - In-Degree Values

- **Real Network – Dataset Two Information Centrality**

From Figure 22, the users with the more receptive to information flow through them in the network are seen in the network. These users will have an influence on information flow within the network and such are central to effective information diffusion within the network. Most users in the network are active participants in the information diffusion process as seen in the large number of users (blue) with a centrality value of 1.0.

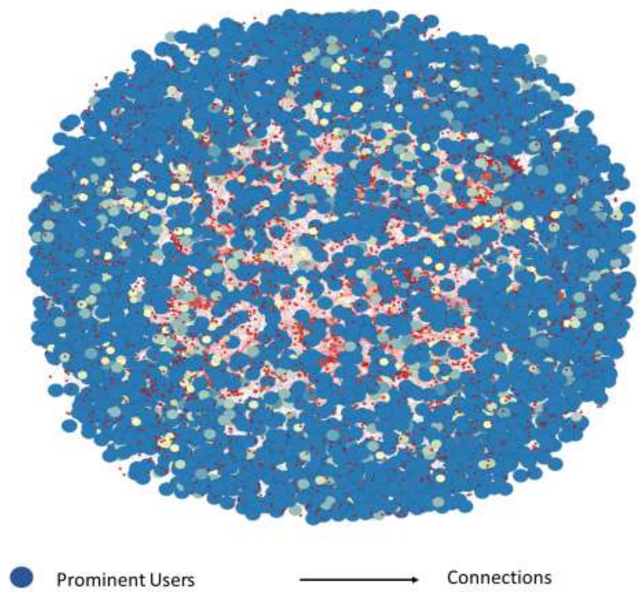

Figure 22. Users based on their IC index.

In Figure 22, users active in information diffusion in the network are seen denoted by their different colours.

Table 16 shows a list of a number of users with the highest IC index in the dataset.

Nodes	**IC Values**
R_{215666}	1.0
$R_{1143204}$	1.0
R_{607558}	1.0
$R_{1141879}$	1.0

Table 16: Top Users based on their IC values

- **Real Network – Dataset Two Eigenvector Centrality (EC)**

Figure 23 shows nodes in the network in a layout by the prominence index of the EC metric. Other prominent users are also seen as differentiated by their colours. From the index report generated, node 1014440 has the highest EC value with a value of 1.0 .

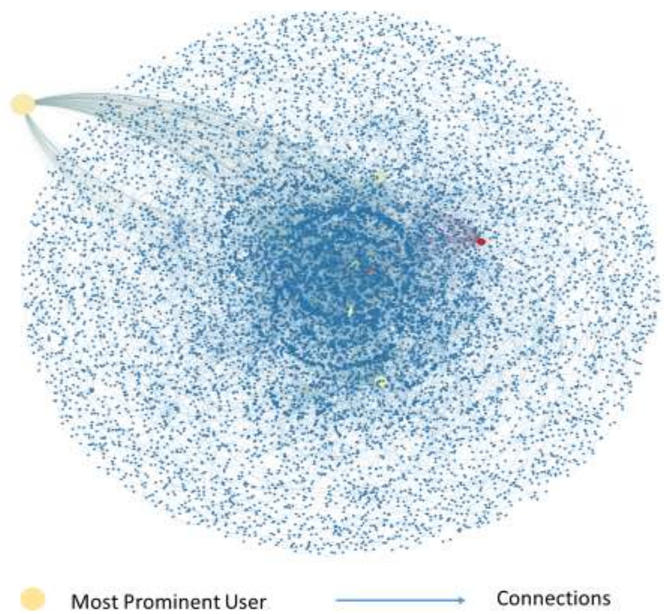

Figure 23. Users based on their EC Index.

In Figure 23, the most prominent user based on the EC index is seen at the edge of the network.

Node	EC Value
$R_{1014440}$	1.0
R_{105478}	0.499623
R_{108465}	0.493745
R_{13568}	0.471244
$R_{1120568}$	0.379091

Table 17: Users in the Network based on their EC values

Table 17 shows the top five nodes in the network based on their EC values. Like the synthetic network and first real network, this indicates that this node is connected to important nodes in the network. Node connections would also be related - friendly nodes establishing connections with each other and depicting the nodes identified in the table as being influential in the network.

4.4.4 Discussion of Model Results

The synthetic network was used to validate the model definition of user types in a network set along various classes posited as part of the model definitions with their set parameters. The results from the simulation confirmed that in a randomly generated network, there are nodes whose characteristics are identical as set in the definitions of the various classes of the model. Hence these nodes can be classified into one of the set classes. The choice of the number of nodes (200) used in the synthetic network is limited by computational resources. However, this was enough to establish node relationships in a typical OSN. the centrality metrics provided the following information:

▪ Indegree Centrality

Using this measure, the aim was to establish the definitions given to users in the super and mirror class in terms of their number of connections in the network. The results from the in-degree centrality as seen in Table 3 shows that users in a network can be grouped into three categories based on their number of inbound connections. Users with high number of inbound connections as seen in S_1 and S_2 (group one), users with mid number of inbound connections - R_3, R_{10} and R_{15} (group two) and users defined aby a small number of inbound connections (group three). This assertion is confirmed by the results shown in Table 8 and provides a ground truth to the definitions given to the user classes.

▪ Information Centrality

Using the information metric, the aim was to establish the definitions given to users in the super and mirror class in terms of their roles as opinion leaders and early adopters of information in the network and their effects on overall diffusion in the network. The information centrality indexes as seen in Table 4 and Table 7 show that super and mirror users in the network are key information sources, and this enables them to have major influence on diffusion within the network with other users acting as information conductors by offering least resistance to information flow. The higher the index number, the more importance the user holds as an information source.

▪ Eigenvector Centrality

The EC highlights connections from important nodes (as identified by degree centrality) as having more value than connections from unimportant nodes. Nodes gain importance based on the number of important nodes to which they are connected. Using this metric, the aim was to establish the definitions given to users in terms of their connections with other users and roles in the network. The results shown in Table 5 show the effects being connected to highly placed users by other regular users has on other users as these users become more attractive to other users of the same class. Users with important neighbours via connections will have more influence than users with non-important neighbours.

For the real network, the two datasets used were representative of social networks and were used to validate the results of and observations made in the synthetic network. The results of the Real Network dataset simulations across both datasets mirrors that of the synthetic network simulation. As with the synthetic network simulation, in the real network datasets simulation, user characteristics were observed to mirror that of the set parameters for the classes of agents created as part of the initial model. For the **first dataset (YouTube),** mirror users and independent users made up the entire network as no super-agent was identified. In the **second dataset (Weibo)**, all three user classes were identified in the network.

The number of nodes across the real network datasets was also limited by computational resources but was significantly (about 50x) higher than that of the synthetic network. Identical centrality metrics were used in data collection and analysis across the simulations. With more computational resources the number of nodes can be significantly scaled up to several hundred thousand which should be much more reflective of node relationships in an OSN. The results of such a simulation are expected to mirror that of the initial artificial network simulation.

The following observations were also made from the results across both network types:

▪ Super users were observed to have by far the greatest number of connections in the network establishing the prominence and importance to the internal dynamics of the networks.

▪ Super users were also observed to be central to information flow within the network as seen in their IC values. This indicates that such users will have influence over diffusion events in the network.

▪ Mirror users were observed to be connected to super users of other mirror users. This satisfies the research's model definition assumptions that such users will either have a unidirectional or bidirectional relation with super/mirror users.

▪ Partial transitivity can be inferred from the results - node connections are observed to be in a pattern that depicts connections amongst friendly nodes.

Referencing the research questions and hypotheses, the identification of user classes in the networks simulated offer a validation to the solution proffered for Question one and to the hypothesis. Considering the assumptions made and referring to the user classes defined, it is established across the simulations that users in a social network play differing roles in the diffusion process and that these roles will affect their belief and interactions with other users. The results helped to prove the first hypothesis put forward as an answer to the first research question and provided a validation for the research while also satisfying one of the second research objective.

4.5 Conclusion

The model – NetTv1, simulated information diffusion on a synthetic network amongst several established classes of users with the aim of understanding their roles in the network and visualising the formation of edges and the diffusion process. Computing limitations limited the number of nodes used in the synthetic network and the size of the datasets used in the real networks. The focus of the synthetic network was to understand the social network formation process while testing the hypothesis in the concept of user classes by deducing nodes key to the diffusion process using several centrality metrics (in-degree, eigenvector, and information). Results from the simulations mirrored each other in line with the model description with all users of the various user classes (super, mirror and independent) identified in the networks. The results helped to prove the second hypothesis put forward as an answer to the first research question and provides a validation for the research.

As a progression from the current model - NetTv1, NetTv2 will focus on introducing trainable agents using neural networks and information classification for the purpose of identifying the interactions between users in the presence of differing belief types. Bias will be introduced to see how users interact in the presence of different degrees of bias amongst them and how this affects information diffusion. The aim will be to identify the diffusion thresholds for information in the presence of trained agents and forceful agents and if this will be enough to trigger the diffusion to the super-critical stage. Contributing to the overall aim of the research of developing an architecture to accurately classify users in an OSN based on their beliefs.

5. Second Model – Network Translation Version Two (NetTv2)

5.1 Introduction

Chapter five presents the second model of this research. In relation to the project specifics, the model seeks to answer the second research question asked by simulating how users in a social network will interact with differing sets of beliefs establishing the role that beliefs have in the diffusion and hence adoption of information.

Using networks implemented in a simulation, the functional context of the model learns the internal states of users in a network in the presence of various belief systems. Having already established the veracity of the model's novel user classes posited, **to establish the connection between beliefs and information adoption, a network scenario where users seek to learn the state in a network - such as information being diffused through the network, is simulated**. The effects of public interactions observed in neighbouring users and user structure on users in the network is also observed. The connections the network used are defined in a manner that allows for the classification of propagation paths as either biased or unbiased based on the beliefs adopted.

5.2 Model Overview

Previous research done within the context of this project established that users in an OSN tend to endorse claims that adhere to their system of beliefs and to ignore dissenting information (Usó-Doménech and Nescolarde-Selva, 2015). Users also form links with other users with similar beliefs fostering the aggregation of like-minded people where debates tend to enforce group polarisation. Confirmation bias plays a pivotal role in viral phenomena in information diffusion within social networks as it reinforces user beliefs and strengthens existing connections (Del Vicario *et al.*, 2017).

The second model extends the first model by adopting a graph neural network (GNN) based framework for learning a user's internal state in an OSN. Through user classification, a better understanding and characterization of user beliefs and their roles in diffusion within an OSN is made possible. Belief profiles are introduced as node embeddings (label propagation) to see how nodes interact in the presence of differing beliefs amongst them and how it affects the adoptions by nodes in the network.

The diagram in Figure 24 shows the basic GNN process for a network. Following from the graph representation of the problem, for each node in the graph, information is embedded in a vector getting an initial representation of itself – these functions as the input to the GNN. The GNN trains this input and produces an output. The output is the same graph, each node also has a vector representation but in this case the vectors have information about how they feature within the graph. This output can then be passed to perform a specific task such as a classification task that produces a representation of each node in the network.

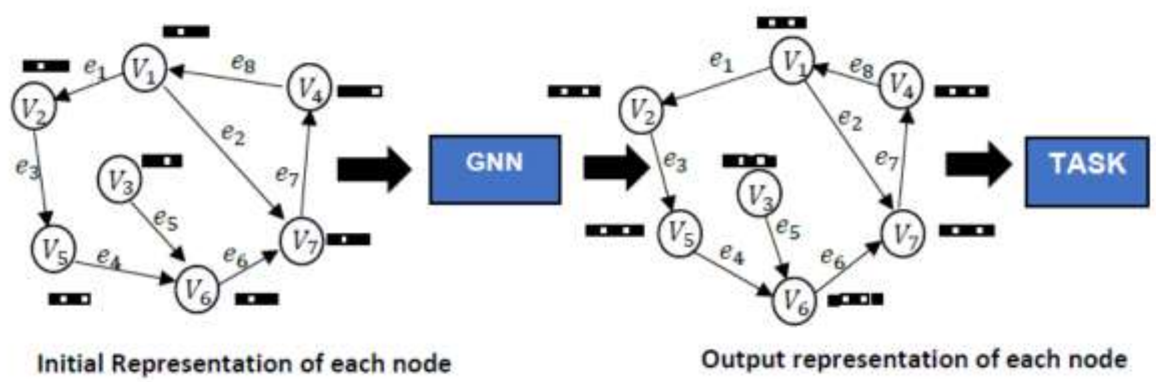

Figure 24. GNN Process in a Network

Figure 24 shows a graph representation of a social network and how GNNs can be used to simulate changes in states for each user in the network. Each user (v) in the graph knows about neighbouring users through their links (e).

The GNN process above can be further described within the context of this project. Using a graph to represent an OSN, users in the network are represented as nodes (V) in the graph and their links (e). For each user, there exists a set of features local to them. These features define the internal state of each user in the network. Using the Twitter OSN platform as an example, at every point in time during their interactions, the users in the network all have an initial state which is active in their interactions with other users in the network. For each tweet they interact with, users because of their initial state already hold a view on the content of such tweets and observe the public signals in the form of interactions and precisions about the subject of interest from other users.

At every time, the public signal in the network generated from neighbouring users is updated. Each user encounters this signal and interprets it, computing their opinions and precision from it. This allows users to set their final opinions on whether to adopt/endorse the belief/information contained in the tweet through a combination of their initial state with the interpreted signal and the opinions and precisions of neighbours' users encountered at the various time steps. In such a scenario, it is assumed that connectivity among users is fixed, informed by their preferences and strong bias toward these preferences. It is also assumed that all users can see all the other users either through a directed or undirected path in the platform.

Several assumptions put forward for the model framework are taken up here with respect to the functional context of this model. As a practical application, the model aims to show that a node's belief profile will have an impact on the diffusion within the network across all diffusion stages. It is assumed that if most of the users within an OSN sub-network are of similar belief profile, then the adoption of information between users with such a belief profile will have the information spread transcend the three diffusion regimes (sub-critical, critical and supercritical). This highlights the possibility of nodes having selective exposure to information leading to the echo-chamber phenomenon (Sasahara *et al.*, 2020). Irrational information diffusion is also prevalent within such a scenario as the nodes can establish a web of self-supporting links based on the already existing belief system enabling the diffusion process.

5.2.1 Problem Statement

In social networks, a node's importance within the network is often in correlation with its degree (Albert and Barabasi, 2002). The nodes with larger degrees are likely to be very influential within an OSN and hence key to the diffusion process. The node with the largest degree will be the hub node in such a network, central to all activities in the network. Misinformation diffusion within such networks presents several dilemmas. Referencing the research questions one and two, the problem of classifying nodes (users) involved in the spread of misinformation within OSN clusters that make up such networks is one of them. To address this, the model introduces user representation learning for user-level classification using GNNs. GNNs bring the power of deep neural networks on relational data which are predominant in most applications like graph networks like social networks (Wang *et al.*, 2019). The model is based on spatial graph convolutional neural networks, first introduced in Micheli (2009). The model is implemented as a graph-based convolution neural network setup; where the input into the network is a graph represented as a matrix of user features.

In the baseline implementation, three node types (super (s) and mirror (r) nodes, ignorant (f) nodes and informed (a) nodes) are assumed and created within the graph. Node profiles are available for the node types, and these constitute the input into GNN. The aim of the model is to simulate user interaction in a network using a network structure and parameters that allow for the testing of the model's hypotheses. Using the said interaction to learn the states of users in the network through representation learning operations.

Statement - Given a graph, $G = (V, E)$, where V is the set of users and $E = \{(v_i, v_j)(v_i, v_j \in V)\}$, is a set of edges between two users (i, j), part of the set of users. The neighbourhood of user v is defined as $N(v) = \{u \in V | (v, u) \in E\}$. For the graph, there exists user attributes X, where $X \in \mathbb{R}^{n \times d}$ serves as the user feature matrix where n is the number of nodes and d is the dimension of a node feature vector. For each user, there is a set of defining features created as embedded vector representations. This indicates their current state for each time step with $x_v \in \mathbb{R}^d$ representing the feature vector of a node v with d of the feature vector for the node. Table 18 summarises the key notations in the problem statement.

Notation	Definition
G	Graph
V	Set of users
E	Set of edges between users
$N(v) = \{u \in V \mid (v, u) \in E\}$.	Neighbourhood of user v
n	Number of nodes
d	Dimension of a node feature vector
$X \in \mathbb{R}^{n \times d}$	Node feature matrix

Table 18: Problem Statement Notations

5.3 Methodology

Network Translation (NetTv2) is presented as an extension of NetTv1. Introducing GNNs, the model composed of four modules (input module, a labelling module, a neural network (NN) module and an output module) is implemented using a synthetic network with a real-world network used for validation. The network is implemented on the basis of Deep Graph Library (DGL), a Python package built for easy implementation of graph neural network model family, on top of other existing DL frameworks (Wang *et al.*, 2019). Using this framework allowed for the simulation of user interactions in the presence of user labels analogous to beliefs in this research in what is termed user-level classification which uses graph filtering to generate user representations for each user/node in the network. This allows for a better understanding and characterization of user beliefs and their roles in diffusion within an OSN. Model simulation is done in two parts: a synthetic network and a real-world network.

For training, model is simulated over the same period for each network type with the results of the loss over the period computed for each network using an evaluation metric. Table 19 presents a summary of NetTv2 model structure.

Model	Components	Description
NetTv2	Model Components	Input module for network generation
		Labelling module for dataset labelling
		Network module for training the network
		Output module for evaluating and displaying the network
	Datasets	Synthetic network (16 nodes) and real-world (Zachary's Karate Club – 34 nodes) network
	Neural network	3-layer graph convolution network (GCN)
	Optimizer	Adam Optimizer
	Period	100 epochs
	Evaluation Metric	Negative Likelihood loss

Table 19: Overview of NetTv2 Model components

5.3.1 NetTv2 - Synthetic Network

A Graph representation of the problem is created as a synthetic network. The network is initiated as a directed weighted graph with nodes and edges. As a directed graph, the relationships between the edges on the nodes have a well-defined direction and order. Given a network $G = (V, E)$, where V is the set of nodes and E is the set of edges connecting the nodes. On top of various DL frameworks already in use, the network is developed using deep graph library (DGL), a Python module designed for simple construction of graph neural network model family (Wang *et al.*, 2019).

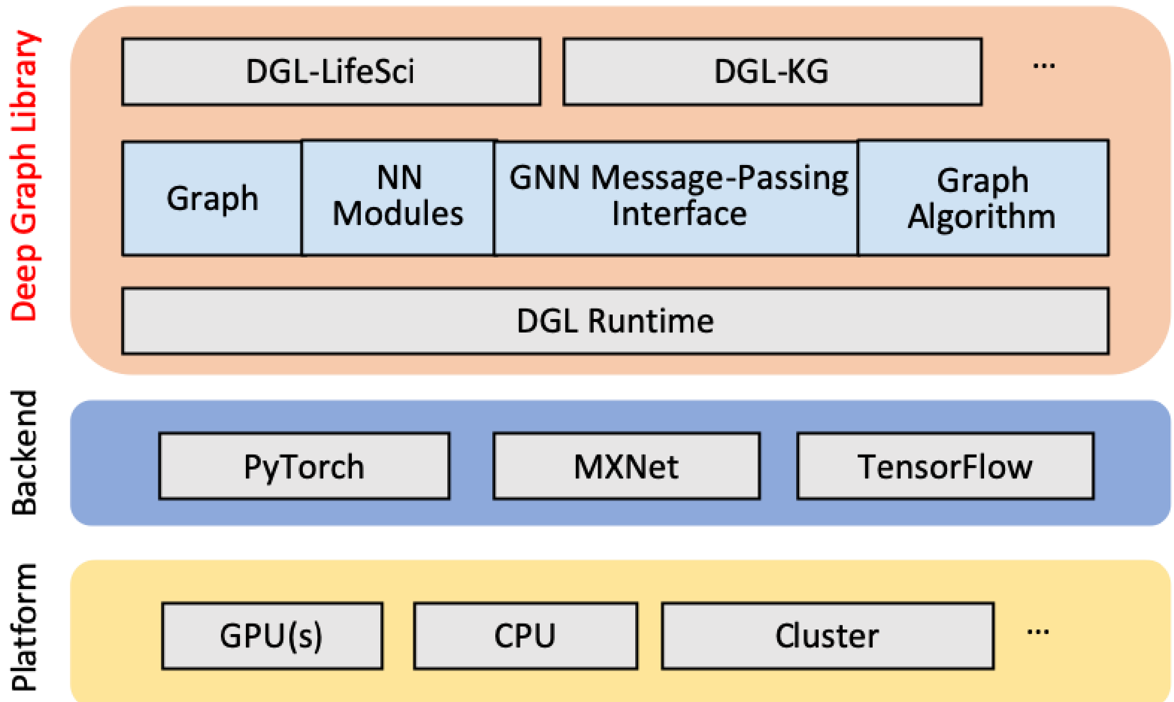

Figure 25. DGL Architecture (Wang *et al.*, 2019)
A layered architecture of the Deep Graph Library (DGL), showcasing its integration with popular deep learning frameworks and its deployment across different computational resources.

Figure 25 shows the overview of the stacks of DGL. The platform layer consists of the hardware, the backend layer consists of the frameworks and the top layer consists of the DGL core. The model programming is implemented as a graph-centric programming abstraction with the model's software codes written in python language.

- **Input Module**

The input module handles the network creation and display and for the synthetic network. The networkX library and the deep graph library (DGL), part of libraries that make up the model's software component, handles the network creation, initiation, and display process. These are imported as part of the library packages required for a successful program build in the IDE. The synthetic network is implemented as a graph-based convolution neural (GCN) network setup. Given a graph $G = (V, E)$ representative of an OSN, the input graph is represented as matrix of user features, $X \in \mathbb{R}^{N \times F}$, where N is the number of users and F is the number of input features for each node and $A \in \mathbb{R}^{N \times N}$ represents an adjacency matrix for the graph G.

For the synthetic network, the network is created using the network initiation procedure. A network (16 nodes) is created using the network creation and initiation procedures (Appendix B) and serves as the dataset. The node edge connections in the graph as shown in Table 20 were specifically defined to allow for the testing of the differing belief adoption by users with already existing propagation paths. The graph is a featureless graph and relies on the use of an embedding matrix to add features to the nodes in the graph.

Users	Class	Outdegree	Nodes Connection
0	Super	0	$2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15$
1	Super	0	$2, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15$
3	Mirror	1	$2, 4, 5, 15$
6	Mirror	2	$7, 8, 10, 14$
12	Mirror	2	$8, 9, 10, 11, 13, 15$

Table 20: Node Connections in Synthetic Network

Figure 26 shows a visual of the network with nodes plotted pre-training of the synthetic network. The nodes with the largest connections are seen at the edges of the network (See Appendix B for algorithm for network creation and display).

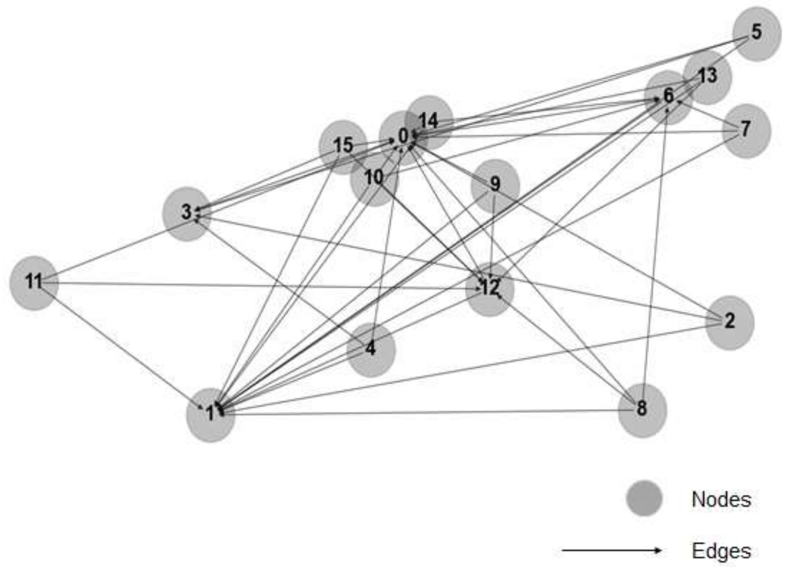

Figure 26. Pre-trained Synthetic network
A visual representation of a network graph, illustrating the interconnected relationships between numbered nodes through connecting edges.

- **<u>Labelling Module</u>**

This module handles the creation and definition of node labels analogous to belief profiles in this research. The node profiling defines features that are unique to nodes in the graph. They function as features that are unique to each node in the network. The profiles are implemented as an embedded vector representation for each node. A graph representation of the problem is presented where an initial representation of each node is encoded as Distributed Vector Representations (DVR) – an embedding containing information for each node. This provides an initial representation of each node through features that gives an insight as to how each node/node class features in the network and are associated with nodes for network training.

Embeddings are fundamental to doing classification tasks in graph networks. They make categorical variables less dimensional and meaningfully reflect categories in the converted space (Gu *et al.*, 2021). The idea behind embeddings is to embed the nodes of the graph into low dimensional space so that these embeddings capture essential specific information tasks of the nodes and use them to train off-the-self classifiers. With respect to the model, the specific information tasks being captured are the node belief profiles with ANNs used as the classifiers. They use both the structure of the graph network and the features of the nodes and their edges.

The embeddings serve as the input into the network and find the nearest neighbour nodes in the embedding space - clustering related categories of nodes together.

During training, for each time step, the node learns about its neighbours and their profiles in the form of Neural message passing (exchange of information by adjacent nodes), with the distance increasing for each time step. For the simulation purpose, it is assumed that nodes in the network in terms of their features are classified based on their beliefs. Hence a node with a negative belief profile is considered as being biased and will share/adopt false information within the network.

For users initiated with profile state 2, their belief state is biased. Such users are unable to verify the veracity of information and would adopt information based on it matching their beliefs. As a part of the node representation learning task, a percentage of the nodes in the network are labelled with the profiles created. The node profiles are initiated in one of two states - 0 (profile state 1) and 1 (profile state 2) for nodes classed as super nodes and for nodes classed as mirror nodes. Table 21 shows the nodes initiated and their profile types.

Nodes	**Belief Profile State** **(state 1 = non-biased, state 2 = biased)**	**Initiated value**
$s0$	state 2	1
$s1$	state 1	0
$r3$	state 1	0
$r6$	state 1	0
$r12$	state 2	1

Table 21: Selected nodes and their Profiles

In the scope of the model simulation, the node classes are assigned learnable embedding vectors which post model training create representation of nodes in the network where similar nodes are closer to each other (See Appendix B for full python codes and DGL Library functionality).

- **Neural Network Module**

This module handles the network training and classification/prediction tasks. The neural network architecture and parameters are defined and created in this module. To perform node-level classification a Graph Convolution Network (GCN) is used. The GCN model architectures and hyperparameters follow the design from the GCN original model (Kipf and Welling, 2017). The basic GCN framework is defined below:

▪ At layer l, each node v_i^l carries a feature vector h_i^l.

▪ Each layer of the GCN tries to aggregate the features from u_i^l where u_i are neighbourhood nodes to v into the next layer representation at v_i^{l+1}. This is followed by an affine transformation with some non-linearity – the nodes can have changes in their state while still preserving network properties in terms of links established.

The framework fits with the neural message passing paradigm (Gilmer *et al.*, 2017) – each node will update its own features with information sent from neighbouring nodes. Figure 27 displays a demonstration of this process. Message passing in the graph network is synchronous - each node in the graph knows about itself, then learns about its neighbours. Where for the node update, $f(.)$ – a parametric function serving propagation.

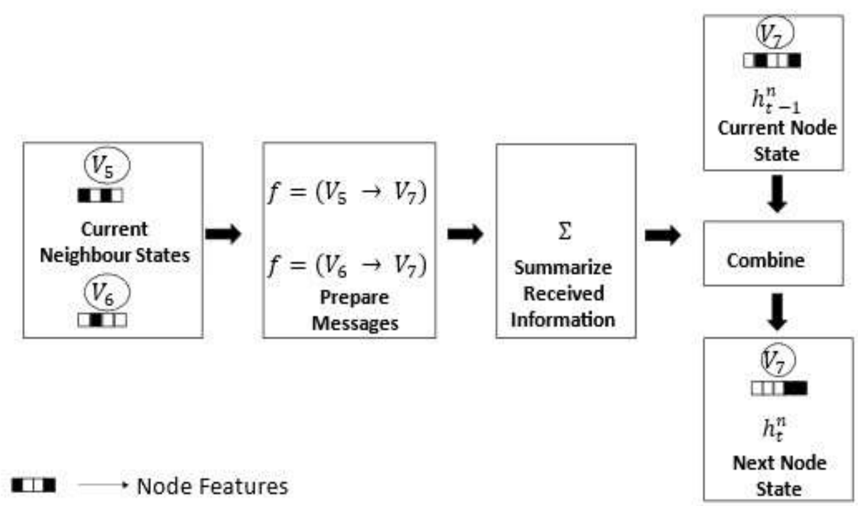

Figure 27. Neural Message Passing Procedure
A schematic flowchart illustrating the process of message passing between nodes in a graph, detailing the steps from current neighbour states to the next node state using functions, summarisation, and combination.

In the standard GNN, users will exist as recurrent units and their connections as simple feed-forward neural networks. Each user initially knows about itself then its neighbours with the distance, $d = 1$ increasing by each time step.

For the GCN variant, node classification aims to learn the node latent features $\boldsymbol{h}_{v_i} \in \mathbb{R}^{\boldsymbol{d}}$, for a node v_i. This operation is analogous to graph filtering, and it refines the features of users in the network. Referencing a user in the network, the neighbouring users pass their messages (embeddings) using the propagation rule $\boldsymbol{f}$ shown in equation 1 below.

$$\boldsymbol{f} = (\boldsymbol{H^i}, \boldsymbol{A}) = \boldsymbol{\sigma}(\boldsymbol{AH^iW^i}) \qquad (2)$$

Where $\boldsymbol{W^i}$ is the weight matrix for layer $\boldsymbol{i}$ and $\boldsymbol{\sigma}$ is a Rectified linear (ReLU) activation function. The hidden layer is represented as a matrix of the users hidden features, $\boldsymbol{H} \in \mathbb{R}^{\boldsymbol{N} \times \boldsymbol{F}}$ where N is the number of nodes and $\boldsymbol{F}$ is the number of input features for each node. These features are combined within the network at each layer using the propagation rule $\boldsymbol{f}$ to create the features for the subsequent layer. Each node will include a self-loop to ensure that a representation of its features is included in the features aggregate. Figure 28 shows the message passing for a node at a given time step.

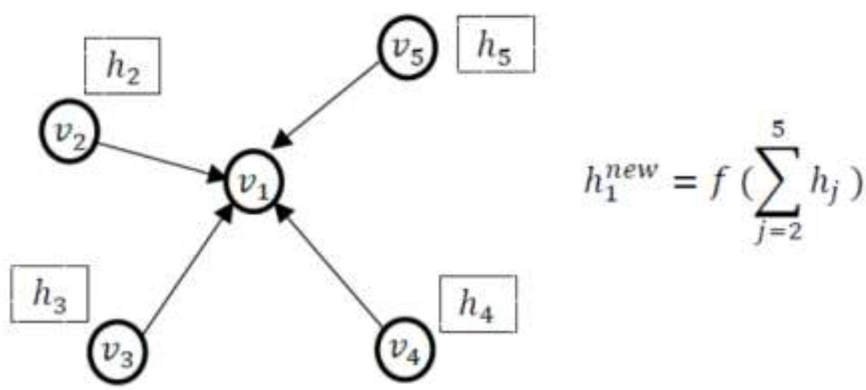

Figure 28. Message Passing amongst nodes in the graph.
A diagram demonstrating the aggregation of features from neighbouring nodes to compute the new state of node v1, using a summation function over the feature vectors h_{j}.

Using the GCN allows the learning operation to be performed in the model. User-level representation learning operation is the learning operation performed by the GCN within the model. User-level representation learning operation refines the features of users in the network. The output for such operations relates to user (node) level classification tasks. This operation will be done by the convolution layer of the model, which is responsible for extracting high-level user representations, transforming, and aggregating such representations before propagation throughout the graph. The GCN model will learn the node's hidden representation not only based on its own features, but also that of its neighbours.

▪ Network Module - Training the Network

1. Transformation

Firstly, the graph network is transformed into a vector space using the adjacency matrix A. Given a network of N nodes, this a N by N square matrix whose ij element A_{ij} corresponds to the number of connections between users i and j.

2. Normalisation

Post-multiplication, normalisation of A is carried out to prevent a change in the scale of the feature vectors. Using the inverse of the Degree Matrix D^{-1} to multiply A is a way to normalise the feature representations.

3. Propagation

A is inserted in the forward propagation equation and this enables the network to learn the feature representation based on node connectivity. Within the network, forward propagation is used to propagate the features to the next layer. The forward pass in the network at the first layer is achieved using:

$$H^{(i\,+\,1)} \;=\; \sigma(W^{(i)}\, H^{i}\, A^{*}) \qquad (3)$$

Where $\boldsymbol{\sigma}$ is the activation function and $\boldsymbol{A}^{*}$ is the normalised adjacency matrix, $\boldsymbol{H^{i}}$ is the feature representation at layer i and $\boldsymbol{W^{(i)}}$ are the weights at layer i.

A three-layer GCN module is used in the model - one input layer and two hidden layers. The input layers receive the model input - the synthetic network generated. With respect to the model operations, the learning process of the representation learning operation is performed with the layers. Using a three-layer GCN allows the GCN to perform three propagation steps during the forward message pass and effectively aggregates the neighbourhood features of every user in the network up to the third order (i.e., nodes up to three hops away). Each node's field of view is limited to 3 steps during the training.

- **Output Module**

This module produces the given outputs for the model and classification tasks application - a two-class (binary) user classification problem within a social network. The module handles the several model functions including:

▪ Model optimization and epoch training count.

Optimization tunes the model via several hyperparameters to ensure that the algorithm delivers best performance as measured on the dataset. To train the network, a conventional Adam optimizer (adaptive moment estimation) which combines two gradient descent methodologies into an algorithm for optimization technique for gradient descent. A learning rate of 0.01 is used. The learning rate affects how quickly the model arrives at its best accuracy. Using an Adam optimizer enables better optimization with faster computation time whilst requiring fewer parameters for tuning.

The number of learning cycles by the algorithm on the entire dataset is set by the number of epochs set at 100 epochs. The choice of 100 epochs was chosen as an optimal number of epochs required to mitigate overfitting while ensuring generalisation capacity of the GCN when applied to the model problem.

▪ Computing the values for the loss

The loss is computed for the model over the epoch counts using the evaluation metric. The Negative Likelihood loss (NLL) is used as the evaluation metric in the model. This metric is a cost function used for machine learning models. Using the NLL, the aim is to estimate the model's performance. A low NLL value indicates that the model's prediction accuracy is high.

5.3.2 NetTv2 - Real-World Network

Similar to the synthetic network, the network is equally initiated as a directed weighted graph with nodes and edges with network implementation using the DGL library. The real-world network (RN) - Zachary's Karate Club Network is used to validate the results obtained from the synthetic network. The various modules used as part of the synthetic network are equally used in the real-world network simulation with similar programming codes while performing the same functionalities.

- **Input module**

The network is created from two arrays which store all edges in the network. The first array (Src) array contains values representing the edge source points while the second array (dst) contains values representing the edge destination points. A DGL graph is constructed using the values within the array. The graph network generated is visualised using NetworkX – a python-based package for visualising. Figure 29 shows the real-world network pre-training (See Appendix B for programming codes).

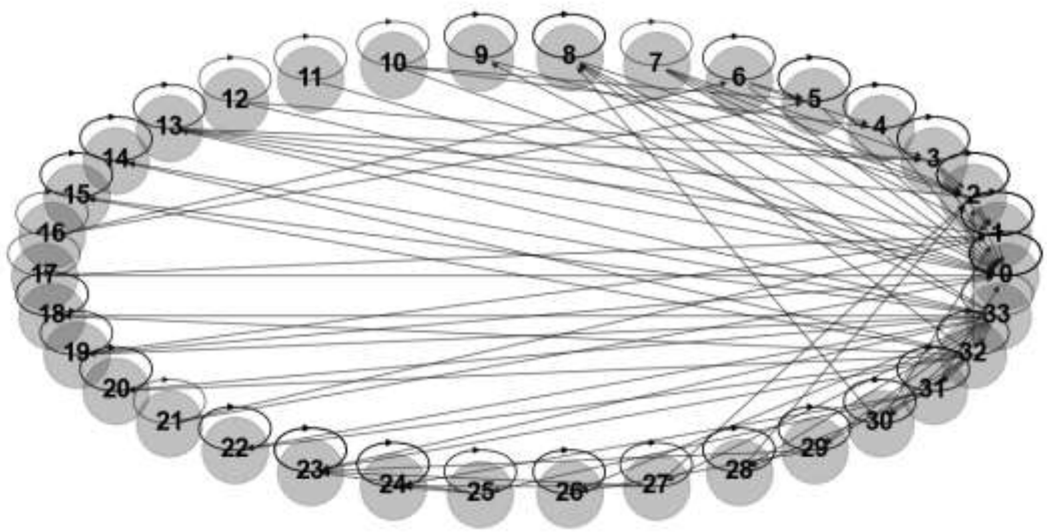

Figure 29. Zachary's Karate Club Pre-Trained Network
A densely connected circular network of 46 nodes, showcasing intricate interconnections and relationships between them.

The network features 34 nodes and 190 edges between the nodes. Two super nodes are identified in the network - nodes s_0 and s_1 with indegrees (number of inbound edges) of 33 and 17. Mirror nodes are identified as nodes r_3, r_8 and r_{23} having an in-degree of 7, 7 and 11 respectively. All other nodes in the network are classed as normal nodes.

- **Labelling Module**

User-features/Profiling is analogous to node labelling in GNN which adds additional information to nodes. Nodes and edges can have several user-defined named features for storing graph-specific properties of the nodes and edges. For feature creation and initiation in the real-world network, the most connected nodes in the network are labelled with profiles created for such users. Profile creations and initiation of the nodes in the real network mirrors that of the synthetic network. Six nodes ($0, 3, 5, 8, 23$ and 33) are initialised with two profiles $(0, 1)$.

Table 22 shows the selected nodes in the network and their initiated belief profiles. A learnable embedding is assigned to all nodes in the network as the model's input feature (See Appendix B).

Nodes	Belief Profile Type	Initiated value
$s0$	Type 2	1
$s33$	Type 1	0
$r3$	Type 1	0
$r5$	Type 1	0
$r8$	Type 1	0
$r23$	Type 2	1

Table 22: Real-world nodes and their belief profiles

- **Neural Network Module**

A three-layer GCN variant of GNN is used as the Neural Network (NN) module to learn each node's features. GNNs are naturally inductive because they learn the same neural networks on all the nodes and edges (Wang *et al.*, 2019). Using the defined GCN, user-level representation learning is achieved.

- **Output Module**

The output module handles the outputs for the model. The model's hyperparameters are like those used in the synthetic network. Adam Optimizer is used to perform model optimization with a learning rate of 0.01. The negative likelihood loss is used to compute the loss in the model with the duration of the model training being 100 epochs.

5.4 Results

The user-level representation learning operation performs node classification. Node level representation where an input outputs a vector learned representation for each node that preserves the individual node attributes and graph structure. The classification is performed using the semi-supervised learning approach where a small amount of labelled data is combined with a larger amount of unlabelled data during training (Zhu, 2008). The simulation results from the synthetic network are validated by the results from the real-world network.

5.4.1 Synthetic Network

Rationale: With a graph network representing a social network, using a set of parameters, the structure of the relational data that informs connections within the network is examined.

The graph was created with two nodes serving as the seed nodes. The graph network output identifies nodes (0 and 1) as the super users, user 0 has an in-degree value of 15 and user 1 has an in-degree value of 14. User 0 has an in-degree connection form every other user in the network excluding user 1. Nodes ($3, 6$ and 12) established as the mirror users having in-degrees of (4, 4 and 6) respectively. All other nodes in the network are identified and classed as normal users with their sub-class dependent on the belief profile types created. As a featureless dataset, features are added using an embedding matrix in which node in the graph is assigned an embedding vector which is updated during the training together with the model parameters. This is done using the pytorch embedding model.

The network graph is featureless, hence, each node in the network contains a word count vector as its features - embeddings, which serves as the identity of each node. The matrix generated sees each row representing the embedding vector of each node. As stated earlier for the initialization, five users are initialised with two different profiles. With the model parameters created, the model is trained. The network training is realised using the network a conventional Adam optimizer is used. The task is to classify the nodes in the network based on their edge information considering the profile states initiated amongst a select group of nodes within the network.

At epoch 0 (training cycle one), all users in the network have been initialised with their initial profile state. This generates an initial data set which defines the initial representation of graph data, assigning features to users. The user positions generated reflect the initial positions of nodes in the network in relation to their initial states. The input layer defines the initial representation of graph data, which becomes the input to the GCN layer(s). The idea is to assign a feature representation to the users within the network. The layers encode the information on the structure of the network, then exploit this information to update the initial representation of nodes and their edges.

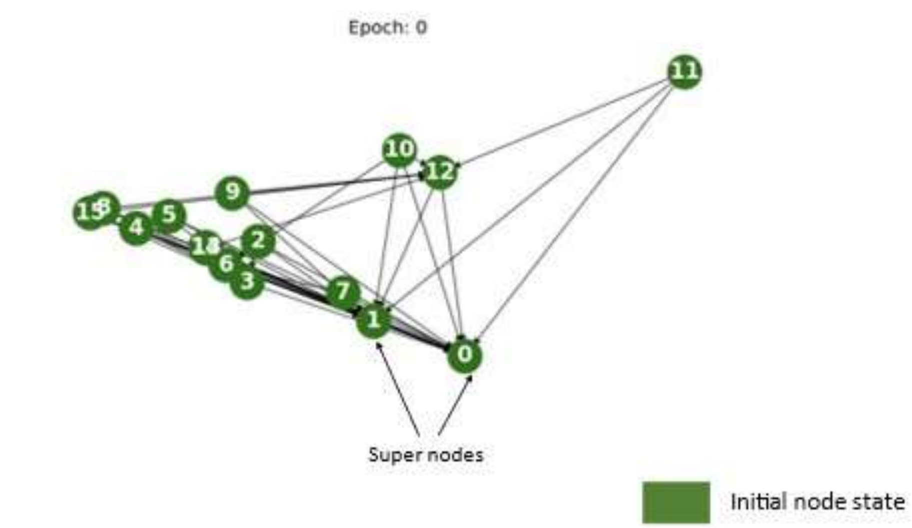

Figure 30. Network at Epoch 0
A network graph at Epoch 0 displaying a cluster of interconnected nodes, with nodes 0 and 1 being prominent super nodes.

Figure 30 shows a visual of the network at Epoch 0. Using the GCN layer, the goal is to update the *d*-dimensional representation of the nodes obtained from the input layer using the - message passing framework. Consequently, the new representation of the node encodes and represents the local structure of the graph.

At epoch 99, the users in the network output shows feature representations that separate users in the network into two distinct profiles as indicated by the distinct colours of the users as seen in Figure 31 – the user states have been updated and affine transformation of the network has taken place which sees nodes taking up positions in the network that are relative to their beliefs. The nodes are clustered together based on their distinct states with the tightly grouped nodes at both ends of the graph network indicating strong attachment to their final state. The clustering can also be attributed to the effects of message passing which allows nodes to be updated with neighbouring node states up to three hops away. The distance between the two groups of nodes shows strong polarisation between the nodes in the network.

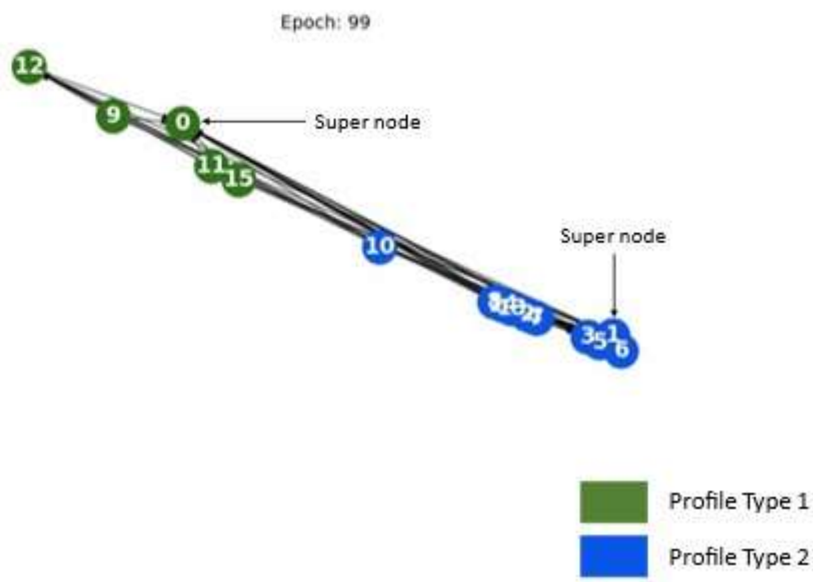

Figure 31. Synthetic Network at Epoch 99
At Epoch 99, a linear network graph showcases nodes of two distinct profile types, with Nodes 0 and 1, being emphasised as super nodes, acting as major connection hubs

With this being a multi-class classification problem – there are two output user profiles. The last GCN layer computes the user embeddings and a SoftMax function is applied on the output layer. This computes the probabilities for the profile class by outputting a value between 0 and 1 where the value 0 indicates a negative state and 1 indicates a positive state.

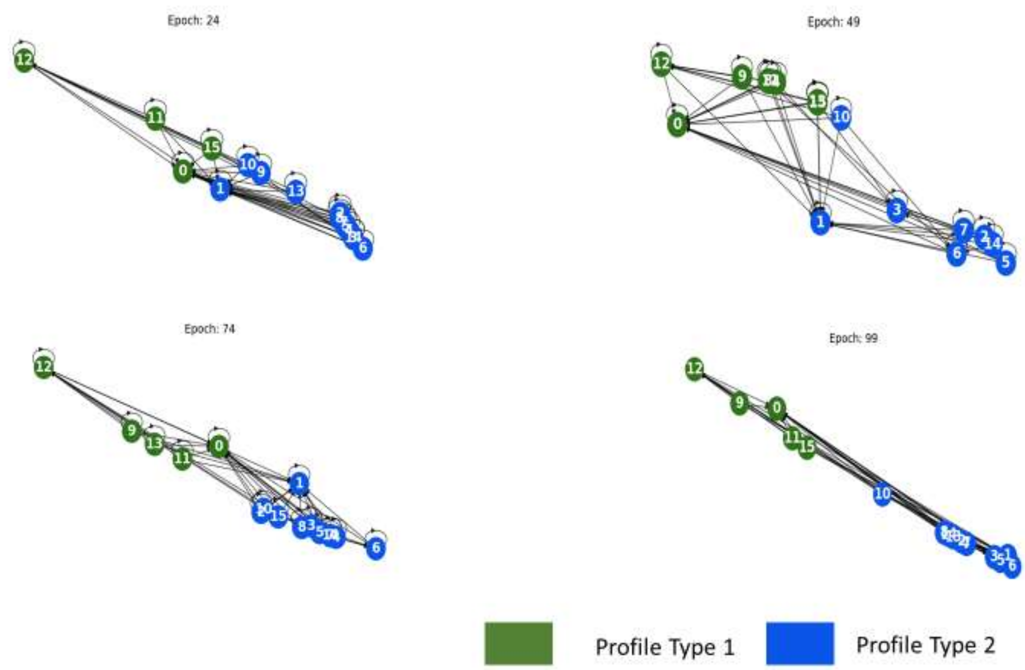

Figure 32. Animation of Synthetic network Training
A series of network graphs displaying the evolution of connections between nodes of two distinct profile types across different epochs: 24, 49, 74, and 99

The model classifies the network along the two profile types used. As shown in Figure 32, the nodes in the network are observed to have formed clusters according to their belief profiles informed by both their field of view and initial connections. The users in the network can be seen to have formed clusters according to their profiles informed by both their field of view and their initial connections. Using a 3-layer GCN means that each user's field of view is limited to three steps - users will only be able to aggregate neighbour user features up to 3 steps away over the duration of the training.

The loss is computed for nodes with profiles initialised (5 nodes). Figure 33 shows a graph of the NLL of the model over the epoch count of 100. The model algorithm converges very quickly at around 50 epochs for the synthetic network. When training a model, given the input features, the aim is to find the minima of the loss function. This maximises the model accuracy by increasing the probability that the unlabelled nodes will cluster with the right category.

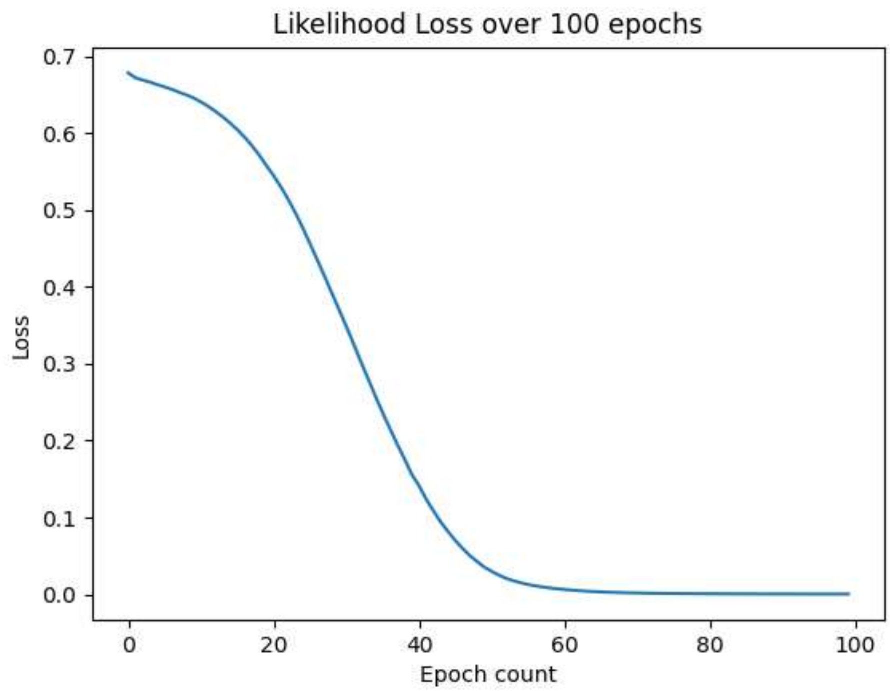

Figure 33. Negative Likelihood Loss - Synthetic Network
A graph illustrating the decline in likelihood loss over the span of 100 epochs

Table 23 shows loss values for the model at intervals of 5 up to epoch 40. The values are seen to be identical to the plot above.

Epoch count	Loss
00	0.7802
05	0.7400
10	0.7141
15	0.6987
20	0.6749
25	0.6435
30	0.6004
35	0.5317
40	0.4562

Table 23: Model Loss Values

- **Discussion on Synthetic Network Results**

The dataset of the synthetic network was specifically created to allow such that it simulates an environment that features several of the model's definitions. Referring to the first hypothesis of question one, from the results at epoch 99, nodes are clustered along the lines of the two belief systems found in the network. The super and mirror nodes with their belief strengths informed by their number of connections can be seen to play a pivotal role in diffusion within the network. The simulation shows the effects neighbour nodes can have in belief adoption in the network (hypothesis three). The role of a node's neighbourhood is seen in the ego networks of the nodes in the graph with node states being influenced by state updates from neighbouring nodes. As a function of the number of layers used in the NN module for each node in the graph, their neighbourhood consists of neighbouring nodes within 3 hops from the target node.

Further analysing the results of the synthetic graph network simulation shows that some relevant real-world situations are captured by the model. In relation to the role beliefs play in the links nodes establish (research objectives), it is observed that the presence of strongly biased users in networks that are split along two belief systems has a significant effect on the expected belief consensus adoption amongst users in such networks. Regardless of the user states and the nature of users in a network, the model results infer that in most interactions in social networks only two types of opinions will emerge - both on the opposites of the bias scale.

Furthermore, the internal structure of the network plays a role in users having the initial private beliefs confirmed or mitigated. While public signals generated in networks can vary over time, they are dependent on the state of users in the network and the user connection/relations of such users. The ego neighbourhood (in and out) of such users are important in determining the level of efficiency of the public signals and hence the final state of the network.

5.4.2 Real-World Network

Rationale - The learning problem of the synthetic model is node classification under a semi-supervised learning setting. The real-world network is used to validate the performance and results of the synthetic network using identical model definitions, settings, and parameters.

The Zachary's Karate club dataset is used to validate the model results from the initial training set. The dataset is a representation of user relationship and interaction on the network with respect to information diffusion. In this network, the nodes represent members of a karate club and the edges, their mutual relations. The identified super and mirror nodes form the labelled nodes which are used in conjunction with unlabelled data (rest of the nodes) for the learning problem.

As with the synthetic network simulation, in the RN simulation node features representation were observed to mirror that of the set parameters for the classes of nodes created as part of the simulation. Epoch 0 (training cycle 1) as shown in Figure 34 sees all users in the network being initialised with their initial profile state. This generates an initial data set which defines the initial representation of graph data, assigning features to users. The input layer defines the initial representation of graph data, which becomes the input to the GCN layer(s).

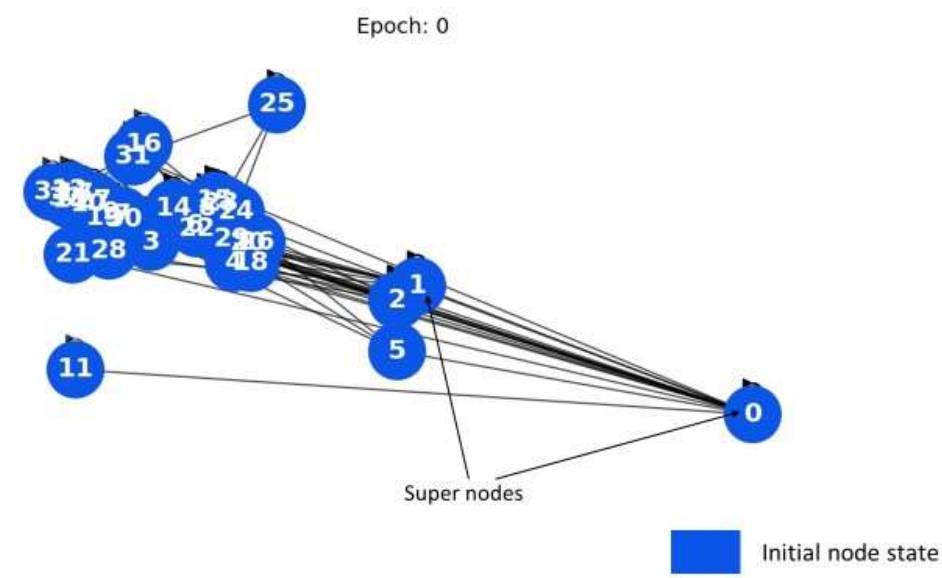

Figure 34. RN at Epoch 0
A visual representation of a network at Epoch 0, highlighting the initial node state in blue and indicating a set of super nodes interconnected, with node '0' as one of the prominent focal points.

At epoch 99 shown in Figure 35, the users in the RN network output show feature representations that separate users in the network into two distinct profiles as indicated by the distinct colours of the nodes. Two differing node states are identified, represented by nodes in the green and blue clusters. Nodes are observed to be clustered around other nodes with a similar state - private state. Message passing which allows nodes to be updated with neighbouring node states up to three hops away influences the final state of nodes in the network. The groupings of the nodes show the effects of neighbouring node states and transitivity. Amongst the clustered nodes, groupings of tightly clustered nodes seen across the graph network indicate strong attachment to their final state.

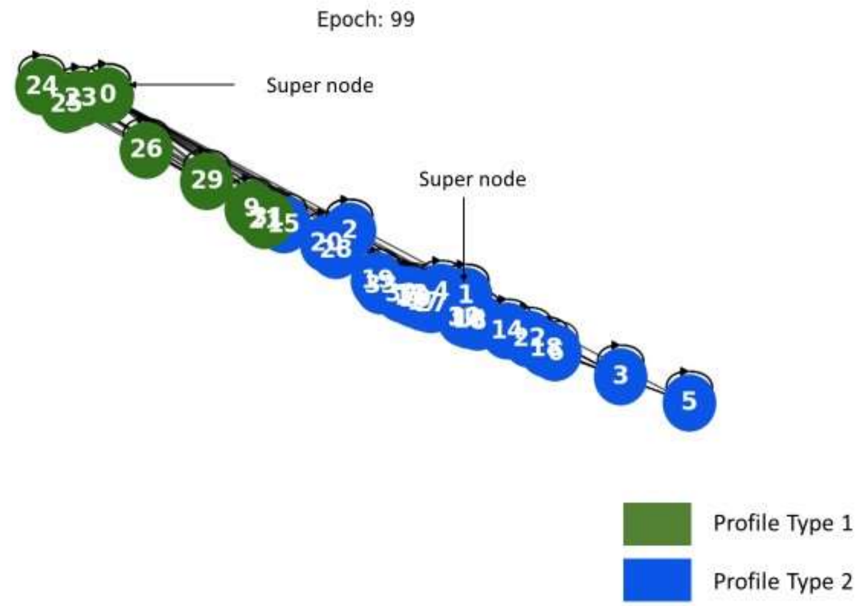

Figure 35. RN at Epoch 99
A network diagram at Epoch 99 showcasing two distinct profile types. The green nodes represent Profile Type 1, while the blue nodes symbolize Profile Type 2. Two super nodes serve as central connectors within the network structure.

The RN results are used to validate the results of the synthetic network. In both model simulations, only a select number of nodes in the network are initiated with profiles while the profiles of other users in the same network remain unknown and are available for training - enabling the network to learn about the node structure – connections within the network. The NLL as the loss function is computed for the model as shown in Figure 36.

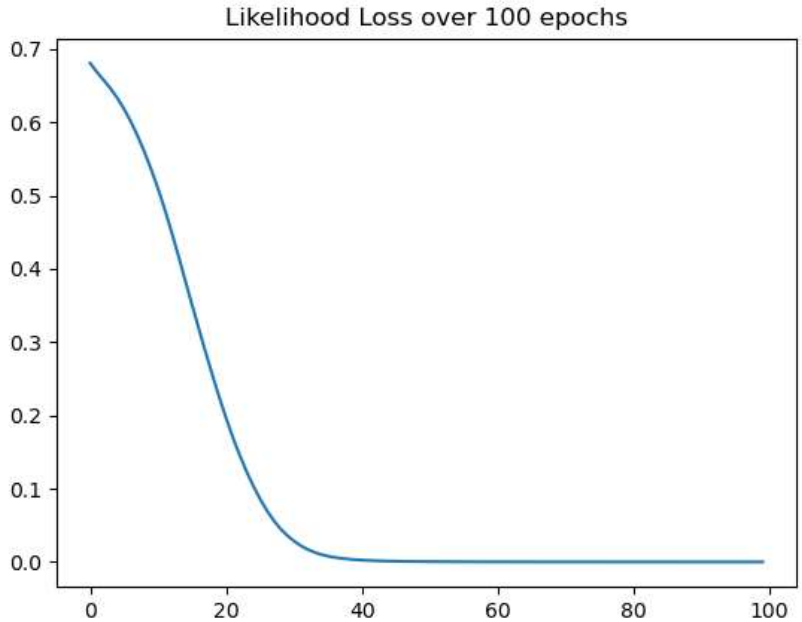

Figure 36. Negative Likelihood Loss - Real-World Network
A graph illustrating the decrease in likelihood loss over 100 epochs. The loss starts near 0.7 and steadily diminishes to approach 0 by the 100th epoch.

5.4.3 Discussion of results of Synthetic and Real-world network

In comparing the results from the synthetic network to those of the RN for validation, the following observations were made and inferred.

1. **User Connections and Profiles**

From the post-training results from both networks, it can be inferred that in a network, a relationship exists between a user's initial belief, the connections established, and final beliefs adopted for a given piece of information diffused in the network. This relationship is however dependent on the user's vision (initial connection and the number of hops during message passing which informs the final user state) within the network. It justifies the assumption made in the project's definitions that mass adoption by agents will only occur when neighbouring agents have the same belief profile and confirmation bias has already been achieved. This also provides grounding to the assertion made regarding the second research question that "Private beliefs and the beliefs systems which inform them play a major role in the connections users establish in a social network influencing diffusion in such networks".

In answering the question "*How can instances of irrational information diffusion in OSN interactions be identified?"*, post-training, the relational information contained in both graph networks using belief profiles as inputs into the networks establishes that users in a social network will form clusters with likewise users. This supports the assertion made in the first hypothesis in relation to the question. Furthermore, the roles that the user classes play in the diffusion process was further supported. The more the number of highly placed users with a particular belief the more likely that other users having such users within their vision will adopt such beliefs lending support to the assertions of the role of neighbouring users (second hypothesis) in the spread of information in the network.

2. Variance

The model's learning algorithm demonstrates low variance. Subtle changes were noticed when there was a change from one training dataset to the next, indicating that the model algorithm performs well in terms of picking out the hidden underlying mapping between the inputs and the output variables. This should allow for a framework that is applicable to a wide range of real-life scenarios.

A t-test is carried out on the values of the loss functions of both networks over two simulation runs. The t-test is used to determine if there is a significant difference between the means of simulation runs for each of the network types. To establish the equality of the variance of both networks, an F-Test is carried out. The test statistic for the F-Test has an F-distribution under the null hypothesis (Lomax and Hahs-Vaughn, 2013).

- **F-Test**

The test was performed on both networks using their loss values as the sets of observations recorded. A first test was performed on the loss values at selected epoch intervals of the synthetic network over two full training runs. A second test was performed on the loss values of the RN over two network training runs.

Synthetic Network	Zachary's Network			F-Test Two-Sample for Variances		
1st Run	**1st Run**					
0.7802	0.7787				**Synthetic**	**Zachary's**
0.7400	0.6893			Mean	0.6489	0.3132
0.7141	0.5847			Variance	0.0108	0.0984
0.6987	0.4286			Observations	9	9
0.6749	0.2374			df	8	8
0.6435	0.0805			F	0.1095	
0.6004	0.0162			P(F<=f) one-tail	0.0026	
0.5317	0.0029			F Critical one-tail	0.2909	
0.4562	0.0007					

Table 24: F-test on both Networks

The third test was performed on the NLL values of the synthetic network and the RN as shown in Table 24. The value of focus in the F-test was the F critical value which is compared against the F value. A larger F critical value shown indicates that the variances of both networks are equal.

- **T-Test**

The two-sample t-Test assumes a homogeneity of variance between the data. The first test was done on the NLL values of the synthetic network and a second test on the values of the RN. Both tests used an alpha value of significance level 0.5, used as a reference value to either accept or reject the null hypothesis.

t-Test: Two-Sample Assuming Equal Variances		
	1st Run	**2nd Run**
Mean	0.6489	0.6276
Variance	0.0108	0.0103
Observations	9	9
Pooled Variance	0.01052636444	
Hypothesised Mean Difference	0	
df	16	
t Stat	0.4397	
P(T<=t) one-tail	0.3330	
t Critical one-tail	1.7459	
P(T<=t) two-tail	0.6660	
t Critical two-tail	2.1199	

Table 25: T-Test Synthetic Network

t-Test: Two-Sample Assuming Equal Variances		
	1st Run	**2nd Run**
Mean	0.3132	0.27031
Variance	0.0984	0.0909
Observations	9	9
Pooled Variance	0.0946	
Hypothesised Mean Difference	0	
df	16	
t Stat	0.2959	
P(T<=t) one-tail	0.3856	
t Critical one-tail	1.7459	
P(T<=t) two-tail	0.7711	
t Critical two-tail	2.1199	

Table 26: T-Test for Real Network

The highlighted columns in Table 25 and Table 26 indicate the key values of importance in the test. The output shows the mean for the synthetic network (1st run - 0.65, 2nd run - 0.63) and that of the real network (1st run - 0.31, 2nd run - 0.30) with a mean difference of zero indicates that there is no difference between the two simulation runs. A total of nine observations (NLL for epoch intervals of 5 up to 9 times) are the differences between the two sets of NLL values used in each of the tests.

The p values (two tail) seen in the tables is compared against the significant level (0.5). The values are significantly larger than the significant level ($p > 0.5$) indicating that the results of both simulations are not statistically significant. This means the null hypothesis is not rejected and there is no significant difference between the loss values of both datasets.

3. Model Learning Algorithm

The algorithm helps the model to find patterns in both datasets based on the model parameters. It is observed from the simulation that a slightly different model is learnt by the algorithm each time it runs on the same data set while the model's performance remains the same in terms of loss and accuracy (convergence). This is particularly seen when the training data is quite small and can be attributed to the NN modules used within the simulation. With no test or train subsets used within the dataset cross validation is not performed and the algorithm trains the dataset.

In NetTv1 a centrality-based approach was used to find the most important users in a network. Contrary to centrality-based techniques, which generate a node characteristic based on its immediate neighbourhood. NetTv2 uses a learning-based approach implemented using GNNs, where the learning process considers the complete graph structure, and the user feature is viewed as the user embedding and allows for the individual states of users in the network to be known. Referencing the project aims and objectives, the following results inferred/observed from the model simulations addresses some of the stated research aims and objectives as well as the second research question.

▪ In the representative OSN datasets

Users are inferred to exhibit beliefs and that signals generated in interactions with other nodes in the network can reinforce or mitigate these beliefs as a function of confirmation bias. Levels of bias can be said to be dependent on a user's initial private beliefs and the belief signals from the node's neighbourhood. A resulting effect of this is that some nodes would be strongly biased while others will be weakly biased with the intensity of the resulting polarisation and establishment of node clusters in the network dependent on the size of these groups of nodes.

▪ The interdependency and dynamic nature of interactions in OSNs is observed

Users and their beliefs within the network evolve over time. A user can easily engage different users adopting new beliefs that can be easily destroyed. The relations informing these beliefs evolve over time and form the backbone of diffusion in social networks. In the model simulation, this is seen in the process of the establishment of a consensus belief - largest node clusters of a particular profile state as adopted by most nodes in the graph network over the model training period of 100 epochs.

▪ It is observed that a node's structural position influences the overall network structure.

A complex network's structural characteristics provide important insights into its dynamics and function in terms of the diffusion potential and potential spreaders/adopters. This holds true when the relationship between the position occupied by a user and the role played by that user is observed in the network during diffusion. It has been established that not all users have the same impact on the spreading of information in a network, users with a higher number of connections would contribute much more to the spreading of misinformation than users with few connections. This is seen in the model in the definitions created for the super and mirror nodes who occupy key positions in the network and figure prominently during the model training. A cluster of such users with similar beliefs in a network is expected to leave the overall belief system of such a network strongly aligned with the prevalent belief of the users.

The model is evaluated on node classification tasks under a semi-supervised setting. The results from the synthetic network are validated against the results from Zachary's network. The model algorithm is shown to demonstrate good generalisation of data and sufficient data simplification. The applications for analysing diffusion and evolving networks can be applied in diverse real applications, one of which is the focus of this research project - the problem of misinformation diffusion and adoption where the interest is in observing not only how information is diffused within the network but on the users who enable the diffusion process.

5.5 Conclusion

NetTv2's task was representation learning by simulating user interactions in the presence of two differing belief states. The node classification task requires the model's algorithm to identify the labels of samples (shown as nodes) by examining the labels of their neighbours. This task was used to predict the belief state to be adopted by nodes in a graph network representative of an OSN. Hence, creating meaningful representations of node states for a graph network using the existing belief structure answering the research question asked on the "identifying instances of irrational information diffusion in user interactions". The role belief plays in user interactions and connections was established as seen in the final node state in the network - proving the first hypothesis associated with question two. Examining the outcomes of the model simulations reveals that the model algorithms can anticipate a node property that doesn't already exist based on current node properties. This property is seen in the algorithm correctly predicting the final node state based on the node connections and vision. Using the embeddings assigned to all the nodes in the network and the summation of neighbourhood node information, the model can class nodes into one of two said belief classes - proving the second hypothesis.

Following from this model, NetTv3 will extend the functional context of NetTv2 introducing agents with differing attributes and training these agents on these attributes using neural networks and a learnable embedding for the purpose of classifying final user states post interactions between users in the presence of differing attributes - implemented in GNNs. The model's task will be multi-class representation learning, learning the internal states of users whilst still preserving their unique attributes.

6. Third Model - Network Translations Version Three (NetTv3)

6.1 Introduction

This chapter introduces the third model - NetTv3 which extends representation learning operations into multiclass classification. The model sees the introduction of more distinct profile features and more users initialised with these features in line with the user classes already established in the research. Diverse user attributes are introduced within each profile feature allowing the model to replicate much more realistic user characteristics found in OSNs.

To answer if "*How might Artificial Intelligence (AI) algorithms be employed to mitigate the spread of irrational information*" (question 3), this model simulation focuses on the more intentional dynamical attempts to spread misinformation. To achieve this, the model introduces several user attributes to replicate heterogeneous user types and relationships found in social networks - independent features between users and their connections. This allows for more complex relations and forms of bias to be simulated over varying time states. **This model aims to establish if an AI agent can be used to enforce specific views in a social network. By using the knowledge of a user in terms of private beliefs and role in the network it is shown that influencing the beliefs adopted by regular users in a network is possible.** Hence, getting access to users' private information is a way to promote the spread of misinformation through such a user. User interaction in the presence of biased and unbiased agents is simulated to understand their effects on other users in the network and establish if the hypothesis posited with respect to the question is valid.

6.2. Model Overview

One of the most studied problems in social networks is the problem of selecting the most influential nodes in the network in terms of opinion dynamics in models (Agha Mohammad Ali Kermani, Ghesmati and Pishvaee, 2021). This problem centres around agents that engage in the diffusion of information/ideas/opinions by means of existing social ties. In the exchange of information between users in a network, adoption of beliefs/opinions is dependent upon the state of the users and the state of the world (Kempe, Kleinberg, and Tardos, 2003). Several machine learning-based models have been developed to solve this problem using algorithms that focus on the role of a small set of seed nodes for link prediction/classification tasks (Sun, Zhou and Guan, 2016; Zhao, Li and Jin, 2016b; Liu and Wu, 2018; Zhang and Chen, 2018). One drawback of such models is that they consider only the internal state of these nodes.

To address this drawback, in defining the functional context of this model for the diffusion of beliefs in an OSN G, represented by a directed graph, each node is individual in actions and state in the graph network. Figure 37 illustrates the network described. Nodes can be in one of three states in the diffusion process - active, passive, and phasing. The level of engagement of a node is dependent on its vision - how far out a node can see in terms of the states of other nodes. Through neural message passing, this also influences a node's final state.

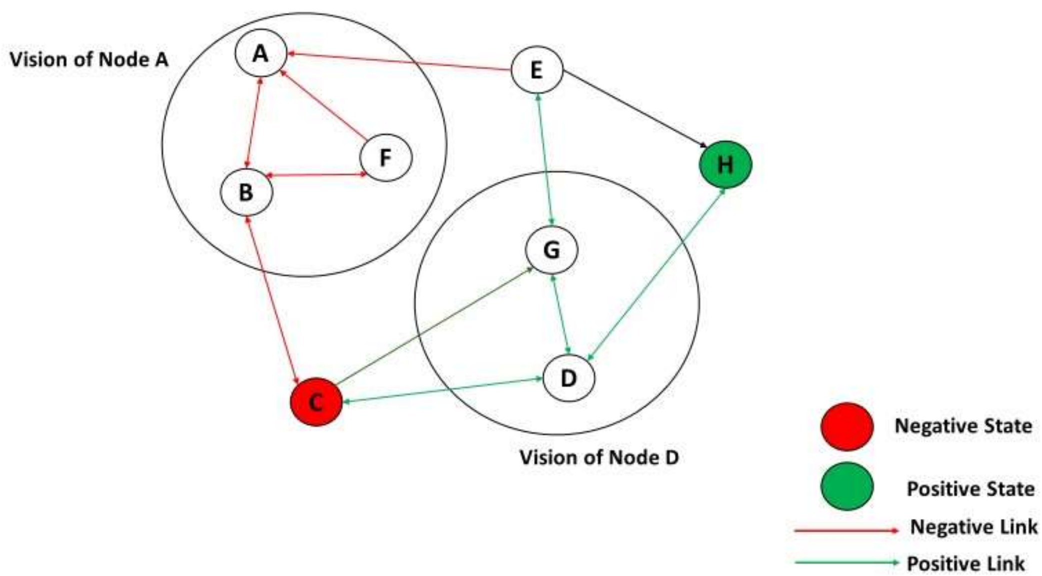

Figure 37. Network State Visualisation

Figure 37 shows the states of nodes based on their links and engagements. Each node is in either a positive state (green) or a negative state (red). The circles labelled "Vision of Node A" and "Vision of Node D" highlight the subjective perception of the network from the perspective of nodes A and D, suggesting that each node has a limited view or understanding of the entire network's structure and the states of other nodes within it.

In the previous model, it was established that users will form clusters with users of likewise beliefs as shown in the results of the synthetic network and real network. While all users in the network were given a learnable embedding that captures their belief profiles and is used in the training process. **The previous model had two drawbacks: the node features/characteristics - as implemented using embeddings were largely homogenous (two user attributes) limiting the scope of influence maximisation in belief adoption and only a two-class classification was possible**. To address this, in NetTv3, multiple attributes are used in defining the features of nodes in the network which permits a more diverse set of nodes and node capabilities. The model is also able to further explore the role of bias, its relation to belief formation, adoption, and diffusion within the network.

The model also adopts the Graph Neural Network (GNN) framework for learning the internal state of the graph nodes and their edges. User classification in the network is achieved using representation learning, which allows for the model to classify users into multiple classes. The model's aim in its functional context is to perform multi-class user classification in a network in line with definitions and parameters. The model also adopts the definitions and assumptions made with respect to the project framework. One such assumption is the dynamics of the network - with changes at the node level corresponding to changes at the network level. Specific assumptions are also made with respect to this model:

• In addition to the earlier assumed belief adoption, a belief evolution is assumed. It is assumed that nodes will update their beliefs based on the influence of other nodes within their vision and on public opinions within their immediate network.

• Nodes are assumed to be biased or unbiased, with biased nodes also assumed to have a confirmatory bias threshold (the threshold at which nodes adopt a new state due to feedback from their environment.

6.2.2 Problem statement

Like the previous model, NetTv3 uses representation learning operations in GNN for node predictions. The input into the network is a graph represented as a matrix of node and edge specific features.

The problem statement of NetTv3 is like that of the previous model iteration - NetTv2. In the model implementation, the user classes (super (s) and mirror (r) nodes, ignorant (f) nodes and informed (a) nodes) are implemented as labels on the nodes in the graph. The aim of the model simulation is to the aim of node clustering based on their features.

Statement - When it comes to OSNs, a graph representation of the social network is the most accurate because pairwise connections between individuals do not form a grid. Users are represented by the graph's nodes, while connections between nodes are represented by the edges between them (users). A three-dimensional feature matrix for each user contains messages, pictures, and videos. The connectivity amongst nodes is defined as a graph $G = (V, E)$, the input graph is represented as a matrix of user features, $X \in R^{N\,x\,F}$, where N is the number of users and F is the number of input features for each node and $A \in R^{N\,x\,N}$ represents an adjacency matrix for the graph G which defines the node relations in the graph. For each edge E between nodes in the graph, their weights serve as the features. For each node V In the graph, there is a set of defining features created as embedded vector representations.

6.3 Methodology

Network Translation (NetTv3) extends the functionalities and tasks of NetTv2. Like the previous model, the model is a GNN centric model with an identical composition in terms of the number and types of modules, namely, input module, labelling module, neural network (NN) module and an output module. There is however a difference in the internal structures of each module in NetTv3 from that of NetTv2. The model is implemented and simulated using a synthetic network with the real network (Zachary's Karate Club) used to validate the results of the synthetic network. A cost function is used to validate the result of both models. The model is summarised in Table 27.

Model	Components	Description
NetTv3	Model Components	Input module for network generation
		Labelling module for dataset labelling
		Network module for training the network
		Output module for evaluating and displaying the network
	Datasets	Synthetic network (80 nodes) and real-world (Zachary's Karate Club – 34 nodes) network
	Neural network	3-layer GraphSAGE network.
	Optimizer	Adam Optimizer
	Period	100 epochs
	Evaluation Metric	Cross Entropy

Table 27. NetTv3 - Model Summary

6.3.1 Synthetic Network

The problem of intentional attempts to spread misinformation is presented as a graph and implemented as a generated synthetic network. The network is initiated as a directed weighted graph with nodes and edges. There are differing weight values on the edges of the nodes giving the nodes in the network differing diffusion capabilities. The direction between the edges of the nodes in the network are defined. As a directed graph, $G = (V, E)$, where V is the set of nodes and E is the set of edges connecting the nodes, the relationships between the edges on the nodes are purpose defined to test the model hypothesis with regards to connections between the different classes of users and belief types.

- **<u>Synthetic Network - Input module</u>**

The synthetic network creation and display is handled by this module. As with the previous model, several library packages are required to handle graph creation and display - NetworkX library, deep graph library (DGL), matplotlib library, NumPy library and pandas library. The dataset consists of a single graph, its nodes and features and is representative of a network of users and their political leanings. In the dataset, the members of the network are represented as nodes and their interactions as edges (see Appendix C for dataset description).

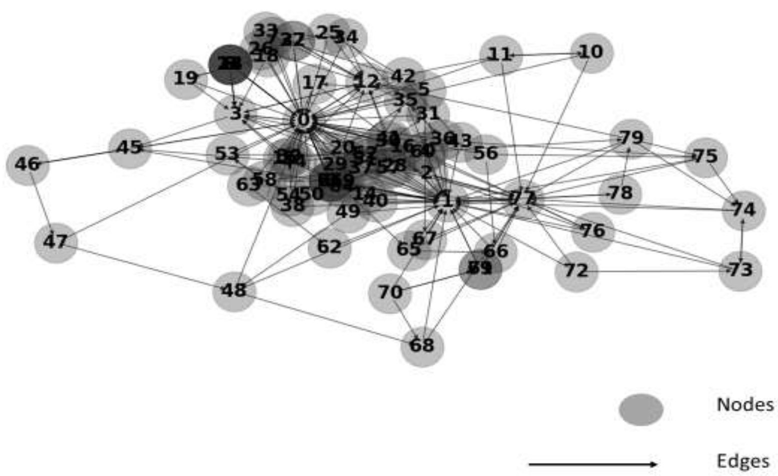

Figure 38. Graph Network Generated from CSV Files
This image shows the nodes of the synthetic network (labelled with numbers) connected by edges.

As shown in Figure 38, a graph of 80 nodes is created from a pair of NumPy arrays which takes beliefs and bias created numerically and class created as a string as attributes of the nodes, these attributes form the labels of the nodes and edge weight as a numeric feature of the edges. The indegree (incoming connections) of nodes in the graph is used to identify the key nodes (user s and mirror r) in the graph network as shown in Table 28.

Node	Indegree
$s0$	63
$s1$	43
$r3$	15
$r12$	16
$r77$	24

Table 28: Indegree of Nodes for the Synthetic Network

- **<u>Synthetic Network - Labelling module</u>**

The labelling module is responsible for the definition and creation of features for both the nodes and edges in the network. This is achieved using embeddings – vector representations of each node which contain information for each node supported by the DGL Library. In defining the dataset for the synthetic network nodes except for the super and mirror nodes, the rest of the nodes are randomly assigned to a class label randomly. In the graph network, both nodes and edges have feature data with the features stored as key/value pairs.

In the definition of attributes, two node features are created for the graph - belief and bias features. **Three belief profile types are created for the simulation namely: belief type 0, belief type 1 and belief type 2.** Nodes with belief type 0 are defined as having a negative belief profile and would share information considered to be false. Nodes with belief type 1 are defined as having a positive belief profile and would share information considered to be accurate. Nodes with belief type 2 are defined as having a neutral belief profile. **Two bias profiles were also created indicating different bias levels - levels "0" and "1".** Nodes with a bias state of "0" are biased and are assumed to be unable to verify the veracity of information and would adopt information based on it matching their beliefs. Nodes with a bias state of "1" are unbiased and are assumed to be able to accurately verify the veracity of information relying on their beliefs alone.

Table 29 shows the nodes attributes for the first 10 nodes as created in the nodes.csv file.

ID	Class	Belief	Bias
0	super	1	0
1	super	0	1
2	informed	2	1
3	mirror	1	0
4	informed	2	1
5	ignorant	0	0
6	informed	2	1
7	ignorant	0	0
8	ignorant	0	0

Table 29: Node Characteristics for Graph

Table 30 shows the edge attributes for the first 10 defined relationships as created in the edges.csv file. The weights on the connections are used as the edge features as seen in column 3.

Source	Destination	Weight
0	2	0.3185
0	3	1.0000
0	4	0.2274
0	5	0.2669
0	7	0.4754
0	8	0.8863
0	9	0.1604
0	10	0.7460
0	11	0.5893

Table 30: Edge Definitions and Weights

The level of influence and strength of nodes in the network can be ascertained from the weight values on the edges. The independence and dependence of relations in the graph can be established from these values. For example, the value on the connection between nodes 0 and 3 seen in Table 30 indicates that node 0 has greater influence over node 3 in terms of information flow in the network indicating a dependency relation from node 3. Regarding the functional context of the model, the node and edge features serve as the specific information tasks captured in the model and used in the representation learning operation to learn the state of each node in the network (see Appendix C for programming codes).

- **Synthetic Network - Neural Network (NN) module**

Responsible for the training operations, like NetTv2, this module defines the NN architecture and hyperparameters used in the model training. For the classification task, a 3-layer (1 input layer and 2 hidden layers) Graph Sampling and Aggregate (GraphSAGE) is used as the NN. Traditional Convolution Networks (ConvNets) are powerful architectures to solve high dimensional learning problems. Compared to ConvNets, GraphSAGE is one type of NN architecture that utilises the structure of data in a graph. Unlike GCN which can only be used transductively, it is a framework for inductive representation learning on large graphs allowing it to be used on networks with rich node attributes (Hamilton, Ying and Leskovec, 2018).

Figure 39 gives a graphical overview of the convolution process at the first layer while showing how the size of the matrices are factored in the process. Firstly, at the first Convolution layer (Conv1), AX is the matrix multiplication of the adjacency matrix (A) with the features matrix (X) giving a matrix of 80×227. The weights matrix $W^{(0)}$ thus, has rows and columns leading to a matrix x_1 of 80×32. At the following layer, second convolution is fed with x_1 following the same process and the third convolution with x_2, the result is the number of features (for the class attributes) at the end of the 4 features (same number as the total classes) thus giving a matrix x_3 of 80×4.

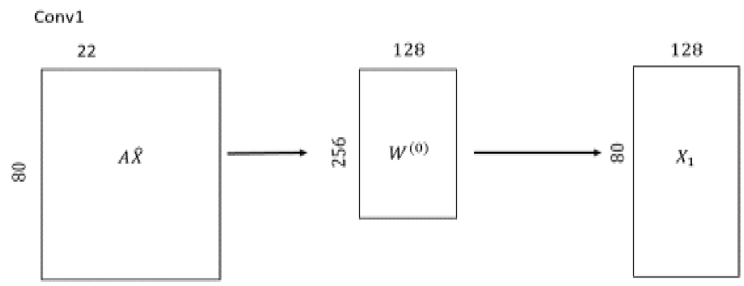

Figure 39. Convolution at Layer 1

Figure 39 show the models convolution operation at layer 1 in the GraphSAGE network. The adjacency matrix A of the graph is multiplied with the feature matrix X to obtain the AX matrix. This process combines the features of connected nodes in the graph.

The architecture and hyperparameters used in the model's NN module follow the GCN model architectures and hyperparameters put forward by Kipf (Kipf and Welling, 2017).

- **Synthetic Network - Output module**

The output module is responsible for the definition of the final model outputs - the multiclass classification. As with NetTv2, the output module in NetTv3 encompasses key model functionalities with similar compositions/implementations including:

▪ Model optimization

Achieved using an Adam (adaptive moment estimation) Optimizer to update network weights iteratively based on training data. As defined earlier, optimization tunes the model via several hyperparameters to ensure that the algorithm delivers best performance as measured on the dataset. A conventional Adam optimizer and a SoftMax function is used in model training. Regarding multiclass learning issues, the SoftMax activation function is frequently employed at the output layer of a neural network.

▪ Learning rate

When training neural networks Gradient Descent is used to optimise the weights amongst the neurons. The learning rate value chosen determines how quickly or how slowly the weight (parameter) values will be updated. Normal selection convections stipulates that it should be high enough to prevent the model from taking too long to converge, while also being low enough so that the model finds the local minima. The model uses an identical learning rate value of 0.01 as with NetTv2 which allows the model to arrive at its best accuracy in optimum time.

▪ Training regime

The number of learning cycles by the algorithm on the entire dataset is set by the number of epochs. The model is trained for 100 epochs.

▪ Loss Function

An effective loss function is required to validate how the algorithm models data. The loss is computed for the model over the epoch counts using the chosen metric. Cross entropy is used as the loss function and to evaluate the performance of the model. This metric is a cost function used for multiclass classification machine learning models. A low value indicates that the model's prediction accuracy is high.

$$loss = CrossEntropyLoss(h_v^{(l)}W, label_v) \tag{4}$$

where $h_v^{(l)}$ is the representation of node v after l layers in the GCN, W are the parameters of the Neural network and $label_v$ is the actual ground truth label of the node.

▪ Accuracy

The degree to which a measured value is accurate in relation to a reference or known value. The model features the training, validation and test accuracy done on the training, validation and test sets created respectively within the dataset.

6.3.2 Real Network

The real network dataset used is a modified Zachary's Karate club (Zachary, 1977) network dataset for the validation. This is also initiated as a directed weighted graph with the relations between the nodes and edges defined. Network implementation and training is done using the DGL library and several imported library packages. The results from the synthetic network are validated by the results obtained from the real network. The various modules used as part of the synthetic network are equally used in the real-world network simulation with similar internal structure, functionality, and programming codes.

- **<u>Real Network - Input module</u>**

Like the synthetic network, the network is created from a dataset consisting of two CSV files. One file - nodes.csv defines the nodes and their attributes and the second file - edges.csv defines the edge relations as edge source points and destination points amongst the nodes in the network. These files are imported into and defined in the model as the inputs for network creation. Using the CSV format allows for the modification of the dataset to create similar attributes as found in the synthetic network dataset. The steps for loading the datasets into the network are identical to those of the synthetic network with similar imported library packages.

The edge source points and endpoints for the datasets are defined and a DGL graph is constructed using the values within the CSV files. The graph is transformed into networkx for visualisation. Figure 40 shows the real-world network pre-training for the first.

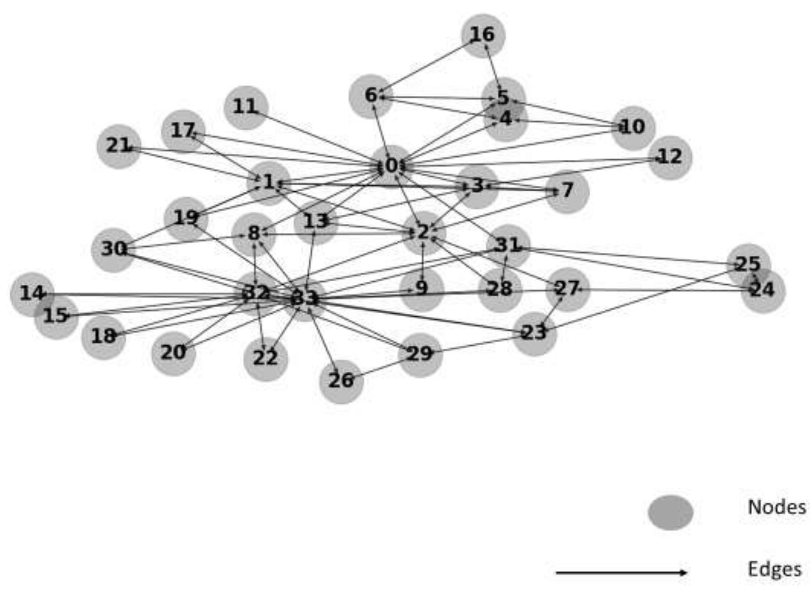

Figure 40. Pre-Trained Zachary's karate club

Figure 40 shows the pre-trained Zachary's Karate Club which is used to validate the results of simulation using the synthetic network

For the dataset, the node relations and number of nodes remains the same with the original Zachary's Karate club featuring 34 nodes and 190 edges between the nodes. With the exception of the prior established super nodes - nodes s_0 and s_{33} with indegrees of 33 and 17 and mirror nodes - nodes r_3, r_8 and r_{23} having an in-degree of 7, 7 and 11 which have their labels in the node.csv file created accordingly, the labels for the rest of the nodes are created as informed or ignorant labels with the nodes being randomly chosen.

- **Real Network - Labelling module**

The labelling module handles the feature assignment for the nodes and edges in the graph network. The features for the nodes use the node attributes created for the nodes in the graph and the features for the edges use the weights created for the edges amongst the nodes. Like the synthetic network, the node attributes of the two datasets used in the real network are created in their CSV files with two feature types as well as the node class label as seen in Table 31 for the first ten nodes.

The profile types and definitions are identical to that of the synthetic network with nodes classed as informed having profile type of 2 and bias profile of 1 indicating neutral and unbiased private beliefs.

ID	Class	Belief	Bias
0	Super	0	1
1	Super	1	1
2	ignorant	0	0
3	mirror	0	0
4	ignorant	0	0
5	ignorant	0	0
6	ignorant	0	0
7	ignorant	0	0
8	ignorant	0	0
9	informed	2	1
10	ignorant	0	0

Table 31: Zachary's network User attributes

The weights on the edges also serve as the edge features as with the synthetic network. Table 32 shows the weight values on the first ten connections in the network.

Src	Dst	Weight
0	1	0.3184510360
0	2	0.5512145529
0	3	0.2274158522
0	4	0.2669188689
0	5	0.4754494733
0	6	0.8862627361
0	7	0.1604260538
0	8	0.7459807864
0	10	0.5892903561
0	11	0.4781588875

Table 32: Zachary's Network Edge features

- **Real Network - Neural Network module**

The NN module used in the real network is identical to the NN module used in the synthetic network in terms of composition - 3-Layer GraphSAGE (1 input layer and 2 hidden layers). The differing network sizes between the synthetic network and the two real networks see them have input feature sizes but the same output feature size.

- **Real Network - Output module**

Responsible for handling the outputs from the NN module post training the hyperparameters and composition are like those used in the synthetic network with an Adam Optimizer used to perform model optimization, learning rate of 0.01 and cross entropy used as the loss function to compute the loss in the model.

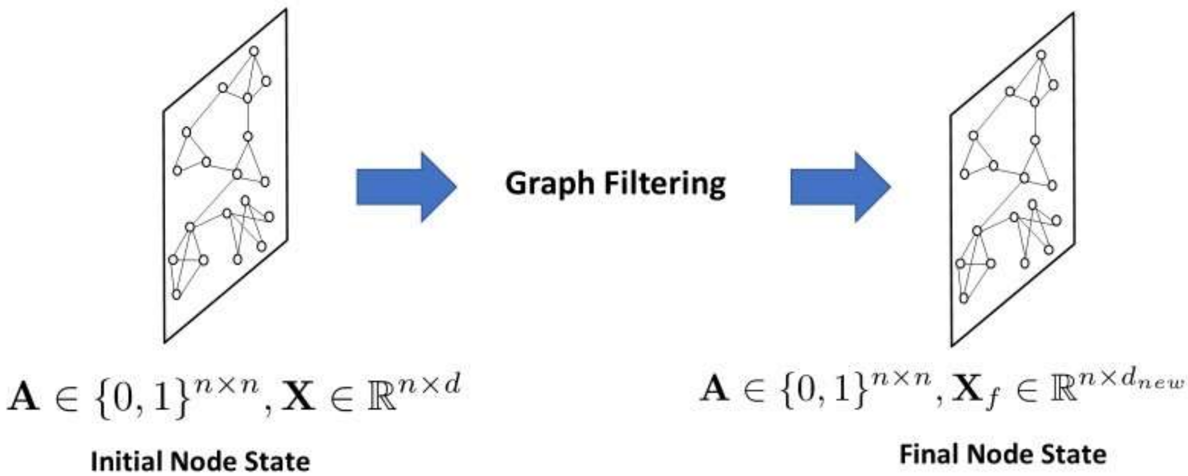

Figure 41. Graph Filtering

Figure 41 shows graph filtering analogous to user-level representation learning operation which refines the features of users in the network.

Figure 41 shows the graph filtering operation which extracts high-level user representations, transforms, and aggregates such representations before propagation throughout the graph.

6.4 Results

The representation learning operation in this model performs multiclass node classification. The learning operation done at the node level works with node attributes and features as inputs into the model where an input outputs a vector-learned representation for each node in the graph while preserving the individual node attributes and graph structure even in the presence of transformation. The learning approach used in the model classification is similar to that of the previous model - the semi-supervised learning. The simulation results from the synthetic network are validated by the results obtained from the two simulations in the real-world network (see Appendix C for link to full simulation codes).

6.4.1 Synthetic Network

Rationale: understanding the results and establishing if they conform to the outputs expected from graph data given the model parameters and assumptions.

The graph network features 2 nodes as the seed nodes. The network output identifies nodes (0 and 1) as the super node, node 0 has an in-degree value of 62 being connected to most nodes in the network and node 1 has an in-degree value of 42. Node 0 has an in-degree connection form every other user in the network excluding node 1. Nodes ($3, 12$ and 77) are created and established as the mirror users having in-degrees of (14, 15 and 23) respectively as seen in Table 33 established from querying the graph structure.

User	**Class**	**Indegree**
0	Super	62
1	Super	42
3	Mirror	14
12	Mirror	15
17	Mirror	23

Table 33. Most Connected Nodes in the Synthetic Network

All other nodes in the network are identified and classed as independent users with either of the sub-class (ignorant and informed) assigned randomly but matching the belief profile types created based on the model definitions. The data of the nodes and edges in the graph network is printed out in a table generated as shown in Figure 42.

	Id	Class	Belief	Bias
0	0	Super	0	0
1	1	Super	1	1
2	2	Informed	2	1
3	3	Mirror	0	0
4	4	Informed	2	1
..	..	...	...	...
75	75	Ignorant	0	0
76	76	Ignorant	1	0
77	77	Mirror	1	0
78	78	Ignorant	0	0
79	79	Informed	2	1

Figure 42. Node Details - SN

The training process of NetTv3 bears some similarities with that of NetTv2. The node attributes and edge weights are fed into the graph as the input features in the graph used for the classification process. **The dataset is divided into sets - training (60%), validation (20%) and test (20%)**. For initialization, being a node classification task, several nodes are activated and fed with a learnable embedding. For the user classes, the values are 0 (ignorant nodes), 1 (informed nodes), 2 (mirror nodes) and 3 (super nodes) (see Appendix C). The tensor of each attribute shows 80 tensor entries indicating that each entry is an attribute of each node (see Appendix C).

Model training involves three sets of simulations each using an attribute (class, belief, and bias) of the nodes, and the training process mirrors that of NetTv2. As with NetTv2, network training is achieved using the network; a conventional Adam optimizer is used. Using the features captured during the learning operation, the classifiers task is to classify the nodes in the graph network into one of the label classes based on their adopted beliefs while considering their internal state - initial belief state and bias state.

For the first simulation, the bias attribute is used as the training feature for the nodes in the graph for learning an embedding that is updated for all nodes in the network. At the training cycle 1 (epoch 0) shown in, nodes in the graph are initialised with their various bias states which serve as their initial states as seen in Figure 43. This generates an initial data set which defines the initial representation of graph data, assigning attributes to nodes based on their features as defined in the dataset. The plotted node positions in the network are randomly generated as seen in the figure below but do not affect the internal dynamics of the individual nodes as their private states based on the bias attribute remain the same as defined. The input layer receives the node features and edge weights which defines an initial representation of the graph data and becomes the input to the GraphSAGE (2 hidden layers).

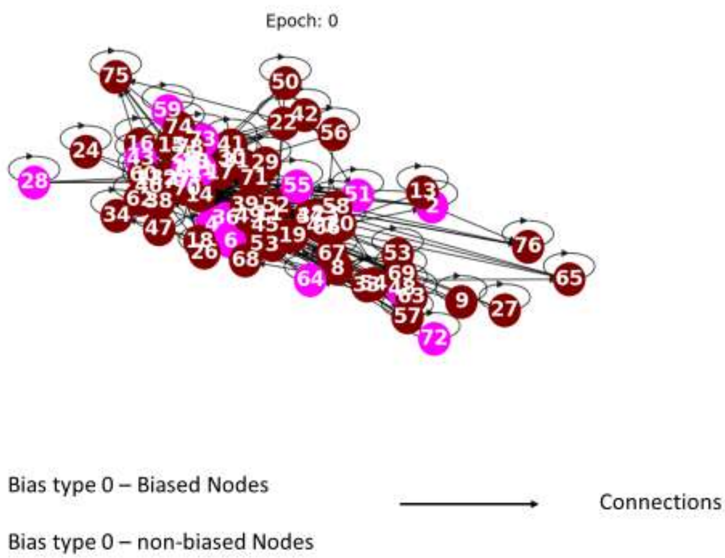

Figure 43. Synthetic Network at Epoch 0 for Bias Feature
Figure 43 shows the states of nodes in the synthetic network at Epoch 0 during training.

Using this, the idea is to learn a representation of the nodes within the graph at each time step while considering the bias states based on the defined values. The layers encode the information on the structure of the network, then exploit this information to update the initial representation of nodes and their edges.

At epoch 99 shown in Figure 44, the nodes in the network output shows feature representations that separate users in the network into their distinct bias profiles as indicated by the distinct colours of the nodes as seen in – the node states have been updated and affine transformation of the network has taken place which sees users' positions in the network being relative to their profiles. The nodes are clustered together into two distinct groups based on the states they have adopted with the node clusters at both ends of the graph network indicating strong attachment to this final state. Message passing allows nodes to be updated with neighbouring node states up to three hops away and these state updates can affect a node’s final state.

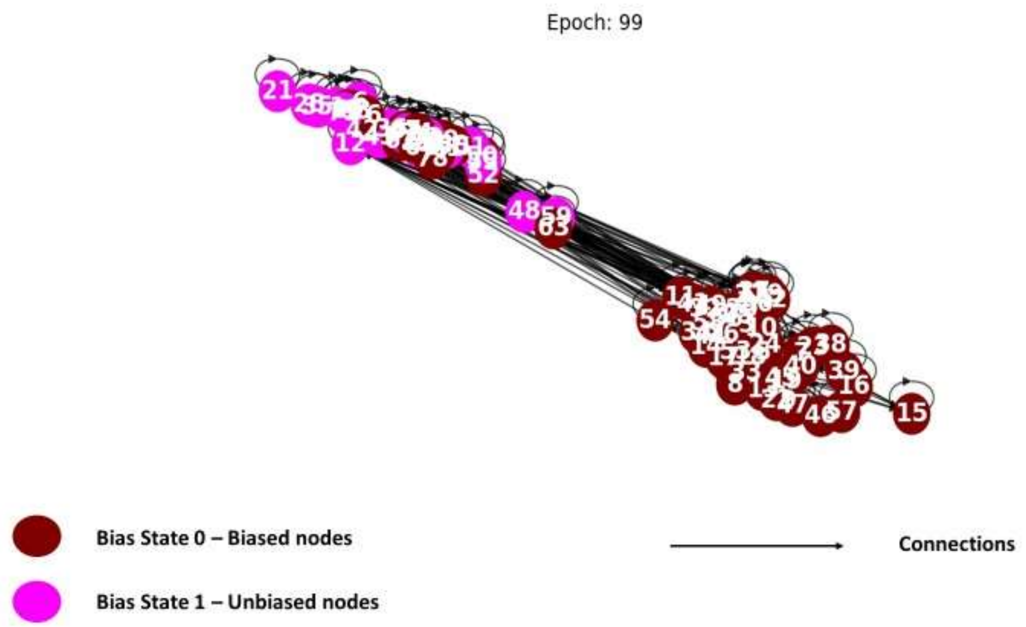

Figure 44. Synthetic Network at Epoch 99 for Bias Feature
The image shows two types of nodes differentiated by colour: 'Bias State 0' in red, indicating biased nodes, and 'Bias State 1' in pink, indicating unbiased nodes. The convergence of nodes in clusters suggest a high level of affinity between these nodes.

With this being a multi-class classification problem – there are two output user profiles when simulated with the bias feature being used as the node feature. Probabilities for the node attributes is computed with the output being a value of 0 and 1 for the bias attribute.

From the evaluation metric - the cross entropy which combines the SoftMax and negative likelihood loss in a single function, the loss is computed only on the nodes in the training set (train masks) for the bias. The accuracy is computed on the training, validation, and test sets with the best accuracy values for the validation and test sets taken. The model loss converges around 40 epochs as seen in Figure 45 with best training accuracy of 1.0, validation accuracy of 0.75 and test accuracy of 0.800 as seen in Table 34.

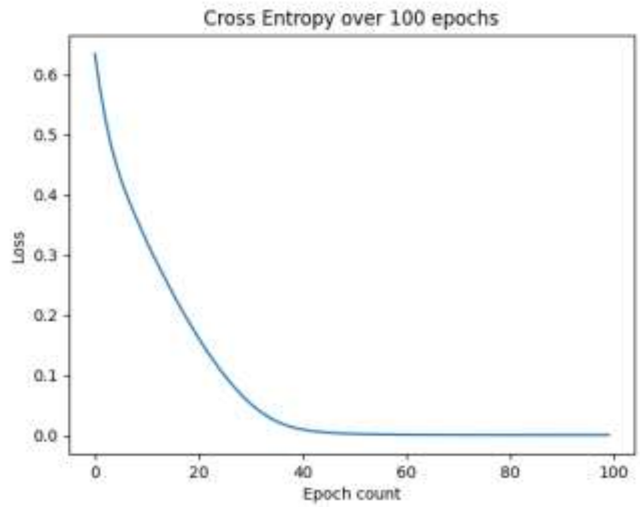

Figure 45. Loss over the training epoch count - Bias Attribute
The sharp decline in loss at the beginning indicates rapid learning or improvement, which gradually levels off as the model approaches a minimum loss.

Epoch Count	Loss	Training Accuracy	Validation Accuracy	Test Accuracy
0	0.6720	0.5750	0.7000	0.8000
10	0.3145	0.7750	0.7500	0.8000
20	0.1430	0.9500	0.7000	0.6500
30	0.0454	1.0000	0.5500	0.3500
40	0.0054	1.0000	0.5000	0.3000
50	0.0012	1.0000	0.6000	0.3000
60	0.0005	1.0000	0.6000	0.2500
70	0.0003	1.0000	0.6000	0.2500
80	0.0003	1.0000	0.6000	0.2500
90	0.0002	1.0000	0.6000	0.3000

Table 34. Loss and Accuracy Values for Bias Attribute

For the second simulation set, the belief attribute is used as the training label for the nodes in the graph for learning an embedding that is updated for all nodes in the network. The internal dynamics for the network training is similar as when training is done using the bias feature. Nodes in the graph are initialised with their initial belief states at epoch 0 providing the initial data which defines the initial representation of graph data with full training at epoch 99, the network has been trained and the output separates nodes in the network into clusters profiles based on the belief attribute.

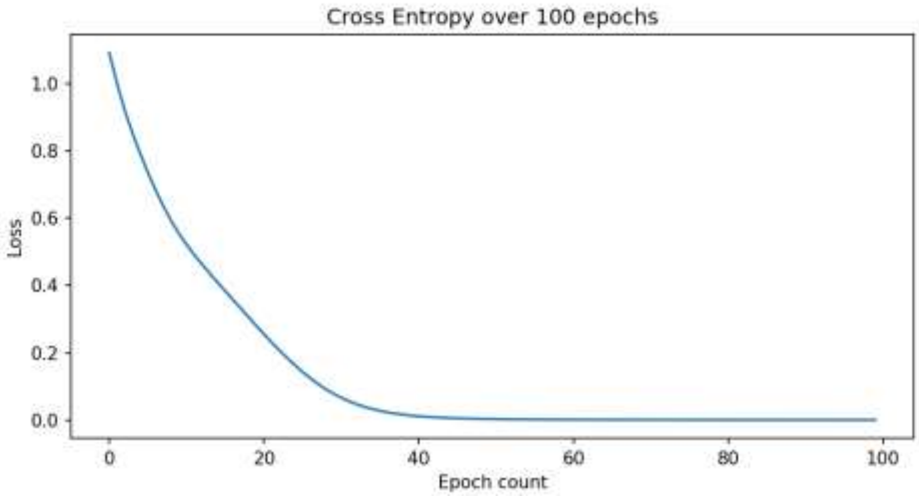

Figure 46. Loss over the training epoch count - Belief Attribute
The graph highlights a steep improvement in the initial epochs, with the loss quickly decreasing and then plateauing, indicating that the model may have reached a point of diminishing returns in learning from the data as it approaches epoch 100

Epoch Count	Loss	Training Accuracy	Validation Accuracy	Test Accuracy
0	1.1868	0.3250	0.2500	0.2000
10	0.5331	0.6500	0.7500	0.7500
20	0.2273	0.6000	0.7500	0.7000
30	0.0670	0.5750	0.7500	0.4500
40	0.0169	0.5750	0.7000	0.4000
50	0.0051	0.5750	0.7000	0.4000
60	0.0024	0.5750	0.7000	0.4000
70	0.0015	0.5750	0.7000	0.3500
80	0.0011	0.5750	0.7000	0.3500
90	0.0009	0.5750	0.7000	0.3000

Table 35. Loss and Accuracy Values for Belief Attribute

Similar evaluation parameters are used to evaluate the model performance for the belief attribute as with the bias attribute - the model loss converges around 40 epochs seen in Figure 46 with top training accuracy of 1.0, validation accuracy of 0.75 and test accuracy of 0.800 as shown in Table 35. Probabilities for the node attributes are computed values of $1, 2, 3,$ and 4 for the belief attribute.

For the third simulation set, the class attribute is used as the training label for the nodes in the graph for learning the same embedding that is updated for all nodes in the network. At the epoch 0 nodes in the graph are initialised with their initial class states providing the initial data which defines the initial representation of graph data. At epoch 99, the network output shows feature representations that separate nodes in the network into clusters based on their class attributes (see Appendix C for full results). Figure 47 and Table 36 show the loss values and accuracy values as well as the cross entropy.

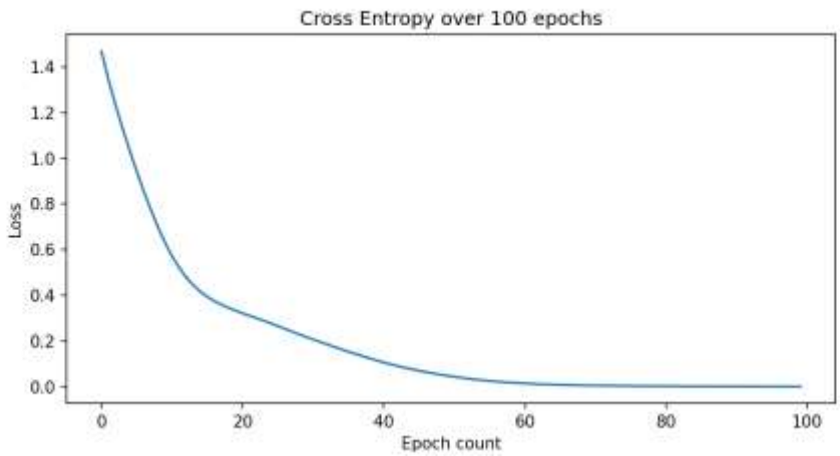

Figure 47. Loss over the training epoch count - Class Attribute

The graph shows a sharp decline in loss at the beginning which suggests rapid learning, which tapers off as the model approaches an optimal level of error minimisation, stabilising towards the end of the training period.

Epoch Count	Loss	Training Accuracy	Validation Accuracy	Test Accuracy
0	1.5181	0.2250	0.2000	0.4000
10	0.6422	0.8000	0.7000	0.7000
20	0.2810	0.9000	0.6000	0.7500
30	0.1583	0.9500	0.5000	0.7000
40	0.0777	0.9750	0.6500	0.7000
50	0.0280	1.0000	0.6500	0.6500
60	0.0112	1.0000	0.6500	0.5000
70	0.0050	1.0000	0.6000	0.5000
80	0.0028	1.0000	0.6000	0.5000
90	0.0019	1.0000	0.6000	0.5000

Table 36. Loss and Accuracy Values for Class Attribute

6.4.2 Real Network

Rationale - The aim of the synthetic model was to perform multiclass node classification under a semi-supervised learning setting to address a representation learning problem in an OSN. Zachary's karate club dataset is used as the real-world network to validate the performance and results of the synthetic network. The dataset is

modified to mirror the internal structure of the synthetic network dataset in terms of node attributes and weights on node connections.

The Zachary's Karate club dataset is the dataset used to validate the synthetic network results from the initial training set. The dataset sees the nodes represent members of a karate club and the edges, their mutual relations. In the original dataset, the users have no attributes associated with them. These are created to mirror the internal structure of nodes in the synthetic network. Self-loops are added to enable effective user state updates across all time steps.

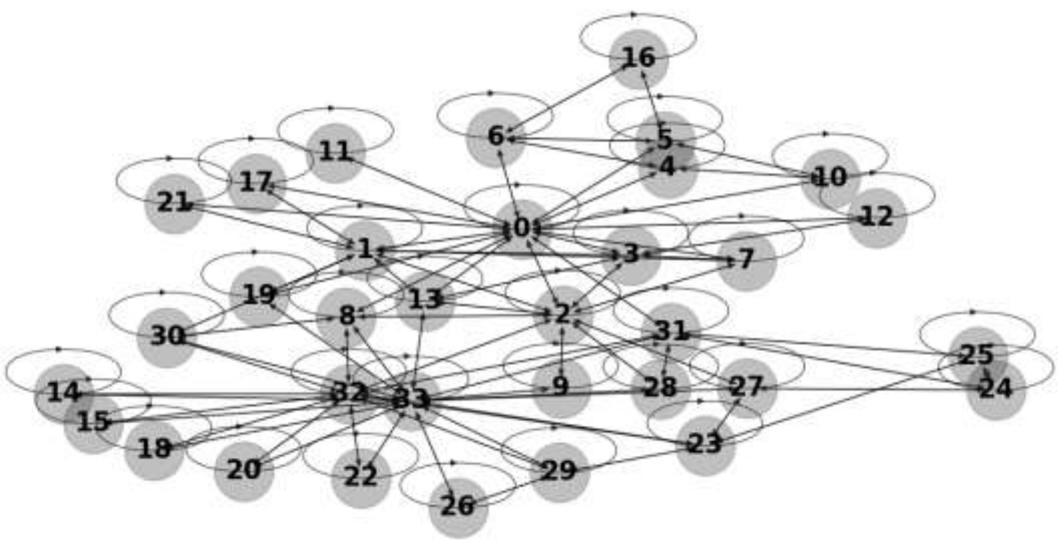

Figure 48. Zachary's Karate Club network

Figure 48 shows a visual of the generated Zachary's Karate club network.

The graph network shown in Figure 48 features 2 nodes as the seed nodes. The network output identifies users (0 and 1) as the super users, user 0 has an in-degree value of 33 being connected to most nodes in the network and user 1 has an in-degree value of 17. Node 0 has an in-degree connection form every other user in the network. Users ($3, 8$ and 23) are established as the mirror users having in-degrees of (7, 7 and 11) respectively.

As with the synthetic network, the node and edge features assignment as the graph input and class label conversion to integer values are identical. The node attributes and edge weights are fed into the graph as the input features in the graph used for the classification process. Features are confirmed by printing the tensors of each attribute and it shows 34 tensor entries with each entry indicating the feature of each node (See Appendix C). Simulations are run using each feature as the input into the network. The dataset is divided into sets created as masks with nodes assigned - training (60%), validation (20%) and test (20%), like the synthetic network.

As with the synthetic network, three simulation rounds are done. The first simulation uses the bias attribute as the network label. At epoch 0 (training cycle 1) as shown in Figure 49, all users in the network are initialised with their initial profile state. This generates an initial data set which defines the initial representation of graph data, assigning features to users. The input layer defines the initial representation of graph data, which becomes the input to the GraphSAGE layer(s).

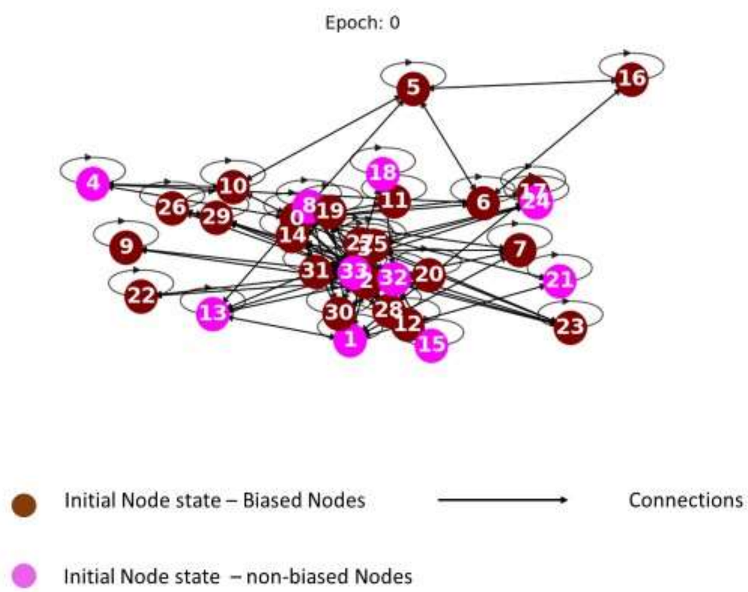

Figure 49. Real Network - Zachary's at Epoch 0
Figure 49 shows the states of nodes in the synthetic network at Epoch 0 during training.

At epoch 99 shown in Figure 50, the nodes in the network show representations that cluster nodes in the network based on the final state adopted. In the RN simulation node feature representations were observed to mirror that of the nodes in the synthetic network for the bias attribute of the users created as part of the simulation. The nodes are clustered into two distinct groups at both ends of the graph. The effects of updates with neighbouring node states via message passage can be seen in the graph as some unbiased nodes are observed in the biased node cluster. Taking node 18 as an example, which has an initial unbiased node state with connections - outbound (nodes 32 and 33 with edge weights of 0.50 and 0.40 respectively) and inbound (nodes 32 and 33 with edge weights of 0.30 and 0.90 respectively), being a part of the biased nodes cluster at epoch 99 indicates that the final states its connections - nodes (32 and 33) which are also affected by their ego networks influences the final state of node 18.

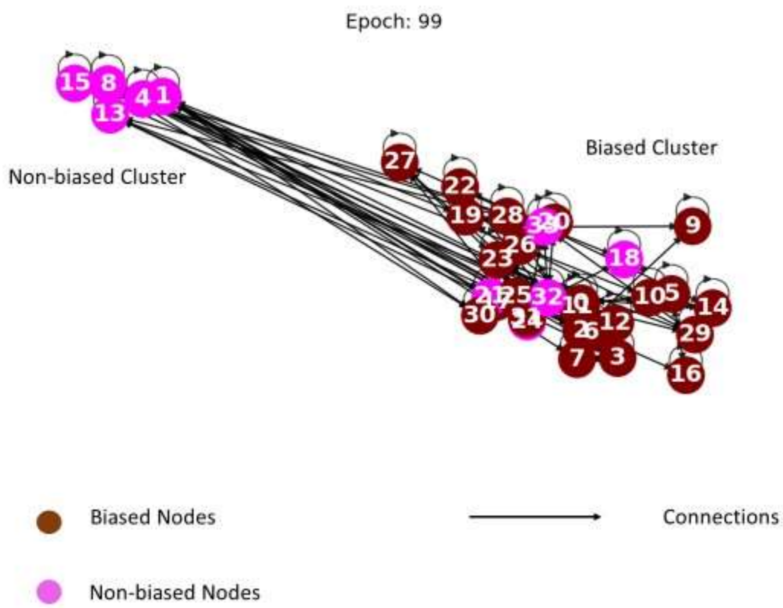

Figure 50. Real Network -Zachary's at Epoch 99
The image shows the network largely split into two clusters. The close convergence of nodes in clusters suggest a high level of affinity between these nodes.

Similar to the synthetic network, the cross entropy (Figure 51) is used as the evaluation metric for the simulations, the loss is computed only on the users in the training set (train masks - 50%) for the bias. The accuracy (Table 37) is also computed on the training, validation, and test sets with the highest accuracy values for the validation and test sets taken.

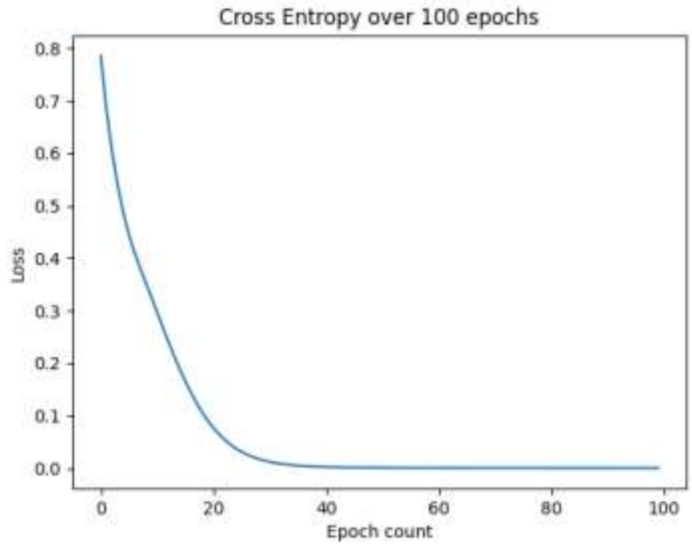

Figure 51. Loss over the training epoch count - Bias Attribute
The sharp decline seen in the graph indicates rapid learning at the initial stages, which plateaus as the model begins to converge to a minimal loss value. This suggests that the model is becoming more accurate in its predictions as training progresses.

Epoch Count	**Loss**	**Training Accuracy**	**Validation Accuracy**	**Test Accuracy**
0	0.6322	0.6471	0.5000	0.5556
10	0.2274	0.9412	0.6250	0.7778
20	0.0456	1.0000	0.3750	0.4444
30	0.0059	1.0000	0.2500	0.3333
40	0.0015	1.0000	0.2500	0.3333
50	0.0007	1.0000	0.1250	0.3333
60	0.0005	1.0000	0.1250	0.3333
70	0.0004	1.0000	0.1250	0.3333
80	0.0003	1.0000	0.1250	0.3333
90	0.0003	1.0000	0.1250	0.3333

Table 37. Loss and Accuracy Values for Bias Attribute - Real Network

For the second simulation set, the belief attribute is used as the training label for the nodes in the graph network, learning the same embedding (feature a that is updated for all users in the network. At the epoch 0, nodes in the network are initialised with their initial belief states providing the initial data for defining the initial representation of graph data. The network output at epoch 99 shows feature representations that separate nodes in the network into their distinct belief states in clusters. Figure 52 and Table 38 show the Cross-entropy loss and training accuracies respectively.

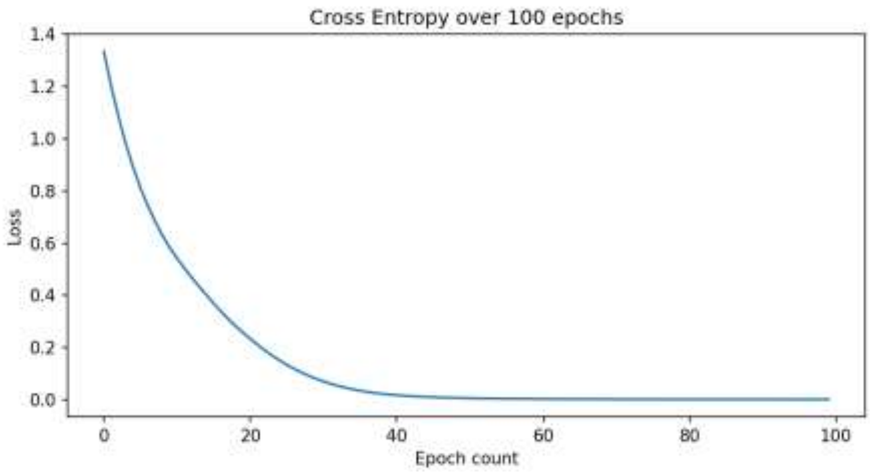

Figure 52. Loss over the training epoch count - Belief Attribute
The graph shows rapid learning at the initial stages converging to a minimal loss value.

Epoch Count	Loss	Training Accuracy	Validation Accuracy	Test Accuracy
0	1.3346	0.1765	0.2500	0.2222
10	0.5422	0.7647	0.6250	0.6667
20	0.2332	0.9412	0.5000	0.7778
30	0.0697	1.0000	0.3750	0.7778
40	0.0172	1.0000	0.2500	0.7778
50	0.0056	1.0000	0.2500	0.7778
60	0.0027	1.0000	0.2500	0.7778
70	0.0018	1.0000	0.2500	0.7778
80	0.0013	1.0000	0.2500	0.7778
90	0.0011	1.0000	0.2500	0.7778

Table 38. Loss and Accuracy Values for Belief Attribute - Real Network

For the third simulation set, the class attribute is used as the training label for the nodes in the graph network, learning the same embedding (feature a that is updated for all users in the network. The simulation results are shown in Figure 53 - cross entropy loss showing a model convergence at around 40 epochs and Table 39 showing a training accuracy of 1.0 and test accuracy of 0.778.

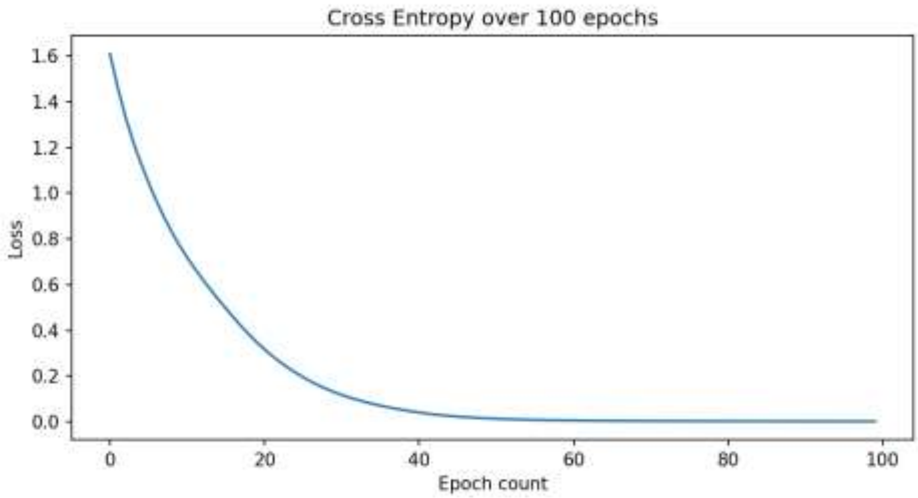

Figure 53.Loss over the training epoch count - Class Attribute
The graph shows the learning converging to a minimal loss value at around 40 epochs.

Epoch Count	**Loss**	**Training Accuracy**	**Validation Accuracy**	**Test Accuracy**
0	1.6101	0.1765	0.3750	0.2222
10	0.7176	0.8235	0.5000	0.5556
20	0.3162	0.9412	0.5000	0.7778
30	0.1172	1.0000	0.6250	0.7778
40	0.0402	1.0000	0.5000	0.6667
50	0.0128	1.0000	0.5000	0.5556
60	0.0053	1.0000	0.5000	0.5556
70	0.0030	1.0000	0.5000	0.6667
80	0.0021	1.0000	0.5000	0.6667
90	0.0017	1.0000	0.5000	0.6667

Table 39. Loss and Accuracy Values for Class Attribute - Real Network

6.4.3 Discussion of Model Results

Using node classification, the goal of this model was to make predictions on each individual node in the network - predicting the state of nodes in the network. Using a GNN framework the embeddings of nodes used in the network are trained in a way that makes them task specific and suited to the multiclass classification task leading to a much better performance. The model exists as an inductive model as it learns the same neural networks on all the nodes and edges in the graph network.

The dataset of the synthetic network was created to allow the network to simulate conditions that are found in an OSN with respect to differing node features/attributes as well as initial node states and connections. Like the synthetic network, the dataset of the real network was modified to allow the network to simulate conditions that are found in an OSN with respect to differing user attributes/features which were created in modifying the dataset. This allowed for the assumptions and definitions put forward in the model to be tested and validated. The model is evaluated by reviewing performance on classification and the model accuracy.

- **Model Performance**

To answer the research question – "*How might Artificial Intelligence (AI) algorithms be employed to mitigate the spread of irrational information?",* the simulation used super nodes with differing states dependent on attribute type as AI agents and differing features attributes created for each node in the network. These attributes together with additional data - neighbouring node states to learn the final states of each node for an embedding representative of a particular belief. With the states of all nodes in the network known, it is possible to show influencing exploits in a network in terms of the beliefs adopted by regular users. Hence, getting access to users' private information is a way to promote the spread of misinformation through such a user.

This can be further understood by reviewing node states. To understand the change in node states via the adoptions of the learnable embeddings within the network, the states of randomly selected nodes across both networks are reviewed. The focus is on the tensor shape of these nodes. The inputs, outputs, and transformations within neural networks are all represented using tensors and these describe the node properties in the network. The simulation uses the bias attribute as the node label with the defined values representing the ground truth of each node. The node features are a two-dimensional matrix, and each row is the feature representation of each node.

Regarding the final state adopted by nodes, using the bias attribute as the node feature as seen in Fig. 54, the states of randomly selected nodes ($3, 6, 12$ and 50) for the synthetic network and ($3, 9, 17$ and 33) for the real network (RN) over a simulation run are reviewed. Node tensors are used to establish the internal states of nodes pre-training and post-training with respect to the learnable embedding - an embedding size of 8 was used. The pre-training (input data - bias attribute) and post-training (computation result) tensor values for the selected nodes can be seen in Table 40 and Table 41 indicating the change in node states over the training duration.

Nodes	Attributes (class, bias)	Pre-training Tensor (size 8)	Post-training Tensor (size 8)
3	Mirror - 0 - 0	[-0.2597, -0.0013, 0.0105, -0.0520, -0.2320, 0.2276, 0.0462, 0.1159]	[-0.2264, -0.1154, -0.2108, 0.1786, -0.1017, 0.5144, -0.1824, 0.3531]
6	Informed - 2 - 1	[-0.0466, -0.2280, -0.2275, 0.2549, 0.1301, 0.0158, -0.0752, 0.0124]	[0.1281, -0.0493, -0.1097, 0.1014, 0.0657, 0.2470, 0.1581, -0.1128]
12	Ignorant - 0	[-0.1304, 0.1442, -0.0472, -0.0588, -0.2266, -0.1827, -0.1258, -0.1378]	[-0.4072, 0.2796, -0.0877, -0.1566, -0.4906, -0.1391, -0.0373, -0.0667]
50	Ignorant - 0	[0.1443, -0.1789, -0.1258, 0.0405, 0.2020, -0.0732, -0.1063, -0.2201]	[-0.0301, -0.1627, -0.0211, 0.0721, 0.0482, 0.0983, -0.1983, -0.2880]

Table 40: Random Nodes (Synthetic Network) - Internal state

Nodes	Attributes (class, belief, bias)	Pre-training	Post-Training
3	Mirror - 0	[-0.1278, 0.2112, 0.2471, 0.2608, -0.0226, -0.3335, -0.3537, 0.0678]	[-0.3772, 0.5300, 0.1004, 0.0050, -0.0296, -0.7309, -0.6136, -0.2880]
9	Informed - 1	[0.0534, 0.3325, -0.2435, -0.2751, 0.3184, 0.1790, 0.1992, 0.0874]	[0.2779, 0.5605, -0.5514, -0.0374, 0.2299, 0.1179, 0.2338, 0.0780]
17	Ignorant - 0	[-0.2708, 0.2033, -0.1107, 0.2560, -0.0915, -0.1343, -0.1060, 0.2048]	[-0.7432, 0.1370, -0.0337, 0.0711, 0.1869, 0.2448, 0.3053, 0.3315]
33	Ignorant - 1	[-0.3417, 0.0626, -0.3644, 0.0429, -0.2085, 0.0629, -0.1232, -0.2011]	[-0.2389, -0.0381, -0.7405, -0.2006, -0.5032, 0.2614, -0.2943, -0.5223]

Table 41: Random Nodes (Real Network) - Internal states

As a function of the NN module, each layer of the NN module computes new node representations by aggregating neighbour information. Using the information of the structure and attributes of a node’s ego network - nodes around and within the node's vision (3 hops away in the model as it uses a 3-layer NN) it learns functions to compute the embeddings of the said node and used in the node state prediction. Referring this to the network training, for the selected nodes - $3, 6, 50, 72$ (Synthetic network) and $3, 9, 17, 33$ (Real network), it is seen that the state of the nodes for the learnable embedding changes over the training run. This indicates that a node's final adopted state is based on and informed by not only the node's features but also the extent of their neighbourhood in relation to their connections and the seed (super) nodes initiating the process.

From the results of the synthetic network, it can then be inferred that the embeddings learnt are done in an end-to-end manner and that the predictions for a given node are a function of the node's features and its immediate neighbourhood (ego network). This indicates that the internal state of a node influences its belief adoption in a network.

The following key observations can be inferred from both networks from the model results as informed by the node states:

▪ Super users are observed as being crucial to the network structure.

In real-world OSNs, super users are celebrities, politicians, influencers etc and often use a variety of ties (silent social relations, advice, friendship relations, coercion, and manipulative relations) to induce other users within their sphere of influence in the network to adopt a particular information/belief. In the research project, the attributes of super users' mirror that of opinion leaders in OSNs who due to their prominence in the number of incoming connections they have can have significant influence in the network.

▪ Adopting agents are observed in the network.

A feature of opinion leaders as seen in the network's super nodes is their influence in the network on other users leading to several other potential adopters. To understand this, one can look at information diffusion from the context of the opinion leaders. Opinion leaders often can be most exposed to information in the network media and more aware of the current trends as well as being hubs in terms of connections, are able to influence others to follow their views with the help of other users in the network who in turn influence others.

- **Model Accuracy**

The model performance was evaluated using cross-validation which sees the dataset split into parts - "masks" in the simulation, to fine-tune its performance. The datasets were divided into a training dataset - to fit the model, a validation dataset - to validate the model's generalisability and the test dataset - to provide an unbiased assessment of the final model fit.

	Belief Attribute	Bias Attribute	Class Attribute
Training accuracy	1.00	1.00	1.00
Validation Accuracy	0.80	0.85	0.81
Test accuracy	0.70	0.73	0.75

Table 42: Best Accuracy Values of the Dataset

Reviewing the accuracy values of the various masks for the datasets as seen in Table 42, it can be deduced that the model performs reasonably well given the small size of the dataset. The accuracy values are the average values for the simulation over 10 runs for each label. The high training accuracy indicates that the model algorithm learnt specific rules that allows it to generalise very well on the training set. The accuracy values of the validation and test set indicates that the hyperparameters used in the model were effective and that it generalises well on unseen data. During training, dropouts were not used to allow for the full training run to detect overfitting or underfitting. The loss is calculated on the training set and is an indication on model performance for that set.

Cross-validation is also performed as part of the model simulation, splitting the dataset into sets - training, validation, and test, to fine-tune its performance. Like the synthetic network the accuracy values are the average of values for the simulation over 10 runs for each label.

	Belief Attribute	Bias Attribute	Class Attribute
Training Accuracy	1.00	1.00	1.00
Validation Accuracy	0.78	0.82	0.80
Test Accuracy	0.73	0.78	0.74

Table 43: Average Accuracy values

Reviewing the accuracy values of the various masks for the dataset shown in Table 43, the model algorithms, high training accuracy indicates high levels of generalisation on the training set. The accuracy values of the validation and test set indicates that the hyperparameters used in the model were effective and that it generalises well on unseen data.

Training was done for the full run of 100 epochs. The cross-entropy function used for loss was calculated on the training set (0.6) of the dataset.

As a function of the number of layers in the NN module, the user's ego network - users around and within the node's vision (3 hops away) informs the updating of the user's state on which belief to adopt while also having an anchor in the form of their defined attributes. Referring this to nodes $3, 9, 17, 33$, it is also seen that for the learnable embedding, the final node state is based on and informed by not only the node's features but also the extent of their neighbourhood in relation to their connections and the two super users.

The results from both simulations shown in Table 44 are also evaluated using the overall model accuracy values to further determine their effectiveness and to determine where the model needs improvement. The accuracy results are evaluated against a baseline model (see Appendix D).

Network	**Accuracy**
Synthetic Network	85.67%
Real Network	88.23%

Table 44: Accuracy Values of Model Networks

The model was implemented on a set of unique assumptions and on the original definitions put forward as part of this research. Using these assumptions and simulating a user representation learning operation, the results from the model showed the following:

• Adaptability -

The networks were constantly evolving; the simulation shows a slightly different model is learnt by the algorithm each time it runs but the relations established during the learning process remain largely the same.

• Neighbourhood aware -

Users in the network are aware of their neighbourhood which also has an effect on their states. Links between nodes in social networks are not random; instead, they often show some sort of connection between the people the nodes represent. A link can show a degree of resemblance between the linked people and provide enough details to be an effective input for a learning system. Information derived from the attributes of the nearby users should be useful in predicting the label of a user as they form a part of the message passing and updating process at each time step.

• Generalisation and Convergence -

The model used a small dataset size and the algorithm demonstrated good generalisation and loss convergence.

In the context of this research several consistencies to real life social networks can be inferred/observed from the results of the simulation. One of such is the information diffusion cycle process as compared to that found in real-world OSNs. Users who decide to adopt information/belief in a network are doing so either dependently (i.e., they received information that other people had adopted the same information) or independently but both are often informed by the user's private belief. Users who adopt an information/belief dependently do so based on neighbourhood users adopting the information and assume by default that such adoption is a strong signal of the value of the information. The information received can be either local - their ego network or global - outside of their ego network.

One of the model’s constraints is the small dataset size which was necessary to test the model definitions. The multiple runs approach, in which the model's algorithm is performed numerous times on the same dataset, is offered as a solution to the small dataset issue. In other words, the effectiveness of a particular NN design is evaluated repeatedly on a set (many run) for each dataset rather than on a single instance of the method. Their performance markers are then given as aggregate statistics for the entire run, enabling reliable performance comparisons despite the dataset's size limitations. This aids in quantifying the various effects of design factors, such as the size of the NN and the length of training, throughout the iterative parameter estimation process.

6.5 Conclusion

The use of featured graphs in NetTv3 allowed the model to demonstrate the diffusion process in a social network and in the presence of several characteristics defined for users in the network which would have an influence on the diffusion process. The model was demonstrated using a synthetic network and validated by applying the algorithm on a real network under the same conditions as the synthetic network. The task of the model task was to perform multiclass classification - using a heterogeneous network, to satisfy the aim of predicting the state to be adopted by nodes in a graph network representative of an OSN to solve the problem of belief adoptions in heterogeneous social networks.

A review of the results from the model simulations showed that for a learnable embedding, the model algorithms can predict a label category for users in a network based on their internal states as defined as their attributes. Using the embeddings assigned to all the nodes in the network and the summation of neighbourhood node information, the model can class users into one of two groups for each attribute simulated. In proving the hypothesis made for question three - "Classification of information propagation path as a way to control misinformation", NetTv3 creates meaningful representations of user states for a graph network using the existing belief structure informed by their private states the simulation results validated the project's assumptions on node characteristics such as number of inbound connections, node state and role in the network as being key to the ability to classify nodes and their propagation paths in a network. This is seen in the case of mirror users who often serve to reinforce the opinions of super users in the network. A real-life example of such users will be forceful agents such as social network bots who interact non-trivially with other users.

7. Conclusion

7.1 Introduction

To summarise the work in preceding chapters, Chapter Two establishes the research background by reviewing existing literature on modelling diffusions in OSNs as well as giving an overview of ABM models, Artificial Intelligence (AI) and Artificial Neural Networks (ANNs). Chapter Three presents the project's research methodology and the model definitions that inform the model design and implementation. Chapter Four presented the first model of this research where the validity of the model novel concept of user classes was established. In Chapter Five, the second model of this research was presented which performed a two-class classification task using representation learning using a Deep model. With the ability of GNNs to integrate node features and graph structure into the learning process, Deep Models, also known as Graph Neural Networks (GNN), allow for the consideration of both the graph structure and the user features in their framework. Chapter Six further extends the model and task in the preceding chapter into multiclass classification using the same approach, introducing more user attributes with more users initialised with these features in line with the user classes already established in the project.

This chapter presents a summary of the research work including a summary of all experiments done. An analysis of the research questions and hypotheses in the context of the various models. The contribution to knowledge of the various models is also detailed as well as a discussion on the problems, limitations, and probable future works.

7.2 Summary of Experiments

Three experiments were done each using a version of model - Network Translation (NetT) created as a framework that addresses the research objectives and questions asked. The model versions - Network Translation Version One (NetTv1), Network Translation Version Two (NetTv2) and Network Translation Version Three (NetTv3). Each version exists as an iteration over the previous version and answers the research questions. Simulations in the first model established a ground truth of some of the model's definitions. The second and third models are used to perform learning operations as an output task in line with the research objectives.

- **Experiment One - Network Translations Version One (NetTv1)**

Experiment One presented the first model of the research which sought to understand and visualise the information diffusion process within a typical social network. Set as an explanatory model, NetTv1 provides a ground truth for the research's model definitions key of which is that of the user classes assumed. Using the model, the aim was to understand and visualise the information diffusion process. The novel user classes were defined and introduced in the model in a functional context with assumptions made regarding the functional and operational context of the model. A synthetic network was used to validate the model's novel definition of user types in a network set along various classes posited as part of the model definitions with their set parameters. Real networks (two datasets) which are representative of social networks, were used to validate the results of the synthetic network. Simulation results were analysed using centrality metrics – indegree, information and eigenvector, providing insight into the network.

- **Experiment Two - Network Translations Version Two (NetTv2)**

Experiment two presented the second model of the research which aimed to classify the users in a network based on the beliefs they adopted. The design architecture introduces the Graph Neural Network (GNN) framework. Network Translation (NetTv2) was presented as an extension of NetTv1 and was implemented using four modules (input module, a labelling module, a neural network (NN) module and an output module) with simulation done in two parts: a synthetic network and a real-world network (Zachary's Karate Club network) which served to validate the results of the prior network. The network is implemented on the basis of Deep Graph Library (DGL), a Python package built for easy implementation of graph neural network model family, on top of other existing DL frameworks (Wang *et al.*, 2019). The model framework allowed for the simulation of user interactions in the presence of user labels analogous to beliefs in this research in what is termed user-level classification which uses graph filtering to generate user representations for each user/node in the network.

- **Experiment Three - Network Translations Version Three (NetTv3)**

Experiment three presented the third model of the research which aimed to perform multi-class user classification introducing heterogeneous users and connections whilst still using a user-Representation learning framework. The design architecture like the previous model was based on the Graph Neural Network (GNN) framework. NetTv3 extends NetTv2 from a binary class classification to a multiclass classification. In addition to adopting the same GNN framework for learning the internal state of the graph nodes and their edges, the model also adopts the definitions and assumptions made with respect to the project framework. NetTv3 was implemented using similar modules as NetTv2 (input module, a labelling module, a neural network (NN) module and an output module) with some internal changes in the modules and simulation done in two parts: a synthetic network and a real-world network (Zachary's which also served to validate the results of the prior network.

7.3 Research Questions and Hypotheses Analyses

The research questions and hypotheses are reviewed and how they are answered by the model simulations are detailed.

1. **Research Question One**

   How do private beliefs influence the diffusion of information within a social network?

This question sought to establish the role user beliefs play in the diffusion of information in a social network.

- **Hypothesis – Hypothesis One**

Users within a social network have differing roles and belief strengths. This can mean different genre of users in the network with differing roles and level of belief strengths This influences interactions between users in the network.

The first hypothesis suggests that a user's private belief is central to their interactions with other users in an OSN. These beliefs shape the view of users across a myriad of opinions and events and are key to a user's decision to endorse/adopt information. The research introduced the novel concept of User classes to classify users based on their roles, belief structure and positions in the network in a class structure - super users, mirror users and independent users (ignorant and informed subclass). The simulation results from experiment one - NetTv1 across three network types (1 synthetic and 2 real-world networks) validated the posited model definition of differing user types in a typical OSN. By analysing self-generated synthetic networks and real networks from data sets all representative of OSNs, the research was able to establish and provide a ground truth for this key definition. Using centrality metrics tools for analysis, it was possible to analyse the state of the networks. Connection patterns between the classes of users established were determined with key users in the networks established and these conformed to the assumptions made in definitions of the user classes supporting the hypothesis put forward - differing user roles mean differing genre of users.

2. **Research Question Two**

How can instances of irrational information diffusion in OSN interactions be identified?

This question sought to establish if it is possible to detect misinformation in daily interactions in social networks.

- **Hypothesis – Hypothesis Two**

Private beliefs and the beliefs systems which inform them play a major role in the connections users establish in a social network and hence significantly influences information diffusion within the network.

The second hypothesis considers that in a typical OSN, there could exist several clusters of users - clustered together by their beliefs. The differences in the belief system of these clusters would see highly polarised interactions between users in the network. The simulation results from experiment one - NetTv1 established user types and their roles in an OSN. Experiment two (NetTv2) extended this to understand network internal dynamics, simulating user interactions in the presence of two differing belief states introducing the use of Graph Neural Networks (GNNs) to represent OSNs. Using a purpose-built synthetic network - with user connections specifically defined as one of two simulation networks, the impact of beliefs in user interactions was explored. Defining specific user connections allowed for the assumptions on the role of beliefs in belief adoption made in the experiment to be tested. For a given information/belief being diffused/adopted in the network, by learning the internal state of users in a network, predicting the final state of users in for that information is made possible. Embeddings replicated vector representations of the internal state of each user - features local to each user which was learnt by users across the network. Results from experiment two established a link between the connections a user establishes in a network and their internal state.

- **Hypothesis - Hypothesis Three**

It is posited that a user can be affected by neighbouring users in the network in terms of the spread of misinformation and belief adoption.

The third hypothesis considers the effects of the neighbouring user beliefs on a particular user. This neighbourhood network - ego network can often serve to reinforce biases in existing beliefs through confirmation bias. Experiment Two established the effects of neighbouring users over a user in the network. Experiment three - NetTv3 established the effects of a user's ego network alongside the effects of their private beliefs. The results of the simulations across both networks showed that final state predictions for a particular user are not just a function of the target user's initial state (private states) but also from state updates over the simulation which are learnt in an end-to-end manner from the ego networks in the network. In the simulation, this is seen in the final states of informed users in the network showing that while diffusion across the network is similar, the contents of the diffusion have differing characteristics due to user interactions informed by not only varied user beliefs but also by the ego network of users.

3. **Research Question Three**

   How might Artificial Intelligence (AI) algorithms be employed to mitigate the spread of irrational information?

This question sought to establish if it is possible to detect misinformation in daily interactions in social networks.

- **Hypothesis – Hypothesis Four**

Classifying information propagation paths in a social network can allow for control in terms of Information diffusion cycle, allowing for false information to be identified early-on. This however will be dependent on the availability of user characteristics as well as on the classification method used.

The fourth hypothesis creates a link between the propagation path and the diffusion of information in the network. Viral information will often take the path of least resistance by propagating through users that adopt the held beliefs. The simulation results from experiment two - NetTv2 established the network internal dynamics, simulating user interactions with users being in either one of two belief states. Experiment three - NetTv3 simulated user interactions using features defined for each user in the network permitting a more diverse set of users. As a solution in classifying information propagation paths, NetTv3 models each user as a node and creates a graphical network of users allowing the adoption of beliefs to be observed in the final state of nodes. By leveraging the structural and graphical properties of users and their differing attributes, from the results, the model was able to determine how specific users spread information, the characteristics of users involved in spreading the information and structure of the ego network of such users - allowing for the delineation of the path to a particular user.

7.4 Contribution to Knowledge

The progressive models used allowed this research to delve deeply into the internal structure of social networks, focusing on the dynamic relationships and dependencies among users. Each model incrementally builds upon the knowledge and understanding of the network's internal dynamics and user interactions, establishing a comprehensive framework for classifying users and understanding their roles and influences within the network. An overview of the contribution to knowledge of the models is shown in Table 45.

Models	Overview of Contribution to Knowledge
NetTv1	The definition of user classes.
	Understanding the role of users in information flow.
	Analysing the links users establish.
NetTv2	Influence of network dynamics on diffusion.
	Effects of user roles and their structural positions on diffusion.
	Role of bias in information flow.
	User beliefs and how they evolve.
NetTv3	Relationship between the links users establish and their beliefs.
	Effects of ego networks on diffusion

Table 45: Summary of Research Contribution to Knowledge

7.5.1 Experiment One - Network Translations Version One (NetTv1)

The task of this model was to establish the internal structure of a social network examining the relationships and dependencies between users in the network. The results from the synthetic network validated the model definition of user types in a network set along various classes posited as part of the model definitions with their set parameters showing that in a randomly generated network, there are users that can be classed as super users, mirror users and independent users. Several contributions to existing knowledge are presented from the results of the simulation:

- **Contribution to Knowledge - NetTv1**

1. **Definition of User Classes**

NetTv1 offers a distinctive method of categorizing users into classes, building on introduction in prior works (Yang, Tang and Leung, 2015; Y *et al.*, 2016; Zhao, Li and Jin, 2016; Pei *et al.*, 2020). The model delineates users into “super users,” “mirror users,” and “independent users,” each class embodying unique roles, characteristics, and network positions, furthering the understanding of internal dynamics within a network's diffusion process.

2. **Understanding Role of Users in Information Flow**

An algorithm introduced by Zhao, Li and Jin, (2016) was expanded upon to identify influential nodes, or "seed nodes," within community structures in social networks. NetTv1 results underscore the prominence of super and mirror users, both crucial to the network’s information flow and influential during diffusion events. Super users' roles, as highlighted in the research, could be linked to concepts of 'influencers' or 'hubs' from other studies (Chen *et al.*, 2014; Yang, Tang and Leung, 2015; Zhao, Li and Jin, 2016).

3. **User Links Analysis**

User links within the network adhere to model definition assumptions for various user classes, with simulation results further validating these links and connections among user classes. The use of Indegree Centrality, Information Centrality, and Eigenvector Centrality is consistent with many network analysis studies (Yang, Tang and Leung, 2015; Y *et al.*, 2016; Zhao, Li and Jin, 2016; Pei *et al.*, 2020) but is applied uniquely to the context of the model’s novel user classes.

7.5.2 Experiment Two - Network Translations Version Two (NetTv2)

Building upon the groundwork laid by NetTv1, the second experiment, NetTv2, further refines the user classification process. The primary task of this model - NetTv2 was user classification using representation learning, where the model predicts the ground truth category of each user in the network based on their features. Through NetTv2, the influence of network dynamics on the diffusion process is examined meticulously, emphasizing the crucial role of users’ structural positions and ego neighbourhoods in shaping the overall network state and the efficiency of public signals within it.

- **Contribution to Knowledge – NetTv2**

1. **Network Dynamics Influence on Diffusion**

Prior works have established the internal structure of a network as playing a role in users having their initial states reinforced or mitigated into a new state (Antal and Balogh, 2009; Jimenez-Martınez, 2015; Yang, Tang, and Leung, 2015; Enders *et al.*, 2021). NetTv2 contributes insights into how network structures impact user influence and information diffusion. The model highlights the significance of users' ego neighbourhoods, showing that public signals within networks are not just dependent on connectivity but also on user states.

2. **Clarification on Users' Role and Structural Position**

Prior research has focused on the importance of influential nodes in information spread (Chen *et al.*, 2014; Zhao, Li and Jin, 2016; Henry, Stattner and Collard, 2017; Pei *et al.*, 2020; Sasahara *et al.*, 2020). This research this understanding, shedding light on super and mirror nodes' roles and how they significantly influence information (or misinformation) spread. The research underlines that not all users uniformly impact network diffusion due to variations in their internal dynamics and positions. Definitions and roles of super and mirror users were further refined, emphasizing their importance in state updates and influence within their respective networks.

3. **Role of Bias in Information Diffusion**

Many studies have tackled cognitive biases (Del Vicario *et al.*, 2017; Sobkowicz, 2018; Fernandes, 2020), but their manifestation in OSNs remains less explored. This research offers empirical evidence on how confirmation bias operates within OSNs, linking individual cognitive processes to collective digital behaviours. The research emphasises the outsized influence of nodes based on their structural position. It's not just about quantity (number of connections) but also about the quality or nature of those connections and the influence they hold. The research demonstrates that users' beliefs in OSNs can be influenced, either strengthened or mitigated, by the interactions they have within their network. This provides a quantified understanding of the psychological phenomenon of confirmation bias within a digital setting.

4. **Dynamic Evolution of User Beliefs**

OSNs are not static; they are evolving ecosystems where users' beliefs change over time based on interactions. While the dynamic nature of social networks is recognised, there's limited literature on how users' beliefs evolve within these networks. This research provides a detailed analysis of belief evolution, emphasising the impermanent nature of beliefs in digital networks. The research illustrates how beliefs can be adopted, altered, or even discarded based on these interactions.

7.5.3 Experiment Three - Network Translations Version Three (NetTv3)

The task of the model - NetTv3 was multi-class user classification using representation learning whilst introducing heterogeneous users in terms of attributes. Model implementation was done within the GNN framework with a similar internal structure to that of NetTv2. To accomplish this task, the model classified the nodes in a network based on their edge information considering the profile states initiated amongst a select group of nodes within the network. Several contributions to existing knowledge are presented from the results of the simulation:

- **Contribution to Knowledge – NetTv3**

1. **Relationship Between User Links and Beliefs**

NetTv3 demonstrates a correlation between users' connections within networks and their belief profiles. Simulation results indicate that consensus beliefs and mass adoption within networks occur when neighbouring agents share similar profiles and belief systems.

2. **Exploration of Ego Network Effects**

Building on the works of works Arnaboldi et al. (2013, 2017), Mcauley and Leskovec, (2014), Bouanan et al*.* (2015), NetTv3 illuminates the role and impact of ego networks in information diffusion across heterogeneous networks. The model highlights how ego networks significantly influence information adoption and diffusion within the network, emphasizing their importance in determining user beliefs and opinions.

7.5 Problems and Limitations

Having provided conclusions for the research question asked and hypotheses posited as answers all while contributing to the field of ABMs, GNNs and social networks, a review of the problems encountered, and limitations of this research is detailed. These should be considered when evaluating this research in its entirety. Some of these are listed below:

▪ Data Size

The data size used across all simulations were small and may not have captured the full extent or relations and dependencies in OSNs. One factor that led to this was the need to define attributes for all nodes (manual labelling) and specify connections needed to test the model definitions and assumptions. A second factor was the research design which was focused on user interactions in a network at the micro-level - hence the need to define user features. Subsequent research with a larger data sample size should fully capture all dependencies and end up with more accurate results.

▪ Computational Resources

Computational resources limited the scope of the simulations in terms of the scale and complexity. There were limitations on the hardware used in terms of processing power that influenced the scale/size of the simulations in the various models.

▪ Functional Context

The functional scope of the three models is limited to applications around social networks, even when generalised, the models were tested using synthetic networks and validated with one real network simulating a real social network environment. This limits their applicability to OSNs and OSNs related applications.

▪ Nature of Graphs

With GNNs being dynamic graphs, it can be a challenge to deal with graphs with dynamic structures as these graphs evolve in terms of their connectivity with connections easily created and destroyed. This also makes finding the best graph generation approach a challenge. There is also the issue of graph embeddings. Applying embedding methods in social networks which can be represented by dynamic graphs can be computationally complex for all graph embedding algorithms, including GNNs. This was factor that also affected the network size used in the simulations. GNNs are also difficult to apply in non-structural scenarios.

7.6 Future Works

Several potential areas/fields exist that can extend the research work done in this project. Some of these fields are detailed.

7.6.1 Serious Games

Games have evolved beyond pure entertainment, with the emergence of Serious Games that aim to effect change through interactive dialogue and narrative. Recent developments in game engines, such as Unreal Engine and Unity, have enabled highly realistic and interactive virtual worlds. These engines incorporate physics simulations, AI-driven characters, and complex narratives, creating immersive experiences for players. These games, with their dynamic architectures, have the potential for personalisation based on player behaviour. Serious Games are designed with the intention of achieving specific non-entertainment goals, such as education, training, or behaviour change. A key feature of Serious Games is their gamification, which is designed to enable "game transfer." Game transfer refers to the idea that the skills, knowledge, or experiences gained in a game can be transferred to real-world situations. This transfer is facilitated through interactive dialogue and narrative elements in the game (Cowley *et al.*, 2008).

Online games have evolved with improved game mechanics, creating highly interactive and engaging virtual worlds. These games are dynamic systems with evolving rulesets and content influenced by player choices during gameplay (Martens, 2013). This dynamic nature allows for personalisation of gaming experiences. In the realm of social networks, information diffusion models like the NetTv3 extension can be applied to understand how news, trends, and opinions spread. Research shows that the spread of information in social networks often follows power-law distributions, with a few influential nodes driving dissemination.

The NetTv3 model framework is applied to modelling information diffusion within an Online Social Network (OSN). Through a set of assumptions and definitions applied, the model allows for an end-to-end framework for simulating social networks in diverse conditions. To accomplish this, an extension to the existing model NetTv3 is posited as an internal state – "belief" learning model. This is also implemented as a graph-based convolution neural network setup; where the input into the network is a graph represented as a matrix of user features, $\boldsymbol{X} \in \mathbb{R}^{\boldsymbol{N} \, x \, \boldsymbol{F}}$, where N is the number of users and $\mathbf{F}$ is the number of input features for each user and $\boldsymbol{A} \in \mathbb{R}^{\boldsymbol{N} \, x \, \boldsymbol{N}}$ represents an adjacency matrix for the graph $\boldsymbol{G}$.

Gamer profiles are introduced, posited as being analogous to belief profiles and formulate the problem - classification of beliefs and prediction of belief adoptions within a gaming clan as a graph. Three player classes are introduced– ***anarchist**, **conformist**,* and ***independent*** players as part of the model definition. ***Anarchist*** players are those who are opposed to learning, and thus should be identified as candidates who require greater attention to change their beliefs. ***Conformist*** players are those whose beliefs are aligned with the aim of the game and have the least friction to the game belief alignments. ***Independent*** players are those whose beliefs are ambiguous or are potentially influenceable to joining either the anarchist or conformist point of view. The concept of gamer profiles extends beyond just classifying players. In modern gaming, player data is used to offer personalized experiences through techniques like recommendation systems. For example, a player's previous game choices, in-game behaviour, and preferences can inform game content recommendations, fostering deeper engagement.

A learnable embedded vector represents the initial input for the neural network, which serves as an interactive narrative. This vector encapsulates the player's style, preferences, and interactions within the game. Each player class is associated with specific attributes, implemented as node and edge features in the graph. This data allows the model to understand how different gamer profiles interact, enabling responsive and adaptive game mechanics based on player abilities. The model's results can help improve the personalisation of games. Assuming that players in a game can communicate or influence each other, and that gameplay can be monitored and measured, the model classifies players into gamer profiles in real-time. This real-time classification enables dynamic changes in game strategy to influence player beliefs. The idea of real-time changes in game strategy based on player profiles aligns with the concept of dynamic difficulty adjustment (DDA) in gaming. DDA algorithms analyse player performance and adapt the game's difficulty level to keep players engaged without overwhelming or boring them.

The integration of gamer profiles, social network dynamics, and advanced machine learning techniques provides a powerful framework for personalising gaming experiences. By classifying players into distinct profiles and adapting game mechanics accordingly, developers can create more engaging and effective Serious Games with the potential for real-world impact. This approach represents a promising direction for the future of gaming and gamification.

7.6.2 Digital Communities

Social network ties form the core of individual interaction, and these happen at the local level rather than the global level (Namatame and Chen, 2016). Numerous elements, including relationships to one's family, tribe, neighbours, classmates, friends, co-workers, education, and geographic area, influence a person's life (Namatame and Chen, 2016). This particular social connection gives people access to knowledge, concepts, and new ideas, which has an impact on their choices, deeds, successes, and relationships. These connections, which frequently coevolve in dependence on one another, were previously founded in real-world groups. The concept of local interactions in social networks is consistent with studies on the strength of weak ties in social network analysis. Weak ties, often formed with acquaintances, can introduce new ideas and perspectives into a person's social sphere. A person's private opinions are frequently influenced by these contacts. Private belief is accepted as a critical element influencing the social connections people make and is even more pervasive in digital networks.

Digital communities have gained increased prominence due to the near ubiquitous nature of online social networks. With diverse users at the heart of these digital communities, it is possible to use an Agent-Based Modelling (ABM) framework to simulate the effects of private-beliefs on interactions in digital communities. NetTv3 model can be extended to simulate these interactions. Using graphs, a representation of the problem is created as a synthetic network and the model is implemented as a graph-based convolution neural network setup. The three user classes – ***super***, ***mirror*** and ***independent*** users (with ignorant and informed as sub types) of the existing model are used to represent different categories of users and interactions in these communities.

NetTv3 was implemented using Deep Graph Library (DGL) Python package (Wang *et al.*, 2019) with the model architectures and hyperparameters following the design from GCN original model (Kipf and Welling, 2017) and features three layers (up to three hops away). To establish the adaptability of digital communities – the number of layers in the neural network can be varied to simulate interactions in different network structures. In real-world scenarios, the results will show interactions in digital communities **as not being bounded by location or geography and able to adapt, from changes in network structure to changes in the states of users.**

The study of digital communities, their dynamics, and the impact of private beliefs within them is essential in the age of online connectivity. Extending the NetTv3 model through Agent-Based Modelling offers a valuable approach to understanding these complex interactions. By exploring how private beliefs shape interactions in digital communities, we can gain insights into the adaptable and ever-evolving nature of online social networks, transcending geographical boundaries and fostering diverse and dynamic connections.

7.7 Conclusion

Motivated by the aim of detecting and classifying misinformation diffusion in online social networks, this research has attempted to model irrational agent beliefs in social networks exploring the relationship between the internal private beliefs of users and the diffusion of misinformation in OSNs. The use of OSNs within the society has gained widespread acceptance as the primary means in the spread of information. At a fundamental level, the management of the ease with which users can create and destroy connections is recognised as one of the most significant challenges in policing social networks (Rainie, Anderson and Albright, 2017).

A good model for modelling misinformation should capture how users interact and influence each other and do this while taking into consideration the unique attributes that users possess. In such interaction models, it is shown that a user with similar beliefs to other users would have a different influence on the belief formation than a user with differing beliefs. Three experiments were conducted using three models with each model an extension of the previous model. Novel definitions and assumptions were introduced to allow for the ability to simulate distinct user profiles with the context of heterogeneous networks using a GNN architecture. With belief and opinion diffusion being at the core of OSNs, recognizing and understanding the internal states of users in these networks will improve the ability to combat the spread of misinformation increasing overall positive experiences of OSNs.

GNNs have been demonstrated to be powerful in learning the internal states of graph data and provides great potential to advance modelling the internal states of users in social networks replicating much more realistic user characteristics and interactions found between users and clusters of users in such networks. The current model achieves user features classification in the form of representation learning operations.

REFERENCES

A brief history of social networks (2018). Available at: https://www.antevenio.com/usa/a-brief-history-of-social-networks/ (Accessed: 17 September 2021).

Abar, S. et al. (2017) ‘Agent Based Modelling and Simulation tools : A review of the state-of-art software’, Computer Science Review, 24, pp. 13–33. Available at: https://doi.org/10.1016/j.cosrev.2017.03.001.

Abdullah, S. and Wu, X. (2011) ‘An epidemic model for news spreading on twitter’, Proceedings - International Conference on Tools with Artificial Intelligence, ICTAI, pp. 163–169. Available at: https://doi.org/10.1109/ICTAI.2011.33.

Acemoglu, D., Ozdaglar, A. and Parandehgheibi, A. (2010) ‘Games and Economic Behavior Spread of (mis) information in social networks ☆’, Games and Economic Behavior, 70(2), pp. 194–227. Available at: https://doi.org/10.1016/j.geb.2010.01.005.

Agha Mohammad Ali Kermani, M., Ghesmati, R. and Pishvaee, M.S. (2021) ‘A robust optimization model for influence maximization in social networks with heterogeneous nodes’, Computational Social Networks, 8(1), p. 17. Available at: https://doi.org/10.1186/s40649-021-00096-x.

Albert, R. and Barabasi, A.-L. (2002) ‘Statistical mechanics of complex networks’, Reviews of Modern Physics, 74(1), pp. 47–97. Available at: https://doi.org/10.1103/RevModPhys.74.47.

Aldridge, C. (2005) ‘Scale-free networks using local information for preferential linking’, in MODSIM 2005 International Congress on Modelling and Simulation, Modelling and Simulation Society of Australia and New Zealand, December. Citeseer, pp. 1196–1202.

Algeo, J. (1989) ‘A Dictionary of Briticisms’, English World-Wide, 10(1), pp. 123–134. Available at: https://doi.org/10.1075/EWW.10.1.11ALG.

Alimov, A., Shabalina, O. and Moffat, D.C. (2019) ‘Development of Digital Game Environments Stimulating Creativity in Engineering Education’. Available at: https://doi.org/10.4018/978-1-5225-3395-5.ch031.

Allcott, H., Gentzkow, M. and Yu, C. (2019) ‘Trends in the diffusion of misinformation on social media’:, https://doi.org/10.1177/2053168019848554, 6(2). Available at: https://doi.org/10.1177/2053168019848554.

Al-Taie, M.Z. and Kadry, S. (2017) ‘Information Diffusion in Social Networks’, Python for Graph and Network Analysis, pp. 165–184. Available at: https://doi.org/10.1007/978-3-319-53004-8_8.

Amini, M. et al. (2012) ‘Alternative supply chain production–sales policies for new product diffusion: An agent-based modeling and simulation approach’, European Journal of Operational Research, 216(2), pp. 301–311. Available at: https://doi.org/10.1016/j.ejor.2011.07.04.

Angelo, G.D.’, Severini, L. and Velaj, Y. (2016) ‘Influence Maximization in the Independent Cascade Model’. Available at: http://ceur-ws.org/Vol-1720 (Accessed: 27 September 2021).

Antal, M. and Balogh, L. (2009) ‘Modeling belief systems with scale-free networks’, Neural Networks, 22(10), pp. 1359–1371. Available at: https://doi.org/10.1016/j.neunet.2009.04.001.

AnyLogic: Simulation Modeling Software Tools & Solutions for Business (2000). Available at: https://www.anylogic.com/ (Accessed: 5 October 2021).

Aparicio, S., Villazon, J. and Alvarez, G. (2015) ‘A Model for Scale-Free Networks: Application to Twitter’, Entropy, 17, pp. 5848–5867. Available at: https://doi.org/10.3390/e17085848.

Arias, E. (2019) ‘How Does Media Influence Social Norms? Experimental Evidence on the Role of Common Knowledge’, Political Science Research and Methods. 2018/02/20 edn, 7(3), pp. 561–578. Available at: https://doi.org/10.1017/psrm.2018.1.

Arnaboldi, V. et al. (2013) Ego Networks in Twitter: an Experimental Analysis, Proceedings - IEEE INFOCOM. Available at: https://doi.org/10.1109/INFCOM.2013.6567181.

Arnaboldi, V. et al. (2017) ‘Online Social Networks and information diffusion : The role of ego networks’, Online Social Networks and Media, 1, pp. 44–55. Available at: https://doi.org/10.1016/j.osnem.2017.04.001.

Arndt, J. (1967) ‘Role of Product-Related Conversations in the Diffusion of a New Product’, Journal of Marketing Research, 4(3), p. 291. Available at: https://doi.org/10.2307/3149462.

Aymanns, C., Foerster, J. and Georg, C.-P. (2017) ‘Fake News in Social Networks’. Available at: http://arxiv.org/abs/1708.06233 (Accessed: 1 October 2021).

Bala, V. and Goyal, S. (2000) ‘A Noncooperative Model of Network Formation’, Econometrica, 68(5), pp. 1181–1229. Available at: https://doi.org/10.1111/1468-0262.00155.

Balci, O. (2003) ‘Verification, validation, and certification of modeling and simulation applications’, Winter Simulation Conference Proceedings, 1, pp. 150–158. Available at: https://doi.org/10.1109/WSC.2003.1261418.

Ball, P. (2004) ‘Critical mass: how one thing leads to another’, p. 520.

Barabási, A.-L. (2013) ‘Network science. Chapter 5. The Barabási-Albert Model’, Network Science, pp. 1–45.

Bass, F.M. (1969) ‘A New Product Growth for Model Consumer Durables’, Management Science, 15(5), pp. 215–227.

Bellinger, G. (2004) Modeling & Simulation - An Introduction. Available at: https://systems-thinking.org/modsim/modsim.htm (Accessed: 10 October 2021).

Berners-Lee, T. (1989) The original proposal of the WWW, HTMLized. Available at: https://www.w3.org/History/1989/proposal.html (Accessed: 17 September 2021).

Besta, M. et al. (2021) ‘Demystifying Graph Databases: Analysis and Taxonomy of Data Organization, System Designs, and Graph Queries’. arXiv. Available at: http://arxiv.org/abs/1910.09017 (Accessed: 15 May 2022).

Beutel, A. et al. (2012) ‘Interacting viruses in networks: Can both survive?’, Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 426–434. Available at: https://doi.org/10.1145/2339530.2339601.

Bhatnagar, S. et al. (2017) ‘Mapping intelligence: Requirements and possibilities’, in 3rd Conference on" Philosophy and Theory of Artificial Intelligence. Springer, pp. 117–135.

Bianchi, F.M., Grattarola, D. and Alippi, C. (2020) ‘Spectral clustering with graph neural networks for graph pooling’, 37th International Conference on Machine Learning, ICML 2020, PartF16814, pp. 851–860.

Blank, G. and Reisdorf, B. (2012) ‘The Participatory Web’, Information, 15. Available at: https://doi.org/10.1080/1369118X.2012.665935.

Bohlmann, J., Calantone, R. and Zhao, M. (2010) ‘The Effects of Market Network Heterogeneity on Innovation Diffusion: An Agent-Based Modeling Approach’, Journal of Product Innovation Management, 27, pp. 741–760. Available at: https://doi.org/10.1111/j.1540-5885.2010.00748.x.

Bonabeau, E. (2002) ‘Agent-based modeling: Methods and techniques for simulating human systems’, Proceedings of the National Academy of Sciences, 99(suppl 3), pp. 7280–7287. Available at: https://doi.org/10.1073/PNAS.082080899.

Borgatti, S.P. and Everett, M.G. (2006) ‘A Graph-theoretic perspective on centrality’, Social Networks, 28(4), pp. 466–484. Available at: http://dx.doi.org/10.1016/j.socnet.2005.11.005.

Bouanan, Y. et al. (2015) ‘Simulating information diffusion in a multidimensional social network using the DEVS formalism (WIP)’, pp. 63–68.

Boyd, D.M. and Ellison, N.B. (2008) ‘Social Network Sites: Definition, History, and Scholarship’. Available at: https://doi.org/10.1111/j.1083-6101.2007.00393.x.

Brachman, R.J. (2006) ‘AI more than the sum of its parts’, AI Magazine, 27(4), pp. 19–19.

Bruna, J. et al. (2014) ‘Spectral Networks and Locally Connected Networks on Graphs’.

Business Wire (1995) Beverly Hills Internet, Builder of Web Communities, Changes Name to GeoCities; Monthly Page Views Top 6 Million. | North America > United States from AllBusiness.com. Available at: https://web.archive.org/web/20081211170054/http://www.allbusiness.com/marketing-advertising/marketing-advertising/7191644-1.html (Accessed: 17 September 2021).

Busselle, R. (2017) ‘Schema Theory and Mental Models’, The International Encyclopedia of Media Effects, pp. 1–8. Available at: https://doi.org/10.1002/9781118783764.WBIEME0079.

Butts, C.T. (2009) ‘Revisiting the Foundations of Network Analysis’, Science, 325(5939), pp. 414–416.

Cangea, C. et al. (2018) ‘Towards sparse hierarchical graph classifiers’, arXiv, pp. 1–6.

Cao, X. et al. (2016) ‘Evolutionary information diffusion over heterogeneous social networks’, IEEE Transactions on Signal and Information Processing over Networks, 2(4), pp. 595–610. Available at: https://doi.org/10.1109/TSIPN.2016.2613680.

Capurro, R. and Hjorland, B. (2003) The Concept of Information. Available at: https://repository.arizona.edu/bitstream/handle/10150/105705/infoconcept.html (Accessed: 20 April 2022).

Carmichael, T. and Hadžikadić, M. (2019) ‘The fundamentals of complex adaptive systems’, Understanding Complex Systems, (July), pp. 1–16. Available at: https://doi.org/10.1007/978-3-030-20309-2_1.

Cassauwers, T. (2019) Can artificial intelligence help end fake news? | Research and Innovation. Available at: https://ec.europa.eu/research-and-innovation/en/horizon-magazine/can-artificial-intelligence-help-end-fake-news (Accessed: 11 October 2021).

Castellano, C. (2009) ‘Statistical physics of social dynamics’, pp. 1–58.

Casti, J.L. (1997) ‘Would-be worlds: how simulation is changing the frontiers of science’, p. 242.

Castillo, C., Mendoza, M. and Poblete, B. (2011) ‘Information credibility on Twitter’, Proceedings of the 20th International Conference Companion on World Wide Web, WWW 2011, pp. 675–684. Available at: https://doi.org/10.1145/1963405.1963500.

Chai, Y. et al. (2017) ‘Crowd science and engineering: concept and research framework’, International Journal of Crowd Science, 1(1), pp. 2–8. Available at: https://doi.org/10.1108/IJCS-01-2017-0004.

Chapdelaine, P. and Manzerolle, V. (2021) ‘The Regulation of Media and Communications in the Borderless Networked Society’, Laws, 10(4), p. 78. Available at: https://doi.org/10.3390/laws10040078.

Chen, B. et al. (2014) ‘Identifying method for opinion leaders in social network based on competency model’, Tongxin Xuebao/Journal on Communications, 35(11), pp. 12–22. Available at: https://doi.org/10.3969/J.ISSN.1000-436X.2014.11.002.

Chen, J., Taylor, J.E. and Wei, H.-H. (2012) ‘Modeling building occupant network energy consumption decision-making: The interplay between network structure and conservation’, Energy and Buildings, 47, pp. 515–524. Available at: https://doi.org/10.1016/j.enbuild.2011.12.026.

Chen, X. et al. (2015) ‘Why Students Share Misinformation on Social Media: Motivation, Gender, and Study-level Differences’, The Journal of Academic Librarianship, 41(5), pp. 583–592. Available at: https://doi.org/10.1016/J.ACALIB.2015.07.003.

Chen, Z., Bruna, J. and Li, L. (2019) ‘Supervised community detection with line graph neural networks’, 7th International Conference on Learning Representations, ICLR 2019, pp. 1–24.

Cheung, V. and Cannons, K. (2002) ‘An Introduction to Neural Networks’.

Chica, M. et al. (2017) ‘A Networked N-player Trust Game and its Evolutionary Dynamics’, IEEE Transactions on Evolutionary Computation, PP, pp. 1–1. Available at: https://doi.org/10.1109/TEVC.2017.2769081.

Cinelli, M. et al. (2021) ‘The echo chamber effect on social media’, Proceedings of the National Academy of Sciences, 118(9), p. e2023301118. Available at: https://doi.org/10.1073/pnas.2023301118.

CNET (1998) TheGlobe.com’s IPO one for the books - CNET News. Available at: https://archive.is/20130119215323/http://news.com.com/2100-1023-217913.html (Accessed: 17 September 2021).

Cook, D.J. (2009) ‘Multi-agent smart environments’, Journal of Ambient Intelligence and Smart Environments, 1(1), pp. 51–55. Available at: https://doi.org/10.3233/AIS-2009-0007.

Cooper, M. (2022) Can AI help fight fake news & misinformation? Here’s what we know, Agility PR Solutions. Available at: https://www.agilitypr.com/pr-news/public-relations/can-ai-help-fight-fake-news-misinformation-heres-what-we-know/ (Accessed: 28 October 2022).

Cowley, B. et al. (2008) ‘Toward an understanding of flow in video games’, Computers in Entertainment, 6(2), pp. 1–27. Available at: https://doi.org/10.1145/1371216.1371223.

Crane, R. and Sornette, D. (2008) ‘Robust dynamic classes revealed by measuring the response function of a social system’, 2008.

Crooks, A. and Heppenstall, A.J. (2012) ‘Introduction to Agent-Based Modelling’, (June 2014). Available at: https://doi.org/10.1007/978-90-481-8927-4.

Dahmann, J.S., Kuhl, F. and Weatherly, R. (2016) ‘Standards for Simulation: As Simple As Possible But Not Simpler The High Level Architecture For Simulation’:, http://dx.doi.org/10.1177/003754979807100603, 71(6), pp. 378–387. Available at: https://doi.org/10.1177/003754979807100603.

DALEY, D.J. and KENDALL, D.G. (1964) ‘Epidemics and Rumours’, Nature 1964 204:4963, 204(4963), pp. 1118–1118. Available at: https://doi.org/10.1038/2041118a0.

Davidavičienė, V. (2018) ‘Research Methodology: An Introduction’, pp. 1–23. Available at: https://doi.org/10.1007/978-3-319-74173-4_1.

De, M. et al. (2010) ‘How Does the Data Sampling Strategy Impact the Discovery of Information Diffusion in Social Media?’ Available at: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.673.5607 (Accessed: 30 September 2021).

Defferrard, M., Bresson, X. and Vandergheynst, P. (2017) ‘Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering’, arXiv:1606.09375 [cs, stat] [Preprint]. Available at: http://arxiv.org/abs/1606.09375 (Accessed: 21 January 2022).

Del Vicario, M. et al. (2017) ‘Modeling confirmation bias and polarization’, Scientific Reports, 7(December 2016), pp. 1–9. Available at: https://doi.org/10.1038/srep40391.

DengLi and YuDong (2014) ‘Deep Learning’, Foundations and Trends in Signal Processing, 7(3–4), pp. 197–387. Available at: https://doi.org/10.1561/2000000039.

Denning, P.J. (2007) ‘Computing is a natural science’, Communications of the ACM, 50(7), pp. 13–18. Available at: https://doi.org/10.1145/1272516.1272529.

Douglas, K.M. (2021) ‘COVID-19 conspiracy theories’, Group Processes & Intergroup Relations, 24(2), pp. 270–275. Available at: https://doi.org/10.1177/1368430220982068.

Duernecker, G. and Vega-Redondo, F. (2017) ‘Social networks and the process of globalization’.

Duong, C.T. and Hoang, T.D. (2019) ‘On Node Features for Graph Neural Networks’, p. 6.

Easley, D. and Kleinberg, J. (2010) Networks, Crowds, and Markets: A Book by David Easley and Jon Kleinberg. Available at: https://www.cs.cornell.edu/home/kleinber/networks-book/ (Accessed: 22 May 2022).

Emirbayer, M. and Goodwin, J. (1994) ‘Network Analysis, Culture, and the Problem of Agency’, American Journal of Sociology, 99(6), pp. 1411–1454.

Enders, A.M. et al. (2021) ‘The Relationship Between Social Media Use and Beliefs in Conspiracy Theories and Misinformation’, Political Behavior [Preprint]. Available at: https://doi.org/10.1007/s11109-021-09734-6.

Erdlenbruch, K. and Bonté, B. (2018) ‘Simulating the dynamics of individual adaptation to floods’, Environmental Science & Policy, 84, pp. 134–148. Available at: https://doi.org/10.1016/j.envsci.2018.03.005.

Eubank, S. et al. (2004) ‘Modelling disease outbreaks in realistic urban social networks’, Nature, 429(6988), pp. 180–184. Available at: https://doi.org/10.1038/nature02541.

Fan, W. et al. (2019) ‘Graph neural networks for social recommendation’, The Web Conference 2019 - Proceedings of the World Wide Web Conference, WWW 2019, pp. 417–426. Available at: https://doi.org/10.1145/3308558.3313488.

Faysal, M.A.M. and Arifuzzaman, S. (2018) ‘A comparative analysis of large-scale network visualization tools’, in *2018 IEEE International Conference on Big Data (Big Data)*. IEEE, pp. 4837–4843.

Fernandes, M. (2020) ‘Confirmation Bias in Social Networks’, SSRN Electronic Journal, pp. 1–44. Available at: https://doi.org/10.2139/ssrn.3504342.

Fogli, A. and Veldkamp, L. (2013) ‘Germs, Social Networks and Growth’.

Fourt, L.A. and Woodlock, J.W. (1960) ‘Early Prediction of Market Success for New Grocery Products’, Journal of Marketing, 25(2), p. 31. Available at: https://doi.org/10.2307/1248608.

Floridi, L., 2005. Is semantic information meaningful data?. *Philosophy and phenomenological research*, *70*(2), pp.351-370.

Fu, G. et al. (2019) ‘Analysis of competitive information diffusion in a group-based population over social networks’, Physica A: Statistical Mechanics and its Applications, 525, pp. 409–419. Available at: https://doi.org/10.1016/J.PHYSA.2019.03.035.

G, A. and M, K. (2001) ‘Social games in a social network’, Physical review. E, Statistical, nonlinear, and soft matter physics, 63(3 Pt 1). Available at: https://doi.org/10.1103/PHYSREVE.63.030901.

Gao, H. and Ji, S. (2019) ‘Graph U-nets’, 36th International Conference on Machine Learning, ICML 2019, 2019-June, pp. 3651–3660. Available at: https://doi.org/10.1109/TPAMI.2021.3081010.

Gent, E. (no date) Sock puppet accounts unmasked by the way they write and post, New Scientist. Available at: https://www.newscientist.com/article/2127107-sock-puppet-accounts-unmasked-by-the-way-they-write-and-post/ (Accessed: 21 October 2021).

Gilbert, G.N. (2019) ‘Agent-based models’, p. 112.

Gilbert, N. and Terna, P. (2000) ‘How to build and use agent-based models in social science’, Mind & Society 2000 1:1, 1(1), pp. 57–72. Available at: https://doi.org/10.1007/BF02512229.

Gilmer, J. et al. (2017) ‘Neural message passing for quantum chemistry’, 34th International Conference on Machine Learning, ICML 2017, 3, pp. 2053–2070.

Goldenberg, J., Libai, B. and Muller, E. (2001) ‘Talk of the Network: A Complex Systems Look at the Underlying Process of Word-of-Mouth’, Marketing Letters, 12(3), pp. 211–223. Available at: https://doi.org/10.1023/A:1011122126881.

Granovetter, M. (1978) ‘Threshold Models of Collective Behavior’, American Journal of Sociology, 83(6), pp. 1420–1443. Available at: https://doi.org/10.1086/226707.

Granovetter, M. (1983) ‘The Strength of Weak Ties: A Network Theory Revisited’, Sociological Theory, 1, p. 201. Available at: https://doi.org/10.2307/202051.

Granovetter, M.S. (1973) ‘The Strength of Weak Ties’, The American Journal of Sociology, 78(6), pp. 1360–1380.

Griffiths, F. et al. (2015) ‘The Impact of Online Social Networks on Health and Health Systems: A Scoping Review and Case Studies’, Policy & Internet, 7(4), pp. 473–496. Available at: https://doi.org/10.1002/poi3.97.

Griliches, Z. (1957) ‘Hybrid Corn: An Exploration in the Economics of Technological Change’, Econometrica, 25(4), p. 501. Available at: https://doi.org/10.2307/1905380.

Gu, W. et al. (2021) ‘Principled approach to the selection of the embedding dimension of networks’, Nature Communications, 12(1), p. 3772. Available at: https://doi.org/10.1038/s41467-021-23795-5.

GuilleAdrien et al. (2013) ‘Information diffusion in online social networks’, ACM SIGMOD Record, 42(2), pp. 17–28. Available at: https://doi.org/10.1145/2503792.2503797.

Gupta, A. et al. (2014) ‘TweetCred: Real-Time Credibility Assessment of Content on Twitter’, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 8851, pp. 228–243. Available at: https://doi.org/10.1007/978-3-319-13734-6_16.

Haenlein, M. and Libai, B. (2013) ‘Targeting Revenue Leaders for a New Product’, Journal of Marketing, 77(3), pp. 65–80. Available at: https://doi.org/10.1509/jm.11.0428.

Hamilton, W.L., Ying, R. and Leskovec, J. (2017) ‘Representation Learning on Graphs: Methods and Applications’, pp. 1–24.

Hamilton, W.L., Ying, R. and Leskovec, J. (2018) ‘Inductive Representation Learning on Large Graphs’, arXiv:1706.02216 [cs, stat] [Preprint]. Available at: http://arxiv.org/abs/1706.02216 (Accessed: 11 December 2021).

Harris, C.R. et al. (2020) ‘Array programming with NumPy’, Nature, 585(7825), pp. 357–362. Available at: https://doi.org/10.1038/s41586-020-2649-2.

Haykin, S. et al. (2009) ‘Neural Networks and Learning Machines Third Edition’.

Hearst, M.A. and Hirsh, H. (2000) ‘AI’s Greatest Trends and Controversies’, IEEE Intelligent Systems [Preprint].

Hegselmann, R. and Flache, A. (1998) ‘Understanding Complex Social Dynamics: A Plea For Cellular Automata Based Modelling’, undefined [Preprint].

Helbing, D. and Balietti, S. (2015) ‘How to Do Agent-Based Simulations in the Future: From Modelling Social Mechanisms to Emergent Phenomena and Interactive Systems Design Why Develop and Use Agent-Based Models ?’, pp. 1–55.

Henry, D., Stattner, E. and Collard, M. (2017) ‘Social media, diffusion under influence of parameters : survey and perspectives’, Procedia Computer Science, 109, pp. 376–383. Available at: https://doi.org/10.1016/j.procs.2017.05.404.

Hertz, J. et al. (1991) Introduction To The Theory Of Neural Computation, Physics Today - PHYS TODAY. Available at: https://doi.org/10.1063/1.2810360.

Holland, J.H. (1996) ‘Hidden Order: How Adaptation Builds Complexity’, Foreign Affairs, 75(4), p. 137. Available at: https://doi.org/10.2307/20047667.

How Web 3.0 will make social media safer (and better) – No Borders Labs (2018). Available at: https://www.noborderslabs.com/how-web-3-0-will-make-social-media-safer-and-better/ (Accessed: 27 September 2021).

Ichikawa, J. and Steup, M., 2014. The analysis of knowledge.

Installation — Matplotlib 3.4.3 documentation (no date). Available at: https://matplotlib.org/stable/users/installing.html (Accessed: 18 October 2021).

Ireton, C., Posetti, Julie, UNESCO (2018) Journalism, ‘fake news’ et disinformation: handbook for journalism education and training. Available at: http://unesdoc.unesco.org/images/0026/002655/265552E.pdf (Accessed: 21 October 2021).

Isaías, P., Miranda, P. and Pífano, S. (2009) ‘Critical Success Factors for Web 2.0 – A Reference Framework’, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 5621 LNCS, pp. 354–363. Available at: https://doi.org/10.1007/978-3-642-02774-1_39.

Jackson, M.O. and Yariv, L. (2006) ‘Diffusion on Social Networks’, p. 17.

Jennings, N.R. (2000) ‘On agent-based software engineering’, Artificial Intelligence, 117(2), pp. 277–296. Available at: https://doi.org/10.1016/S0004-3702(99)00107-1.

Jimenez-Martınez, A. (2015) ‘A Model of Belief Influence in Large Social Networks’, p. 42.

Jin, F. et al. (2013) ‘Epidemiological Modeling of News and Rumors on Twitter’. Available at: http://www.foxnews.com/world/2012/09/08/tweets-false- (Accessed: 3 October 2021).

Jones, M. (2015) History of Social Media: The Invention of Online Networking. Available at: https://historycooperative.org/the-history-of-social-media/ (Accessed: 20 September 2021).

Kapoor, K.K. et al. (2018) ‘Advances in Social Media Research: Past, Present and Future’, Information Systems Frontiers, 20(3), pp. 531–558. Available at: https://doi.org/10.1007/s10796-017-9810-y.

Karinthy, F. (1929) Chain Links.

Kempe, D. and Kleinberg, J. (2003) ‘P137-Kempe’, Kdd, pp. 137–146.

Kempe, D., Kleinberg, J. and Tardos, É. (2003) ‘Maximizing the Spread of Influence through a Social Network’, Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 137–146. Available at: https://doi.org/10.1145/956750.956769.

Kingma, D.P. and Ba, J. (2017) ‘Adam: A Method for Stochastic Optimization’, arXiv:1412.6980 [cs] [Preprint]. Available at: http://arxiv.org/abs/1412.6980 (Accessed: 21 January 2022).

Kipf, T.N. and Welling, M. (2017a) ‘Semi-supervised classification with graph convolutional networks’, 5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings, pp. 1–14.

Kipf, T.N. and Welling, M. (2017b) ‘Semi-supervised classification with graph convolutional networks’, 5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings, pp. 1–14.

Koley, P. et al. (2021) ‘Demarcating Endogenous and Exogenous Opinion Dynamics: An Experimental Design Approach’, ACM Transactions on Knowledge Discovery from Data, 15(6), pp. 1–25. Available at: https://doi.org/10.1145/3449361.

Kreindler, G.E. and Young, H.P. (2014) ‘Rapid innovation diffusion in social networks’, Proceedings of the National Academy of Sciences, 111(supplement_3), pp. 10881–10888. Available at: https://doi.org/10.1073/pnas.1400842111.

Kumar, P. and Sinha, A. (2021) ‘Information diffusion modeling and analysis for socially interacting networks’, Social Network Analysis and Mining, 11(1), p. 11. Available at: https://doi.org/10.1007/s13278-020-00719-7.

Kyza, E.A. et al. (2020) ‘Combating misinformation online: re-imagining social media for policymaking’, Internet Policy Review, 9(4). Available at: https://doi.org/10.14763/2020.4.1514.

Leone, S. (2014) ‘Social Semantic Web for Lifelong Learners – SSW4LL: a learning format | Request PDF’. Available at: https://www.researchgate.net/publication/257815994_Social_Semantic_Web_for_Lifelong_Learners_-_SSW4LL_a_learning_format (Accessed: 18 August 2021).

Lewis, C.W. and Monett, D. (2018) ‘Text analysis of unstructured data on definitions of intelligence’, in Proceedings of The 2018 Meeting of the International Association for Computing and Philosophy, IACAP.

Li, M. et al. (2017) ‘A survey on information diffusion in online social networks: Models and methods’, Information (Switzerland), 8(4). Available at: https://doi.org/10.3390/info8040118.

Li, Y. et al. (2016) ‘Gated graph sequence neural networks’, 4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings, (1), pp. 1–20.

Li, Y., Zhao, H.V. and Chen, Y. (2019) ‘An epidemic model for correlated information diffusion in crowd intelligence networks’, International Journal of Crowd Science, 3(2), pp. 2398–7294. Available at: https://doi.org/10.1108/IJCS-01-2019-0005.

Liu, C. and Zhang, Z.K. (2014) ‘Information spreading on dynamic social networks’, Communications in Nonlinear Science and Numerical Simulation, 19(4), pp. 896–904. Available at: https://doi.org/10.1016/j.cnsns.2013.08.028.

Liu, Y. and Wu, Y.B. (2018) ‘Early Detection of Fake News on Social Media Through Propagation Path Classification with Recurrent and Convolutional Networks’, Thirty-Second AAAI Conference on Artificial Intelligence, pp. 354–361.

Logan, B. (2013) ‘G54DIA: Designing Intelligent Agents Lecture 2: Task Environments & Architectures"’.

Lomax, R.G. and Hahs-Vaughn, D.L. (2013) An Introduction to Statistical Concepts: Third Edition. Taylor & Francis. Available at: https://books.google.co.uk/books?id=t9gC0KzImSIC.

Luo, H., Cai, M. and Cui, Y. (2021) ‘Spread of Misinformation in Social Networks: Analysis Based on Weibo Tweets’, Security and Communication Networks. Edited by C. Gan, 2021, p. 7999760. Available at: https://doi.org/10.1155/2021/7999760.

Luu, M.D. et al. (2011) ‘A survey of information diffusion models and relevant problems A Survey of Information Diffusion Models and Relevant Problems’.

Ma, J., Gao, W. and Wong, K.-F. (2017) ‘Detect Rumours in Microblog Posts Using Propagation Structure via Kernel Learning’, pp. 2016–2023.

Macal, C.M. and North, M.J. (2009) ‘Agent-based modeling and simulation’, Proceedings - Winter Simulation Conference, pp. 86–98. Available at: https://doi.org/10.1109/WSC.2009.5429318.

Macal, C.M. and North, M.J. (2010a) ‘Tutorial on agent-based modelling and simulation’, pp. 151–162. Available at: https://doi.org/10.1057/jos.2010.3.

Macal, C.M. and North, M.J. (2010b) ‘Tutorial on agent-based modelling and simulation’, pp. 151–162. Available at: https://doi.org/10.1057/jos.2010.3.

Macedonia’s Pro-Trump Fake News Industry Had American Links, And Is Under Investigation For Possible Russia Ties (no date) BuzzFeed News. Available at: https://www.buzzfeednews.com/article/craigsilverman/american-conservatives-fake-news-macedonia-paris-wade-libert (Accessed: 21 October 2021).

Macy, M.W. and Willer, R. (2002) ‘From Factors to Actors: Computational Sociology and Agent-Based Modelling’, Annual Review of Sociology, 28, pp. 143–166.

Mahajan, V. and Muller, E. (1979) ‘Innovation Diffusion and New Product Growth Models in Marketing’, Journal of Marketing, 43(4), pp. 55–68. Available at: https://doi.org/10.1177/002224297904300407.

Mansfield, E. (1961) ‘Technical Change and the Rate of Imitation’, Econometrica, 29(4), p. 741. Available at: https://doi.org/10.2307/1911817.

Marsden, P.V. and Friedkin, N.E. (1993) ‘Network Studies of Social Influence’, Sociological Methods & Research, 22(1), pp. 127–151.

Marsella, S.C., Pynadath, D.V. and Read, S.J. (2004) ‘PsychSim: Agent-based modelling of social interactions and influence’, Proceedings of the international conference on cognitive modelling, pp. 243–248.

Mcauley, J. and Leskovec, J. (2014) ‘Discovering social circles in ego networks’, ACM Transactions on Knowledge Discovery from Data, 8(1), pp. 1–28. Available at: https://doi.org/10.1145/2556612.

McKinney, W. (no date) ‘pandas: a Foundational Python Library for Data Analysis and Statistics’, p. 9.

Mér\Ho, L. (1998) ‘John von Neumann’s Game Theory’, in Moral Calculations: Game Theory, Logic, and Human Frailty. New York, NY: Springer New York, pp. 83–102. Available at: https://doi.org/10.1007/978-1-4612-1654-4_6.

Meserole, C. (2018) How misinformation spreads on social media—And what to do about it. Available at: https://www.brookings.edu/blog/order-from-chaos/2018/05/09/how-misinformation-spreads-on-social-media-and-what-to-do-about-it/ (Accessed: 7 October 2021).

Micheli, A. (2009) ‘Neural Network for Graphs: A Contextual Constructive Approach’, IEEE Transactions on Neural Networks, 20(3), pp. 498–511. Available at: https://doi.org/10.1109/TNN.2008.2010350.

Miller, J.H. and Page, S.E. (2019) ‘Complex Adaptive Systems: Views from the Physical, Natural, and Social Sciences’, Princeton University Press, (March), pp. 431–436. Available at: https://doi.org/10.1007/978-3-030-20309-2.

Milli, L. et al. (2018) ‘Active and passive diffusion processes in complex networks’, Applied Network Science, 3(1). Available at: https://doi.org/10.1007/s41109-018-0100-5.

Mitchell, T.M. (1997) Machine Learning.

Mitra, T., Wright, G.P. and Gilbert, E. (2017) ‘A Parsimonious Language Model of Social Media Credibility Across Disparate Events’.

Modgil, S., Singh, R.K., Gupta, S. and Dennehy, D., 2021. A confirmation bias view on social media induced polarisation during Covid-19. *Information Systems Frontiers*, pp.1-25.

Moglia, M., Podkalicka, A. and McGregor, J. (2018) ‘An Agent-Based Model of Residential Energy Efficiency Adoption’, Journal of Artificial Societies and Social Simulation, 21(3), pp. 1–3.

Monti, F. et al. (2019) ‘Fake news detection on social media using geometric deep learning’, arXiv, pp. 1–15.

Moussaïd, M. et al. (2009) ‘Experimental study of the behavioural mechanisms underlying self-organization in human crowds’.

Murase, Y. et al. (2021) ‘Deep Learning Exploration of Agent-Based Social Network Model Parameters’, Frontiers in Big Data, 4. Available at: https://doi.org/10.3389/fdata.2021.739081.

Murayama, T. et al. (2021) ‘Modeling the spread of fake news on Twitter’, PLOS ONE, 16(4), p. e0250419. Available at: https://doi.org/10.1371/journal.pone.0250419.

Myers, S.A. and Leskovec, J. (2012) ‘Clash of the Contagions: Cooperation and Competition in Information Diffusion’, 2012 IEEE 12th International Conference on Data Mining [Preprint]. Available at: https://www.academia.edu/2957241/Clash_of_the_contagions_Cooperation_and_competition_in_information_diffusion (Accessed: 30 September 2021).

Namatame, A. and Chen, S.-H. (2016) ‘Agent-Based Modeling and Network Dynamics’, Agent-Based Modeling and Network Dynamics [Preprint]. Available at: https://doi.org/10.1093/ACPROF:OSO/9780198708285.001.0001.

Negahban, A. and Smith, J.S. (2018) ‘A joint analysis of production and seeding strategies for new products: an agent-based simulation approach’, Annals of Operations Research, 268(1), pp. 41–62. Available at: https://doi.org/10.1007/s10479-016-2389-8.

Newman, M. (2010) ‘Networks: An Introduction’, Networks: an Introduction [Preprint]. Available at: https://doi.org/10.1093/acprof:oso/9780199206650.001.0001.

Nicosia, V. et al. (2012) ‘Components in time-varying graphs’, Chaos: An Interdisciplinary Journal of Nonlinear Science, 22(2), p. 023101. Available at: https://doi.org/10.1063/1.3697996.

Nikolic, I. and Ghorbani, A. (2011) ‘A method for developing agent-based models of socio-technical systems’, in. 2011 International Conference on Networking, Sensing and Control, ICNSC 2011, pp. 44–49. Available at: https://doi.org/10.1109/ICNSC.2011.5874914.

North, M.J., Collier, N.T. and Vos, J.R. (2006) ‘Experiences Creating Three Implementations of the Repast Agent Modeling Toolkit’.

Nowak, S.A., Matthews, L.J. and Parker, A.M. (2017) A General Agent-Based Model of Social Learning.

Odell, J. et al. (2003) ‘The Communication Environment’, Journal Of Object Technology Agents and their Environment, 2(3), pp. 39–52.

Pancs, R. and Vriend, N.J. (2003) ‘Schelling’s Spatial Proximity Model of Segregation Revisited’, SSRN Electronic Journal [Preprint]. Available at: https://doi.org/10.2139/SSRN.375080.

Pastor-Satorras, R. and Vespignani, A. (2001) ‘Epidemic Spreading in Scale-Free Networks’. Available at: https://doi.org/10.1103/PhysRevLett.86.3200.

Pei, S. et al. (2014) ‘Searching for superspreaders of information in real-world social media’, Scientific Reports, 4(1), p. 5547. Available at: https://doi.org/10.1038/srep05547.

Pei, S. et al. (2020) ‘Influencer identification in dynamical complex systems’, Journal of Complex Networks. Edited by J. Gomez-Gardenes, 8(2), p. cnz029. Available at: https://doi.org/10.1093/comnet/cnz029.

Pereira, F.S.F., de Amo, S. and Gama, J. (2016) 'Evolving Centralities in Temporal Graphs: A Twitter Network Analysis', (August), pp. 43–48. Available at: https://doi.org/10.1109/mdm.2016.88.

Popat, K. (2017) 'Assessing the credibility of claims on the web', 26th International World Wide Web Conference 2017, WWW 2017 Companion, pp. 735–739. Available at: https://doi.org/10.1145/3041021.3053379.

Popper, K.R., 1979. *Objective knowledge: An evolutionary approach* (Vol. 49). Oxford: Clarendon press.

Prakash, B.A. et al. (2012) 'Winner takes all: Competing viruses or ideas on fair-play networks', WWW'12 - Proceedings of the 21st Annual Conference on World Wide Web, pp. 1037–1046. Available at: https://doi.org/10.1145/2187836.2187975.

Qazvinian, V. et al. (2011) 'Rumor has it: Identifying Misinformation in Microblogs', pp. 1589–1599. Available at: https://doi.org/10.5555/2145432.2145602.

Qian, F. et al. (2013) 'Neural User Response Generator : Fake News Detection with Collective User Intelligence', pp. 3834–3840.

Qian, F. et al. (2018) 'Neural User Response Generator: Fake News Detection with Collective User Intelligence'. Available at: http://www.theguardian.com/world/2017/jan/09 (Accessed: 3 October 2021).

Rahmandad, H. and Sterman, J.D. (2008) 'Heterogeneity and Network Structure in the Dynamics of Diffusion: Comparing Agent-Based and Differential Equation Models', Manag. Sci., 54, pp. 998–1014.

Railsback, S.F. and Grimm, V. (2011) Agent-Based and Individual-Based Modeling: A Practical Introduction. Princeton University Press. Available at: http://www.jstor.org/stable/j.ctt7sns7.

Rainie, L., Anderson, J. and Albright, J. (2017) 'The Future of Free Speech, Trolls, Anonymity and Fake News Online', Pew Research Center: Internet, Science & Tech, 29 March. Available at: https://www.pewresearch.org/internet/2017/03/29/the-future-of-free-speech-trolls-anonymity-and-fake-news-online/ (Accessed: 10 October 2022).

Ramezani, M. et al. (2019) ‘News Labeling as Early as Possible: Real or Fake?’, Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, pp. 536–537. Available at: https://doi.org/10.1145/3341161.3342957.

Reisdorf, S. et al. (2014) ‘RAPID APPLICATION DEVELOPMENT USING WEB-2.0* TECHNOLOGIES’.

Riva, G., Wiederhold, B. and Cipresso, P. (2016) The Psychology of Social Networking Vol.1. Personal Experience in Online Communities. Available at: https://doi.org/10.1515/9783110473780.

Rogers, E.M. (1995) ‘17 - Rogers 1995 cap 6 plan 5’.

Rogers, E.M., Hagerstrand, T. and Pred, A. (1969) ‘Innovation Diffusion as a Spatial Process’, Technology and Culture, 10(3), p. 480. Available at: https://doi.org/10.2307/3101713.

Rosenblatt, F. (1958) ‘The perceptron: A probabilistic model for information storage and organization in the brain’, Psychological Review, 65(6), pp. 386–408. Available at: https://doi.org/10.1037/H0042519.

Rui, X. et al. (2018) ‘SPIR: The potential spreaders involved SIR model for information diffusion in social networks’, Physica A: Statistical Mechanics and its Applications, 506, pp. 254–269. Available at: https://doi.org/10.1016/J.PHYSA.2018.04.062.

Russel, S. and Norvig, P. (2012) ‘Artificial intelligence—a modern approach 3rd Edition’, The Knowledge Engineering Review [Preprint]. Available at: https://books.google.com/books/about/Artificial_Intelligence.html?id=8jZBksh-bUMC (Accessed: 22 August 2021).

Sasahara, K. et al. (2021) ‘Social influence and unfollowing accelerate the emergence of echo chambers’, Journal of Computational Social Science 2020 4:1, 4(1), pp. 381–402. Available at: https://doi.org/10.1007/S42001-020-00084-7.

Scarselli, F. et al. (2009) ‘The graph neural network model’, IEEE Transactions on Neural Networks, 20(1), pp. 61–80. Available at: https://doi.org/10.1109/TNN.2008.2005605.

Scellato, S. (no date) ‘NetworkX: Network Analysis with Python’, p. 51.

Scott, J. (2010) ‘Social network analysis: developments, advances, and prospects’, Social Network Analysis and Mining 2010 1:1, 1(1), pp. 21–26. Available at: https://doi.org/10.1007/S13278-010-0012-6.

Seo, D., Raman, R.K. and Varshney, L.R. (2020) ‘Social Learning with Beliefs in a Parallel Network’, IEEE International Symposium on Information Theory - Proceedings, 2020-June, pp. 1265–1270. Available at: https://doi.org/10.1109/ISIT44484.2020.9174359.

Shannon, C.E., 1949. Communication theory of secrecy systems. *The Bell system technical journal*, *28*(4), pp.656-715.

Sharma, V. (2020) ‘Role of Artificial Intelligence in Social Media’, Quytech Blog, 10 July. Available at: https://www.quytech.com/blog/role-of-artificial-intelligence-in-social-media/ (Accessed: 21 April 2022).

Shi, G. et al. (2015) ‘The Evolution of Beliefs over Signed Social Networks’, arXiv:1307.0539 [physics] [Preprint]. Available at: http://arxiv.org/abs/1307.0539 (Accessed: 21 January 2022).

Simpson, J.A. and Weiner, E.S. (1989) The Oxford English dictionary. Oxford: Clarendon Press.

Skansi, S. (2018) ‘Introduction to Deep Learning’. Available at: https://doi.org/10.1007/978-3-319-73004-2.

Social Network Visualizer: SocNetV Manual (no date). Available at: https://socnetv.org/docs/index.html#toolbar (Accessed: 24 October 2021).

Song, Y. and van der Schaar, M. (2013) ‘Dynamic Network Formation with Incomplete Information’, Economic Theory, 59(2), pp. 301–331.

SourceForge.net: Project Info - NetworkX (no date). Available at: https://web.archive.org/web/20050429205335/http://sourceforge.net:80/projects/networkx/ (Accessed: 18 October 2021).

Squazzoni, F. (2010) ‘THE IMPACT OF AGENT-BASED MODELS IN THE SOCIAL SCIENCES AFTER 15 YEARS OF INCURSIONS’, History of Economic Ideas, 18(2), pp. 197–233.

Squazzoni, F., Jager, W. and Edmonds, B. (2014) ‘Social Simulation in the Social Sciences: A Brief Overview’, Social Science Computer Review, 32(3), pp. 279–294. Available at: https://doi.org/10.1177/0894439313512975.

Stokman, F.N. (2001) ‘Networks: Social’, in N.J. Smelser and P.B. Baltes (eds) International Encyclopedia of the Social & Behavioral Sciences. Oxford: Pergamon, pp. 10509–10514. Available at: https://doi.org/10.1016/B0-08-043076-7/01934-3.

Strang, D. and Soule, S.A. (1998) ‘DIFFUSION IN ORGANIZATIONS AND SOCIAL MOVEMENTS: From Hybrid Corn to Poison Pills’, Business Week. Group Manager, 24, p. 180.

Streamed Graph Datasets (no date). Available at: https://eecs.wsu.edu/~yyao/StreamingGraphs.html (Accessed: 25 October 2021).

Strusani, D. and Houngbonon, G.V. (2019) ‘The Role of Artificial Intelligence in Supporting Development in Emerging Markets’, The Role of Artificial Intelligence in Supporting Development in Emerging Markets [Preprint]. Available at: https://doi.org/10.1596/32365.

Suk, H.I. (2017) ‘An Introduction to Neural Networks and Deep Learning’, Deep Learning for Medical Image Analysis, pp. 3–24. Available at: https://doi.org/10.1016/B978-0-12-810408-8.00002-X.

Sun, L., Zhou, Y. and Guan, X. (2016) ‘Modelling multi-topic information propagation in online social networks based on resource competition’:, http://dx.doi.org/10.1177/0165551516642928, 43(3), pp. 342–355. Available at: https://doi.org/10.1177/0165551516642928.

Tan, Q., Liu, N. and Hu, X. (2019) ‘Deep Representation Learning for Social Network Analysis’, Frontiers in Big Data, 2. Available at: https://doi.org/10.3389/fdata.2019.00002.

The laws of imitation : Tarde, Gabriel de, 1843-1904 : Free Download, Borrow, and Streaming : Internet Archive (1903). Available at: https://archive.org/details/lawsofimitation00tard?ref=ol&view=theater (Accessed: 9 September 2021).

Thota, A. (2018) ‘Fake News Detection: A Deep Learning Approach’, 1(3).

Toivonen, R. et al. (2006) ‘A model for social networks’, Physica A: Statistical Mechanics and its Applications, 371(2), pp. 851–860. Available at: https://doi.org/10.1016/j.physa.2006.03.050.

Usó-Doménech, J.L. and Nescolarde-Selva, J. (2015) ‘What are Belief Systems?’, Foundations of Science 2015 21:1, 21(1), pp. 147–152. Available at: https://doi.org/10.1007/S10699-015-9409-Z.

Van Lier (2019) The Evolution of Social Networking | by Jessica Van Lier | Stan World | Medium. Available at: https://medium.com/stan-world/the-evolution-of-social-networking-bab2497c0733 (Accessed: 27 September 2021).

Vanin, P., Vanin, P. and A, J. (2002) ‘Network Formation in the Lab: A Pilot Experiment’. Available at: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.515.2011 (Accessed: 28 September 2021).

Walker, J.L. (1969) ‘The Diffusion of Innovations among the American States*’, American Political Science Review, 63(3), pp. 880–899. Available at: https://doi.org/10.2307/1954434.

Wang, M. et al. (2019) ‘Deep Graph Library: A Graph-Centric, Highly-Performant Package for Graph Neural Networks’, pp. 1–18.

Wang, Q. et al. (2015) ‘ESIS: Emotion-based spreader-ignorant-stifler model for information diffusion’, Knowledge-Based Systems, 81, pp. 46–55. Available at: https://doi.org/10.1016/j.knosys.2015.02.006.

Wasserman, S. and Faust, K. (1995) ‘Social Network Data: Collection and Applications’, Social Network Analysis: Methods and Applications, pp. 28–66.

Watts, D.J. and Strogatz, S.H. (1998) ‘Collective dynamics of “small-world” networks’, Nature, 393(6684), pp. 440–442. Available at: https://doi.org/10.1038/30918.

Watts, D.J., Watts and J., D. (2002) ‘A simple model of global cascades on random networks’, PNAS, 99(9), pp. 5766–5771. Available at: https://doi.org/10.1073/PNAS.082090499.

What are Neural Networks? | IBM (2020). Available at: https://www.ibm.com/cloud/learn/neural-networks (Accessed: 9 September 2021).

What is Artificial Neural Network !? | by Ana Jessica | featurepreneur | Aug, 2021 | Medium (2021). Available at: https://medium.com/featurepreneur/what-is-artificial-neural-network-aae02633071d (Accessed: 10 September 2021).

Widrow, B. and Walach, E. (1984) ‘On the statistical efficiency of the LMS algorithm with nonstationary inputs’, IEEE Transactions on Information Theory, 30(2), pp. 211–221. Available at: https://doi.org/10.1109/TIT.1984.1056892.

Wilensky, U. and Wilensky, U. (2002) ‘Modeling nature’s emergent patterns with multi-agent languages’, PROCEEDINGS OF EUROLOGO 2002 [Preprint]. Available at: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.294.8094 (Accessed: 4 October 2021).

Will, M. et al. (2020) ‘Combining social network analysis and agent-based modelling to explore dynamics of human interaction: A review’, Socio-Environmental Systems Modelling, 2, p. 16325. Available at: https://doi.org/10.18174/sesmo.2020a16325.

Wilson, C. et al. (2009) ‘User Interactions in Social Networks and their Implications’.

Wilson, R.J. (2009) Introduction to graph theory. 4. ed., [Nachdr.]. Harlow Munich: Prentice Hall.

Wissler, C. (1915) ‘The Diffusion of Horse Culture Among the North American Indians’, Proceedings of the National Academy of Sciences, 1(4), pp. 254–256. Available at: https://doi.org/10.1073/PNAS.1.4.254.

Wolfram, S. (1983) ‘Statistical mechanics of cellular automata’, Reviews of Modern Physics, 55(3), p. 601. Available at: https://doi.org/10.1103/RevModPhys.55.601.

Wu, F. and Huberman, B.A. (2007) ‘Novelty and collective attention’, 104(45).

Xia, L. et al. (2015) ‘Modeling and Analyzing the Interaction between Network Rumors and Authoritative Information’, Entropy, 17, pp. 471–482. Available at: https://doi.org/10.3390/e17010471.

Yu, L. et al. (2016) ‘Characterizing super-spreading in microblog: An epidemic-based information propagation model’, Physica A, 463, pp. 202–218. Available at: https://doi.org/10.1016/J.PHYSA.2016.07.022.

Yang, C. et al. (2017) 'A Neural Network Approach to Jointly Modeling Social Networks and Mobile Trajectories', ACM Transactions on Information Systems, 35(4), pp. 1–28. Available at: https://doi.org/10.1145/3041658.

Yang, Y., Tang, J. and Leung, C.W. (2015) 'RAIN: Social Role-Aware Information Diffusion', pp. 367–373.

Yilmaz, L. and Ören, T. (2004) 'A Conceptual Model for Reusable Simulations Within a Model-Simulator-Context Framework'.

Ying, R. et al. (2018) 'DIFFpool', Advances in Neural Information Processing Systems, 2018-Decem(NeurIPS), pp. 4800–4810.

Zachary, W.W. (1977) 'An Information Flow Model for Conflict and Fission in Small Groups Author (s): Wayne W . Zachary Published by : The University of Chicago Press Stable URL : http://www.jstor.org/stable/3629752', Journal of Anthropological Research, 33(4), pp. 452–473.

Zeigler, B., Muzy, A. and Yilmaz, L. (2009) 'Artificial Intelligence in Modeling and Simulation', Encyclopedia of Complexity and Systems Science, pp. 344–368. Available at: https://doi.org/10.1007/978-0-387-30440-3_24.

Zhang, A., Bogle, A. and Wallis, D.J. (2021) 'Submission to the UN Special Rapporteur on disinformation and freedom of opinion and expression', p. 10.

Zhang, J. et al. (2018) 'Fake News Detection with Deep Diffusive Network Model'. Available at: http://arxiv.org/abs/1805.08751.

Zhang, L. et al. (2020) 'Dynamic Graph Message Passing Networks'.

Zhang, M. and Chen, Y. (2018) 'Link prediction based on graph neural networks', Advances in Neural Information Processing Systems, 2018-Decem(Nips), pp. 5165–5175.

Zhao, L. et al. (2013) 'Rumor spreading model considering forgetting and remembering mechanisms in inhomogeneous networks', Physica A: Statistical Mechanics and its Applications, 392(4), pp. 987–994. Available at: https://doi.org/10.1016/J.PHYSA.2012.10.031.

Zhao, Y., Li, S. and Jin, F. (2016a) ‘Identification of influential nodes in social networks with community structure based on label propagation’, Neurocomputing, 210, pp. 34–44. Available at: https://doi.org/10.1016/j.neucom.2015.11.125.

Zhao, Y., Li, S. and Jin, F. (2016b) ‘Identification of influential nodes in social networks with community structure based on label propagation’, Neurocomputing, 210, pp. 34–44. Available at: https://doi.org/10.1016/j.neucom.2015.11.125.

Zhu, X. (2008) ‘Semi-Supervised Learning Literature Survey’, p. 60.

Zinoviev, D. (2015) ‘Information Diffusion in Social Networks’, (May). Available at: https://doi.org/10.4018/978-1-61350-444-4.

APPENDIX A

Synthetic Network

▪ Eigenvector Centrality Report

The Eigenvector centrality is shown below for 10 nodes encompassing the key nodes in the network as identified by this metric. The e index of a node n is the ratio of all geodesics between pairs of nodes which run through n (Kalamaras D. *Social Network Visualizer (SocNetV))*.

$Node$	$Node\ Label$	e	e'	$\%e$
1	31	0.2060	1.0000	100.0000
2	61	0.1803	0.8750	87.5000
3	163	0.1546	0.7500	75.0000
4	171	0.1546	0.7500	75.0000
5	18	0.1288	0.6250	62.5000
6	24	0.1288	0.6250	62.5000
7	27	0.1288	0.6250	62.5000
8	29	0.1288	0.6250	62.5000
9	41	0.1288	0.6250	62.5000
10	45	0.1288	0.6250	62.5000

Table 1. Eigenvector Indexes of Network Nodes (Artificial Network)

The e' is the standardized index (e divided by $(N-1)(N-2)/2$) for symmetric networks and $(N-1)(N-2)$ for non-symmetric networks.

▪ In-Degree Centrality Report

The Degree Centrality is shown below for nodes encompassing the key nodes in the network as identified by this metric. The d index of a node n in a directed network is the sum of outbound arc from a node to all adjacent nodes (Kalamaras D. *Social Network Visualizer (SocNetV))*.

$Node$	$Node\ Label$	d	d'	$\%d$
1	$1(s_1)$	110.0000	0.3099	30.9859
2	$2(s_2)$	133.0000	0.3746	37.4648
3	$10(r_{10})$	32.0000	0.0901	9.0141
4	$15(r_{15})$	24.0000	0.0676	6.7606
5	$4(r_4)$	16.0000	0.0450	4.5070
6	$3(r_3)$	14.0000	0.0394	3.9437
7	$13(r_{13})$	11.0000	0.0310	3.0986
8	$7(r_7)$	7.0000	0.0197	1.9718
9	$11(r_{11})$	4.0000	0.0113	1.1268
10	$19(r_{19})$	4.0000	0.0113	1.1268

Table 2. In-degree Indexes of Network Nodes (Artificial Network)

▪ **Information Centrality Report**

The Information centrality index introduced by Stephenson and Zelen (1989) measures the information through all paths between nodes. The values for the 10 nodes encompassing the key nodes in the network as identified by this metric are shown below (Kalamaras D. *Social Network Visualizer (SocNetV))*.

$Node$	$Node\ Label$	i	i'	$\%i$
1	$1(s_1)$	1.6464	0.0094	0.9448
2	$2(s_2)$	1.6545	0.0095	0.9495
3	$10\ (r_{10})$	1.5449	0.0089	0.8866
4	$15(r_{15})$	1.5088	0.0087	0.8659
5	$4(r_4)$	1.4341	0.0082	0.8230
6	$3(r_3)$	1.4262	0.0082	0.8186
7	$13(r_{13})$	1.3772	0.0079	0.7903
8	$7(r_7)$	1.3099	0.0075	0.7517
9	$11(r_{11})$	1.2271	0.0070	0.7043
10	$19(r_{19})$	1.1701	0.0067	0.6715

Table 3**.** Information Indexes of Network Nodes

Full Results

Real Network - Dataset One

▪ **Eigenvector Centrality Report**

The Eigenvector centrality is shown below for the key nodes encompassing the key nodes in the network.

$Node$	$Node\ Label$	e	e'	$\%e$
1	$2565(r_{21265})$	0.8994	1.0000	100
2	$154(r_{888})$	0.1004	0.1116	11.1606
3	$224(r_{1241})$	0.1004	0.1116	11.1606
4	$392(r_{2390})$	0.1004	0.1116	11.1606
5	$471(r_{3060})$	0.1004	0.1116	11.1606
6	$510(r_{3257})$	0.1004	0.1116	11.1606
7	$566(r_{3554})$	0.1004	0.1116	11.1606

Table 4**.** Eigenvector - Real Network 1

▪ **In-Degree Centrality Report**

The Degree Centrality is shown below for the key nodes in the network.

$Node$	$Node\ Label$	d	d'	$\%d$
1	$2249(r_{18143})$	434	0.0343	3.4319
2	$2565(r_{21265})$	332	0.0263	2.6253
3	$5056(r_{60921})$	210	0.0167	1.6601
4	$2609(r_{21939})$	130	0.0103	1.0280
5	$993(r_{6749})$	164	0.0130	1.2970
6	$228(r_{1277})$	111	0.0088	0.8777
7	$3291(r_{30602})$	89	0.0070	0.7037

Table 5**.** In-degree - Real Network 1

APPENDIX B

1. Input Module

▪ **Network Creation:**

In Social networks, a node's importance within the network is often in correlation with its degree. The nodes with larger degrees are likely to be very influential within the network and hence key to the diffusion process. The node with the largest degree will be the hub (super) node. The initiation phase of the model sees the different classes of nodes established with the seed nodes chosen based on their relatively large degree.in real life OSNs, node relationships in the network are often independent but interdependent in the diffusion process and the initiation phase ensures this independence.

The procedure for the network creation and initiation is detailed below:

Step 1: Create an empty set of users () and number of connections connecting the users as a set of users $S = V$.

Step 2: Populate empty node set with several nodes, $S = 16$.

Step 3: Set a subset of the users as seed users (user's 0 and 1) Ω = ∅ from the user set, adding connections from those users to every other user in the network.

Step 4: Set a subset of the users as mirror users (users 3, 6 and 12) adding connections from those users to random users in the network.

Step 5: Check the in-degree of the users to establish the seed users and mirror users in the user set.

Step 6: Set the 2 users with the largest degree part of the seed users set v_{max} as the super users.

Step 7: Output the user set Ω showing the users with the largest degree k.

The python codes are shown below:

```
# 1. Graph Creation:

G = dgl.DGLGraph()
```

```
G.add_nodes(15) # populate the graph and add edges
G.add_edges([2,3,4,5,6,7,8,9,10,11,12,13,14,15], 0) # super user 1
G.add_edges([2,4,5,6,7,8,9,10,11,12,13,14,15], 1) # super user 2
G.add_edges([2,4,5,15], 3) # first mirror agent
G.add_edges([7,8,10,14], 6) # second mirror agent
G.add_edges([8,9,10,11,13,15], 12) # third mirror agent


G = dgl.add_self_loop(G) # allows for users to include their features in aggregate of the features of neighbour users


# Print out the number of nodes and edges in our newly constructed graph:

print ('')

print('We have %d users.' % G.number_of_nodes())
print('We have %d connections.' % G.number_of_edges())
```

2. Labelling Module

In DGL, the nodes and edges of a graph can have several user-defined named features for storing graph-specific properties of the nodes and edges. A feature is created via tensor assignment, which assigns a feature to each node/edge in the graph. The DGL interface - “ndata” is used to access these features. Two profile states are created for the simulation namely: a Positive Profile state (state 2) and a Negative Profile state 1. Users initiated with profile state 1 are defined as having a positive belief profile with unbiased beliefs, hence would share information that is accurate and be able to accurately verify the veracity of information relying on their beliefs alone. The model is set up as a semi-supervised setting, profiles are assigned to the super nodes (0 and 1), mirror nodes (3, 6, 12) as part of the initialization.

The implementation is done using the embedding functionality. All 16 nodes on the network are assigned an embedding which serves as the input for the network. For User representation learning operation, the learnable embeddings serve as the input user features. The python codes are shown below:

```
embed = nn.Embedding(16, 8)  # 16 nodes with embedding dimension equal to 8
G.ndata['feat' 'boy'] = embed.weight
```

```
profile_init = torch.tensor([0, 1, 3, 6, 12])  # users with profiles
profiles = torch.tensor([1, 0, 0, 0, 1])  # their profiles are different
```

3. Network Module

The training loop follows that of other established PyTorch models. It involves the following steps which is carried out using built-in DGL functions (Wang *et al.*, 2019):

▪ Creating an optimizer

▪ Feeding the inputs to the model – the learnable user embeddings created serve as the input into the model.

▪ calculate the loss

▪ Model optimization

```
# 3. Define GCN

class GCN(nn.Module):
    def __init__(self, input_features, hidden_size, hidden_size2,
num_classes):
        super(GCN, self).__init__()
        self.conv1 = GraphConv(input_features, hidden_size)
        self.conv2 = GraphConv(hidden_size, hidden_size2)
        self.conv3 = GraphConv(hidden_size2, hidden_size2)
        self.classify = nn.Linear(hidden_size2, num_classes) # output layer

        print(f"User Classes:  {num_classes}")

    def forward(self, g, inputs):
        x = F.relu(self.conv1(g, inputs))
        x = F.relu(self.conv2(g, x))
        x = F.relu(self.conv3(g,x))
        #print(f"x.size, {x.size()}")
        return self.classify(x)

nnet = GCN(8, 8, 8, 2) # output layer feature of size 2
print (nnet)
```

4. Network Validation Results

▪ F-Test - Synthetic Network

The result of the F-Test for the synthetic network is shown below. The test demonstrates equal variance across two simulation runs.

F-Test Two-Sample for Variances		
	Synthetic	**Zachary's**
Mean	**0.64885556**	**0.313222222**
Variance	**0.01077456**	**0.098402142**
Observations	**9**	**9**
df	**8**	**8**
F	**0.10949521**	
P(F<=f) one-tail	**0.00259654**	
F Critical one-tail	**0.29085822**	

Table 1: F-Test

Second Test - Zachary's Network

The result of the F-Test for the real network is shown below. The test demonstrates equal variance across two simulation runs.

F-Test Two-Sample for Variances		
	0.7787	**0.7765**
Mean	**0.2550375**	**0.2070375**
Variance	**0.077637697**	**0.06269272**
Observations	**8**	**8**
df	**7**	**7**
F	**1.238384571**	
P(F<=f) one-tail	**0.392541601**	
F Critical one-tail	**3.78704354**	

Table 2: F-test Zachary's

APPENDIX C

Synthetic Network

1. Input Module

The input module process can be described in four steps:

Step 1 - Import the library packages required for by the input module as seen in Fig. 44.

Step 2 - Load the dataset - graph data from disk and define the source and destination for the edges.

Step 3 - Transform graph into DGL then into NetworkX for display.

Step 4 - Query graph structure to determine the number of nodes, number of edges, the indegree and the outdegree on the nodes.

Dataset Description

The dataset used as the synthetic network in the model is created as two csv (comma separated values) files - the first (nodes.csv) containing the nodes and its features, the second (edge.csv) for the edges and their features. Both files are loaded into memory in the model with the source point and destination points for edges defined. The details of the synthetic data used in the dataset is detailed below:

- The node.csv file contains all the nodes in the graph as well as their attributes. The file has four columns:

  1). ID column - A column identified nodes by their ID.

  2). Class Column - A column identifying the class that the nodes belong to. This also serves as their labels. It has four different types of values.

  3). Belief Column - A column identifying the nodes by their belief types. A belief profile of "1" indicates a positive belief, "0" indicates a negative belief and "2" indicates a neutral belief. It serves as an integer feature for each node in the network.

  4). Bias Column - A column identifying the bias states of the nodes in the graph. Nodes can be in either one of two states - biased (0) or unbiased (1). It serves the second integer feature for the node in the network.

- The edge.csv file defines all edges between the nodes in the graph showing interactions. The file also has three columns:

  1). Source column - originating node of the connection.

2). Destination column - the destination node and endpoint for the edge.

3). Weight column - the weights on the different edges and serves as the edge feature.

2. Labelling Module

Nodes in the graph are identified by consecutive integers starting from zero and edges can be specified by two endpoints (u, v) or the integer id assigned when the edges are added. Edge IDs are automatically assigned by the order of addition with the first edge being added having an ID of 0, the second having an ID of 1 and the following corresponding edges increasing in their ID numbers respectively. The node and edge attributes defined in the dataset are the user-defined named features that store the graph-specific properties of the nodes and edges in the graph network.

In the model, there are three sets of attributes for each user: class, belief and bias. The bias and belief attributes exist as integer type features while the class attribute based on the user classes defined exist as string type features. The string type features are transformed with one hot encoding to categorical integers, while the other features exist as tensors with feature normalisation performed to ensure that the tensor values are the numerical equivalents. The features are created to contain as much information as possible as found in graph features of social networks related to their users.

Results

The dataset is divided into sets - training (60%), validation (20%) and test (20%) as seen in the figure. For initialization, being a node classification task, several nodes are activated and fed with a learnable embedding.

```
n_train = int(n_nodes * 0.6)
n_val = int(n_nodes * 0.2)
```

Training and Validation Percentage

For the user classes, the values are 0 (Ignorant nodes), 1 (Informed nodes), 2 (Mirror nodes) and 3 (Super nodes) (see Appendix C). Printing the tensors of each attribute shows 80 tensor entries indicating that each entry is an attribute of each node as seen in the figure.

```
tensor([3, 3, 1, 2, 1, 0, 1, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,
        0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,
        1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,
        1, 1, 0, 0, 0, 2, 0, 1], dtype=torch.int8)
```

Tensor values for Class attribute

Features are confirmed by printing the tensors of each attribute and it shows 34 tensor entries with each entry indicating the feature of each node as shown in the figure. Simulations are run using each feature as the input into the network. The dataset is divided into sets created as masks with nodes assigned - training (60%), validation (20%) and test (20%), like the synthetic network.

```
Node features
{'feat': tensor([0, 1, 0, 0, 1, 0, 0, 0, 1, 2, 0, 0, 0, 1, 0, 2, 0, 0, 2, 0, 0, 1, 0, 0,
        2, 0, 0, 0, 2, 0, 0, 0, 2, 1]), 'boy': tensor([0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1,
 0, 0,
        1, 0, 0, 0, 0, 0, 0, 0, 1, 1]), 'label': tensor([3, 3, 0, 2, 0, 0, 0, 0, 2, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0,
0, 0, 2,
        1, 0, 0, 0, 1, 0, 0, 0, 1, 0]), 'train_mask': tensor([ True,  True,  True,  True,  True,  True,  True,  True,  T
rue,  True,
         True,  True,  True,  True,  True,  True,  True, False, False, False,
        False, False, False, False, False, False, False, False, False, False,
        False, False, False, False]), 'val_mask': tensor([False, False, False, False, False, False, False, False, False,
 False,
        False, False, False, False, False, False, False,  True,  True,  True,
         True,  True,  True,  True,  True,  True,  True, False, False, False,
        False, False, False, False]), 'test_mask': tensor([False, False, False, False, False, False, False, False, False
, False,
        False, False, False, False, False, False, False, False, False, False,
        False, False, False, False, False, False, False,  True,  True,  True,
         True,  True,  True,  True])}
```

Zachary's network (RN)- User features tensors

APPENDIX D

Evaluation and Case Study

The model performance is compared against a baseline model.

NetTv3 Model Overview

The final model - NetTv3 was created based on a representation learning mechanism enhanced by a neural network architecture that differentiates users in a network based on their belief system. The data was modelled as labels - attributes associated with users in the network, which are in turn represented as nodes within a graph. The model implements a GraphSAGE architecture using DGL (Deep Graph Library) to perform node classification on social network data. The simulation was performed with a synthetic dataset and validated with a real-network (karate club) with the dataset created to provide a standard on which enabled the model to be tested in three different configurations - the three configurations are based on learning an embedding with each attribute of nodes used as the label in the graph network. Table 1 shows detailed information on the dataset used.

Dataset	Type	Number of Nodes	Number of Edges	Number of attributes	Initialization rate
Synthetic	Social Network	80	307	3	2.5%
Zachary's Karate Club	Social Network	34	190	3	2.5%

Table 1: Dataset Statistics

The simulation setup uses a three-layer GraphSAGE and evaluates prediction accuracy on a test set consisting of a portion of the nodes in the graph network. For training the datasets, the hyperparameters are optimised on the synthetic network only and use the same set of parameters for the real network. The models for a maximum of 100 epochs (training iterations) using Adam Optimizer (Kingma and Ba, 2017) with a learning rate of 0.01. The hidden layers have a size of 32 units and feature an activation function (ReLU). There is no early stopping (model runs for a fixed number of epochs) as loss and accuracies are allowed for the eternity of the epochs. The dataset splits are as follows - 60% training, 20% validation and 20% test.

Parameter	Value
Hidden Layers	2
Network Input	embeddings
Optimizer	Adam
Learning rate	0.01
Number of epochs	100
Loss function	Cross Entropy Loss

Table 2: Parameter Values

The validation metrics - Cross Entropy loss and the dataset splits accuracy, computes how well the model performs in terms of how nodes are well classified with respect to the expected output as described in Table 3. The accuracy values are an average of the model training over the three configurations and enables the model performance to be evaluated.

Model	Average Training Accuracy	Average Validation Accuracy	Average Test Accuracy
Synthetic Network	1.0	0.78	0.86
Zachary's Network	1.0	0.79	0.88

Table 3: Model Scores

Evaluation Model

The model's performance is evaluated against the performance of a GCN based Node classification using representation learning - The Semi-supervised classification model (Kipf and Welling, 2017). Their approach relies on spectral graph convolutional neural networks, which were first developed *et al.*, 2014) and later extended by (Defferrard, Bresson and Vandergheynst, 2017) with fast localised convolutions and features the task of transductive node classification within networks of significantly larger scale. Their model uses a two-layer GCN and evaluates prediction accuracy on a test set. Table 1 below shows the dataset statistics used in the simulation. Bruna et al., (2014) and later extended by Defferrard, Bresson and Vandergheynst (2017) with fast localised convolutions. It includes the task of transductive node classification within networks of noticeably larger scale. Their model uses a two-layer GCN and evaluates prediction accuracy on a test set. Table 4 below shows the dataset statistics used in the simulation.

Dataset	**Type**	**Nodes**	**Edges**	**Classes**	**Features**	**Label rate**
Citeseer	Citation Network	3327	4732	6	3703	0.036
Cora	Citation Network	2708	5429	7	1433	0.052

Table 4: Dataset Statistics

The implementation is done as a neural network model based on graph convolutions built by stacking multiple convolutional layers to perform semi-supervised node classification using a layer-wise propagation rule.

$$H^{(l+1)} = \sigma\left(\underline{D}\blacksquare^{-1/2}\,\underline{A}\,\underline{D}\blacksquare^{-1/2} H^{(l)} W^{(l)}\right) \qquad (1)$$

where, $\underline{A} = A + I_N$ is the adjacency matrix of the undirected graph G with added self-connections. I_N is the identity matrix. $\sigma(\cdot)$ denotes an activation function, such as the $ReLU(\cdot) = max(0,\cdot)$. Their model features a two-layer GCN for semi-supervised node classification on a graph. Using a method based on spectral graph convolutional neural networks, with fast localised convolutions, their model performs transductive node classification within networks of significantly larger scale. Network parameters are shown inTable 5.

Parameter	**Value**
Hidden Layers	1
Network input	Node Features
Optimizer	Adam

Learning rate	0.01
Number of training epochs	200
Loss Function	Cross-entropy

Table 5: Model Parameters

They looked at three citation network datasets for training: Citeseer, Cora, and Pubmed. The datasets include lists of citation linkages between documents, sparse bag-of-words feature vectors for each document, and class labels for each document. The authors generate a binary, symmetric adjacency matrix A from the citation linkages by treating them as (undirected) edges. They employ all feature vectors connected to the dataset's nodes for training, but just 20 labels per class in each dataset.

Analysis

In evaluating the performance of NetTv3, a comparison is made against the methods as in Kipf and Welling (2017) semi-supervised node classification and draw inference from the results. While the model of Kipf and Welling (2017) uses significantly larger datasets than that of NetTv3, major similarities exist between both models that enable direct comparisons to be made. These similarities are:

▪ Learning operation

The learning operation in focus is that of node-level representation learning for node classification purposes.

▪ Graph Neural Network (GNN)

A GNN based framework is used to address the problem of classifying nodes in the network.

▪ Datasets

The datasets used are implemented as graph networks with nodes representing users and edges their relations. The nodes in the dataset have features as feature vectors associated with them. The nodes in the dataset also belong to one of several classes which serve as the labels of the datasets used in Kipf and Welling (2017).

▪ Classifier

The classifier centres around a three-layer GraphSAGE network for NetTv1 and a two-layer GCN for Kipf and Welling (2017).

▪ Model Optimisation

Hyperparameters are defined for the classification phase architecture and the neural network architecture is optimised with parameters as seen in Table 2 and Table 5.

▪ Results

The accuracy of the learning operations is evaluated using accuracies generated from validating the model performance on split datasets.

The model of Kipf and Welling (2017) uses a transductive method that infers the labels of unlabelled instances without generalising to unobserved instances. The hidden layers are concatenated and fed to a SoftMax layer to predict the class label of the instance. In the case of Kipf and Welling (2017) the instances are the input into the graph. They use the node features (word count vector as its features, normalised so that they sum up to one) as the input feature vector into the input layer with the task of the classifier to predict the category of a given paper. This produces an effective graph representation model that can naturally combine structure information and node features in the learning process.

This method's perceived drawback is that each node receives features from all of its neighbours, regardless of whether the node has a dense or sparse link. The node degrees (indegree and outdegree) of real-world graphs, which might range from one to hundreds, thousands, or even millions as seen on OSNs, are not accurately reflected by this. As a result, some nodes would require more neighbours to obtain adequate information, while others might aggregate too extensively, rendering their own features unimportant in their state update.

By using the inherent node features to train the network, their model also doesn't take into account the individuality of each node in interactions in the graph. Additionally, the GCN employed did not pick or weight the features in the feature vector as weights on the node edges. In this scenario, a new representation can be created by aggregating noisy information, confusing the classifier and lowering classification accuracy.

In contrast, an inductive based method is used in NetTv3 which allows the model to factor in a node's local role in the graph, as well as its global position using the structural properties of a node's neighbourhood. The approach used in this research uses node embeddings to return a learnable vector for the nodes in the graph network. This research also uses a GraphSage NN architecture which allows for node connections to be sampled at each layer.

This ensures that information on nodes within the ego neighbourhood of a node is utilised at each time step and at each layer. This is particularly useful to the research as the users/nodes using a learnable embedding are trained in a way that provides new data while not changing the attributes of the nodes. New states can emerge without an underlying change in the user/node fundamentals.

The experimental setup for both models is the same with the dataset statistics summarised in Table 6. In the citation network datasets used in Kipf and Welling (2017) - Citeseer and Cora - nodes are documents and edges are citation links while in NetTv3. The label rate is calculated by dividing the total number of labelled nodes in each dataset by the number of labelled nodes used for training.

Dataset	Type	Nodes	Edges	Classes	Label rate	Average Degree
Citeseer - Kipf and Welling	Citation Network	3327	4732	6	0.036	2.8231
Cora - Kipf and Welling	Citation Network	2708	5429	7	0.052	3.8981
Synthetic network - NetTv3	Social Network	80	307	4	2.5	0.55
Zachary’s Karate Club - NetTv3	Social Network	34	190	4	5.88	4.5882

Table 6: Datasets Overview

Semi-supervised embedding is used to compare the models (NetTv3 - real and synthetic) to the same baseline methods of label propagation. The results are summarised in Table 7. The accuracy values denote classification accuracy in percent. For NetTv3, the accuracy reported is the mean accuracy of 20 runs. Results for all other baseline methods are taken from Kipf and Welling (2017).

Method	Dataset	Accuracy

Kipf and Welling	Cora	81.5%
Kipf and Welling	Citeseer	70.3%
NetTv3	Synthetic	85.67%
NetTv3	Real-Network	88.23%

Table 7: Summary of Results in terms of classification accuracy

Propagation of feature information from neighbouring nodes at every layer improves classification performance. NetTv3 by aggregating features from a node's local neighbourhood to generate embeddings instead of training individual embeddings for each node, can include state information from unseen nodes and accurately show the effects of neighbouring nodes on a node's decision on whether to adopt a belief.

The proposed model has shown that by training a learnable embedding whilst incorporating node features in the learning algorithm, it is possible to model belief adoption amongst agents using the agent's features, the topological structure of each agent's neighbourhood as well as the distribution of node features in the neighbourhood.

APPENDIX E

RESEARCH ETHICS

Disclaimer Form

The following declaration should be made in cases where the researcher and the supervisor (where applicable) conclude that it is not necessary to apply for ethical approval for a specific research project.

PART A: TO BE COMPLETED BY RESEARCHER

Name of Researcher:	FRANKLIN C. CHUKWUMA
School	COMPUTING AND DIGITAL TECHNOLGIES

Student/Course Details (If Applicable)		
Student ID Number:		C0026831I
Name of Supervisor(s)/Module Tutor:		
PhD/MPhil project: ☒		
Taught Postgraduate Project/Assignment: ☐	Award Title:	
Undergraduate Project/Assignment: ☐	Module Title:	

Project Title:	MODELLING IRRATIONAL AGENT BELIEFS		
Project Outline:	▪ Review and critically analyse the available literature on OSNs, Agent-Based Modelling (ABM), Artificial Neural Networks (ANN) and Artificial Intelligence (AI). ▪ Develop a method to accurately detect, classify and stop false information diffusion. ▪ Identify key nodes that propagate false information and shut them down. ▪ Investigate the role of bias in the links that nodes establish, the state of nodes and its role in the diffusion process ▪ Validate the model by applying the model to real-life applications.		
Give a brief description of research procedure (methods, tests etc.)	The objectives of the project classify the research as an applied research. With specific objectives asked and research questions to be answered, an inductive research approach is used as well as quantitative methods. Simulations will be used as the primary project evaluation method as this offers the best way to define and create scenarios relating to the project's objectives. Primary data will be collected from the simulations and analysed. Secondary data from existing diffusion research will also be used within the project. Data analysis will be inferential as multiple variables will be used and considered during the simulations. Results will be presented in a descriptive and visual format using the appropriate data visualization tools.		
Expected Start Date:	05/11/2018	Expected End Date:	04/11/2021

Declaration

I/We confirm that the University's Ethical Review Policy has been consulted and that all ethical issues and implications in relation to the above project have been considered. I/We confirm that ethical approval need not be sought. I/We confirm that:

University Research Ethics Committee – February 2018

The research does not involve human or animal participants	☒
The research does not present an indirect risk to non-participants (human or animal).	☒
The research does not raise ethical issues due to the potential social or environmental implications of the study.	☒
The research does not re-use previously collected personal data which is sensitive in nature, or enables the identification of individuals.	☒
Has a risk assessment been completed for this project?	☒ Yes ☐ N/A

Signature of Researcher:	Franklin C. Chukwuma	Date:	03/03/2020
Signature(s) of Project Supervisor(s) (If student) OR Signature of Head of Department/ Senior researcher (if staff)		Date:	5th March 2020

NB: If the research departs from the protocol which provides the basis for this disclaimer then ethical review may be required and the applicant and supervisor (where applicable) should consider whether or not the disclaimer declaration remains appropriate. If it is no longer appropriate an application for ethical review **MUST** be submitted.