TOWARDS MODELING AND MITIGATING MISINFORMATION PROPAGATION IN ONLINE SOCIAL NETWORKS

A DISSERTATION SUBMITTED TO

THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR

THE DEGREE OF

DOCTOR OF PHILOSOPHY

IN

COMPUTER ENGINEERING

By

Tolga Yılmaz

January 2023

Towards Modeling and Mitigating Misinformation Propagation in Online Social Networks
By Tolga Yılmaz
January 2023

We certify that we have read this dissertation and that in our opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.

Özgür Ulusoy(Advisor)

İbrahim Körpeoğlu

Uğur Güdükbay

Ahmet Coşar

İsmail Sengör Altıngövde

Approved for the Graduate School of Engineering and Science:

Orhan Arıkan
Director of the Graduate School

ABSTRACT

TOWARDS MODELING AND MITIGATING MISINFORMATION PROPAGATION IN ONLINE SOCIAL NETWORKS

Tolga Yılmaz
Ph.D. in Computer Engineering
Advisor: Özgür Ulusoy
January 2023

Misinformation on the internet and social media has become a pressing concern due to its potential impacts on society, undermining trust and impacting human decisions on global issues such as health, energy, politics, terrorism, and disasters. As a solution to the problem, computational methods have been employed to detect and mitigate the spread of false or misleading information. These efforts have included the development of algorithms to identify fake news and troll accounts, as well as research on the dissemination of misinformation on social media platforms. However, the problem of misinformation on the web and social networks remains a complex and ongoing challenge, requiring continued attention and research. We contribute to three different solution aspects of the problem. First, we design and implement an extensible social network simulation framework called Crowd that helps model, simulate, visualize and analyze social network scenarios. Second, we gamify misinformation propagation as a cooperative game between nodes and identify how misinformation spreads under various criteria. Then, we design a network-level game where the nodes are controlled from a higher perspective. In this game, we train and test a deep reinforcement learning method based on Multi-Agent Deep Deterministic Policy Gradients and show that our method outperforms well-known node-selection algorithms, such as page-rank, centrality, and CELF, over various social networks in defending against misinformation or participating in it. Finally, we promote and propose a blockchain and deep learning hybrid approach that utilizes crowdsourcing to target the misinformation problem while providing transparency, immutability, and validity of votes. We provide the results of extensive simulations under various combinations of well-known attacks on reputation systems and a case study that compares our results with a current study on Twitter.

Keywords: Misinformation Propagation, Online Social Networks, Reinforcement Learning, Cooperative Games, Blockchain, Crowdsourcing, Reputation Systems.

ÖZET

ÇEVRİMİÇİ SOSYAL AĞLARDA YANLIŞ BİLGİ YAYILIMININ MODELLENMESİ VE AZALTILMASI ÜZERİNE

Tolga Yılmaz
Bilgisayar Mühendisliği, Doktora
Tez Danışmanı: Özgür Ulusoy
Ocak 2023

İnternet ve sosyal medyada yer alan yanlış bilgiler toplum üzerinde oluşturabileceği güvensizlik ve sağlık, enerji, politika, terörizm ve afetler gibi önemli alanlarda insan karar mekanizmalarında oluşturabileceği etki yüzünden ciddi bir endişe haline gelmiştir. Yanlış bilgilerin tespiti ve yayılımının önüne geçmek için hesaplamalı yöntemler devreye alınmaya başlamıştır. Bu yöntemler, sahte haberler ve trol hesapların algoritmik açıdan tespiti ve yanlış bilgilerin sosyal medyada yayılması üzerine geliştirilmektedir. Fakat, internette ve sosyal medyada yer alan yanlış bilgi problemi sürekli ilgi ve araştırma gerektiren karmaşık ve devam eden bir problemdir. Biz bu problemin çözümünde üç farklı açıdan katkı sağlamaktayız. İlk olarak, çeşitli sosyal ağ senaryolarında modelleme, simulasyon, görselleştirme ve analiz için kullanılabilecek *Crowd* isimli genişletilebilir bir sosyal ağ çatısı tasarlayıp geliştirdik. İkinci katkımız ise, yanlış bilgi yayılımının oyunlaştırılarak ağda yer alan düğümler arasında bir işbirlikçi oyun haline getirilmesi ve yayılımın farklı şartlarda nasıl gerçekleştiğinin anlaşılması üzerinedir. Sonrasında, düğümlerin ağ seviyesinde kontrol edildiği ağ seviyesinde bir oyun tasarladık. Bu oyunda, *Multi Agent Deep Deterministic Policy Gradients* yöntemini baz alarak geliştirdiğimiz derin pekiştirmeli öğrenme metodunun *page-rank*, *centrality* ve *CELF* gibi düğüm seçme algoritmalarından daha iyi performans verdiğini gösterdik. Son olarak, yanlış bilgi probleminde şeffaflık, değişmezlik ve doğruluk gibi kriterleri sağlayabilecek bir blokzincir ve derin öğrenme hibrit yöntemini kitle kaynak kullanımı ile önerdik. İtibar sistemleri üzerine iyi bilinen saldırılar altında detaylı simulasyonlar ile yöntemin performansı ölçülmüş ve Twitter üzerinde yer alan bir çalışmayla kıyas yapılmıştır.

Anahtar sözcükler: Yanlış Bilgi Yayılımı, Çevrimiçi Sosyal Ağlar, Pekiştirmeli Öğrenme, İşbirlikçi Oyunlar, Blokzincir, Kitle Kaynak Kullanımı, İtibar Sistemleri.

Acknowledgement

I am deeply grateful for all the support and guidance from my advisor Prof. Dr. Özgür Ulusoy, during my study over the years. Through his patience, this is completed.

I would like to express my gratitude to Prof. Dr. İbrahim Körpeoğlu and Prof. Dr. İsmail Sengör Altıngövde for their valuable time and advice for the duration of my thesis monitoring committee.

I would like to thank Prof. Dr. Uğur Güdükbay and Prof. Dr. Ahmet Coşar for their kind acceptance to be on my dissertation jury.

I would like to thank all the present and past members of Bilkent University and the Computer Engineering Department for facilitating my studies, starting from my undergraduate journey until now, in a warm and challenging environment.

I would also like to wholeheartedly thank Mahmut Ortaboz, Mustafa Can Çavdar, and Nagihan Özgür for their support during various periods of my study.

Finally, I am indebted to my parents and sister for every part of my life. I would like to dedicate this thesis to them.

Contents

List of Figures

List of Tables

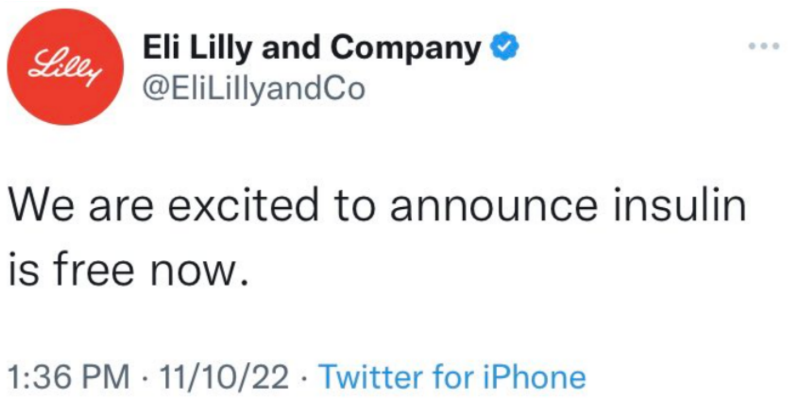
Chapter 1

Introduction

Where people receive news now includes online media such as news websites and social networks in addition to traditional media such as TV, radio and newspapers [1], mainly due to the convenience in terms of quickness and socializing aspects [2]. These platforms have become sources to spread not just news but also ideas and have been utilized as tools to influence people for different purposes, such as altering the public mind to vote for certain parties [3], advertising certain products for commercial advantage [4], generating awareness for certain issues in health [5], and global problems [6, 7].

The fast progress in communication comes with disadvantages. With the fast pace of information exchange, the validation process also seems to be done faster, less thoroughly, and completely overlooked in most cases [8]. As a result, people tend to believe the information they see on the Internet and even help it propagate to others on social networks [9]. While these fake news stories occur daily, a recent example is given in Figure 1. In this example, a fake account successfully impersonated a company that manufactures insulin and caused a major stock crash for the company.

Figure 1.1: A misinformation tweet by a fake account impersonating a health-care company


Eli Lilly and Company
@EliLillyandCo

We are excited to announce insulin is free now.

1:36 PM · 11/10/22 · Twitter for iPhone

Consequently, the proliferation of misinformation has become a major concern in recent years. With the rise of social media and the ease of access to information, it has become increasingly difficult to distinguish between factual and false information. This has led to the spread of false narratives and even conspiracy theories in politics [3], vital health-related issues such as pandemics [10], disasters [11], energy [12], terrorism [13], and armed conflicts [14], which can have serious consequences for individuals and society as a whole.

To combat the spread of misinformation, some organizations have attempted to raise awareness about "fake news" by manually curating content. Duke Reporter's Lab[1] maintains a database of fact-checking websites around the world. However, this approach has its limitations. These organizations are often centralized, meaning that they are controlled by a small group of people. This issue can make it difficult for them to guarantee transparency and prevent manipulation, such as cherry-picking in favor of a particular party [15, 16, 17]. In addition, "fact-checking" is labor-intensive, and the curators are unlikely to able to keep up with emerging news stories or cover the entire news ecosystem. By the time a human curator checks the content, it would already propagate to many people [18]. [19] surveys the recent efforts on automated fact-checking to accelerate the process.

[1]`https://reporterslab.org/fact-checking/`

News literacy education is also important to teach individuals how to critically evaluate the information they encounter and to understand the various biases and motivations that can influence the production and dissemination of news. Receiving this education can help individuals to become more discerning consumers of information and to avoid being swayed by false narratives [20]. However, as in any form of education, news literacy education can take time to have an effect, and it may not reach everyone who needs it.

Computational approaches focus on the definition of the misinformation concept [21], its propagation mechanisms [22, 23], and detection and prevention of its propagation [24, 25, 26]. In addition, recent studies on this field iterate over various types of textual, behavioral, and media aspects to identify misinformation using machine learning as explained by [27], [28], and [29]. Computational efforts should also take the concepts of decentralization, transparency, and objectivity into account and should have demonstrative aspects against tampering for any outside agenda [30, 31, 32].

It has been reported that large social media companies use techniques to detect fake news and troll accounts [33], which are intentionally misleading or disruptive online postings. However, despite such efforts, these platforms continue to be sources of misinformation [21] as ongoing research on the spread of misinformation on these platforms highlights the ongoing issue [28]. In addition, these platforms are commercially motivated and centrally controlled. This means that they may not always be able to provide transparency and objectivity in how they store, distribute, and present content and interactions [34, 35].

It is evident from the literature that misinformation spread is encouraged by various threat actors with the help of fabricated text, media, and an army of fake accounts that are used in the process. There is an ongoing game maintained by various actors, most frequently for various reasons such as political propaganda, that could have profound effects on both internal and international stages.

In this work, we first developed a social network simulation framework that would enable us to easily define and run various types of scenarios on various social networks. We name our approach “network as code”. We describe our approach as well as a set of examples in Chapter 3.

Often, the spread of misinformation does not occur as an isolated incident but happens in a repeated fashion. The users of various social media platforms encounter various stages and forms of misinformation daily, and so far, there does not seem to be a permanent solution. Therefore, it might be useful to address the misinformation problem as a repeated game in a social network environment where there exist multiple stages involving multiple actors that affect the end-user (whether they participate or not) in some way.

In the second part of this work, we approach the misinformation propagation problem from a game-theoretic perspective. The setting where spatial relation between players plays a role in deciding the game’s outcome is called spatial games [36, 37]. In the particular context of social networks as graphs, we approach the problem as a game on graphs. A vast amount of literature exists on graph games, and a particular study [38] approaches the problem of cooperation on graphs. Their work indicates that cooperation is only favorable if players’ benefit/cost ratio exceeds the average number of degrees (i.e., connections) in the graph. They show that otherwise, cooperation is not favorable. We construct a misinformation exchange game based on cooperative game theory. Our simulations also yield that the probability of countering misinformation is increased if the benefit/cost ratio exceeds the average degree. This approach, however, displays some disadvantages, such as the ambiguity of determining the actual or expected values of benefits and costs.

We then construct another game that is played at the network level, where players are given a set of nodes and try to maximize the number of affected nodes. Since we find this setup suitable for a learning environment for a multi-agent reinforcement learning setting, we propose to utilize a method based on the Multi-Agent Deep Deterministic Policy Gradient algorithm. For the same setting, we also implement other algorithms based on the highly-established centrality, page-rank, and CELF algorithms and share the comparison results of these methods.

In the final part of this work, we approach the news-sharing notion as an issue of trust. As in any transaction between two parties, the news delivery process between the curator and the news-reader is established on some notion of trust [39] and can be investigated from a trust management point of view. People should be able to find reputable news sources they can trust, while news curators are rewarded in parallel with the quality of their work and establish a form of reputation. However, the research on trust management systems that are engineered and deployed for various tasks shows that these systems can also be exploited with various types of attacks [40, 41, 42]. Nonetheless, these systems deliver resilient enough functions that we incorporate into our lives, such as e-shopping [43].

A trust-based platform that can incorporate this kind of transparency and that can prove the validity of transactions with a transaction-based reward mechanism is a blockchain. A blockchain is a distributed database to deliver specific functionality while ensuring transaction correctness. In our work, we explore a blockchain-based crowdsourcing system using the Ethereum blockchain and smart contracts as a proof-of-concept solution where people create and vote on news stories. We employ a reward-based reputation system that encourages people to post correct stories and identify the truthfulness of stories correctly by voting on them. A blockchain-based solution practically ensures that the transactions are not altered or in the custody of a single body while providing incentives in the form of rewards for user traction. However, we show that, as in any reputation system, this system is also subject to various forms of attacks involving malicious users and should not be presented as a silver bullet but rather needs to be supported by various mechanisms for increased resilience and robustness. To this purpose, we reinforce the system with a deep learning architecture that learns the malicious actions of the users and joins the final decision of the crowd on the truthfulness of news stories. We simulate various types of attack forms as well as provide a case study based on Twitter Birdwatch. The experiments yield promising results and encourage further studies, which we specify in the study.

The rest of this dissertation is structured as follows. Chapter 2 reviews the existing work on detecting and preventing misinformation. We also review the literature on trust management systems. Finally, we briefly introduce blockchain systems as this will form a basis for understanding how our proposed system tries to enforce “trust”. In Chapter 3, we describe *Crowd*, which we developed to be able to perform various simulations in social networks. In Chapter 4, we introduce and discuss our findings on how misinformation spreads in a social network and list our future research directions. In Chapter 5, we describe the underlying methods of our blockchain-deep learning approach to detect misinformation. Finally, Chapter 6 summarizes and concludes our research.

Chapter 2

Background and Related Work

2.1 Efforts to Mitigate Misinformation

We divide the research on misinformation into three main parts. First, we list the literature on how misinformation is viewed and modeled. Then, we give a short review of the studies on the detection and containment of misinformation. We finally provide an overview of more specific studies involving game theoretic approaches, crowdsourcing, and blockchains.

2.1.1 Definitions, Analysis, Modeling

Kumar et al. [21] provide an analysis from a "false information" perspective. According to their categorization, false information can be classified into *intent* and *knowledge*, which are further classified into misinformation, disinformation, opinion-based, and fact-based, respectively. An example of misinformation would be urban legends, and an example of disinformation would be fake news. For opinion-based and fact-based, we have examples such as fake reviews and hoaxes, respectively.

Rubin et al. [44] define three types of fakes: serious fabrications, large-scale hoaxes, and humorous fakes. Serious fabrications are false or misleading information presented as accurate and can cause harm or damage. Large-scale hoaxes are elaborate schemes or scams that involve the creation and spread of false information on a large scale, often for financial or political gain. Finally, humorous fakes are intentionally humorous or satirical pieces of false information that are not meant to cause harm or be taken seriously. Wardle expands on this classification by identifying several types of misinformation: satire or parody, false connection, misleading content, false context, imposter content, manipulated content, and fabricated content.

- *Satire or parody* refers to false or humorous information that is not meant to be taken seriously and does not cause harm.
- *False connection* refers to content, material, or headlines that do not match and can be misleading.
- *Misleading content* refers to cherry-picking information that might harm or benefit someone.
- *False context* refers to true content that is presented in a false or misleading context.
- *Imposter content* refers to cases of impersonation, where someone pretends to be someone else to spread false or misleading information.
- *Manipulated content* refers to images or information that have been altered to deceive.
- *Fabricated content* refers to entirely new and false information created to deceive.

Examples of these types of misinformation might include a satirical news article that is mistaken for a real news story, a social media post that presents false information as accurate in order to benefit a particular person or group, or a manipulated image that is used to spread false information about an event or person.

Therefore, it is essential to be aware of these different types of misinformation to identify and counter them effectively.

While there exist recent studies that have similar classifications [45, 46] or ones with from an intent perspective [29] regarding the types of misinformation or fake news, we have a broader perspective on false information in our work and treat misinformation as a general term encapsulating all the mentioned subcategories identified by various studies.

2.1.2 Detection

Shu et al. [47] survey fake news detection techniques involving data mining and list the following feature categories: News Content (Linguistic-based, Visual-based) and Social Context (User-based, Post-based, Network-based). Conroy et al. [48] make the division as linguistic and network approaches.

Other studies approached the problem by combining a variety of features. Qazvinian et al. [24] implement a classifier based on content-based, network-based, and microblog-specific features in a tweet data set. Gupta et al. [49] utilize user and tweet features with Naive Bayes and Decision trees to detect tweets that contain fake images during Hurricane Sandy. Yang et al. [50] experiment with rumor detection on Weibo and add client and location-based features. Perez-Rosas et al. [51] describe a classification method using linguistic features. Similarly, Volkova et al. [52] approach the problem from a linguistic perspective and list these features to generate a classification method. Kwon et al. [53] use user, structural, linguistic, and temporal features and monitor these features' significance in detecting rumors over time. They continue to build a classifier on the best-performing features. Rubin et al. [54, 55] use rhetorical structure theory to create relations of significant parts in the text.

Some studies investigate the spread nature of misinformation on social networks. Jin et al. [56] establish a method that utilizes opposing views to detect false information. Ruchansky et al. [57] utilize three characteristics of fake news;

the article itself, the user behavior on the article, and the community that spreads it. They build a classifier of three modules, the first of which uses the first two characteristics with a Recurrent Neural Network. The second module is learning the third characteristic, and the third module combines the results. Wu et al. [58] use a method that represents users with network structures as embeddings and Long Short Term Memory (LSTM) to learn the propagation structure in the network to classify misinformation. Ma et al. [59] represent propagation structures as trees and use a technique called *Propagation Tree Kernel* to learn these structures to identify rumors.

Comprehensive reviews of recent misinformation detection techniques are given in [29, 28].

2.1.2.1 Malicious User Detection

There exists a variety of user types that could take part in the creation and spread of misinformation including bots, trolls, and Sybil Accounts.

Ferrara et al. [60] give an overview of bot detection techniques. They list three main categories of techniques; bot detection systems based on social network information; crowdsourcing, and machine-learning methods.

In one of the network-based studies, Cao et al. [61] introduce a method called SybilRank based on network propagation that can be used to identify fake accounts in social networks. They give results for an online social network in Spain.

Crowdsourcing-based studies include the ones by Wang et al. [62], and Ghosh et al. [63] as they propose crowdsourcing-based systems to detect Sybil accounts.

Varol et al. [64] employ a feature-based bot detection solution based on user, friend, network, temporal, content, and sentiment features. Additionally, they report multiple classes of bot types. Wang et al. [65] describe a method to detect Sybil accounts using clickstream data: the collection of user actions. They show a greater performance over using click-transitions and a Support Vector Machine-based classifier that combines click-based features (for instance, counts of clicks for specific actions).

Cresci [66] outlines the studies on social bot detection in the past decade, how bots have evolved, and the research on their detection. Latah [67] provides a taxonomy of different types of bots and types of attacks performed by them in a recent study. Alharbi et al. [68] also survey types of identity deception in social networks, including Sybil attacks, sockpuppets, and social botnets.

2.1.3 Containment

Budak et al. [69] define the misinformation limitation problem as limiting the spread of an adversarial campaign with a good one. In order to achieve that, they need to find which influential nodes should start the campaign against. They show that the problem is NP-hard, use heuristics and greedy approaches, and show that node degree centrality is as good as greedy approaches.

Gupta et al. [70] describe a system that can classify tweets based on their credibility after collecting user feedback and learning a Support Vector Machine based classifier from them.

Nguyen et al. [71, 72] define a concept of *node protectors*, which is the smallest subset of nodes when contained, which will prevent misinformation from spreading. Furthermore, they experiment with heuristic-based methods to identify the nodes in question and report competitive results. Some recent studies focus on the cost effectiveness of node blocking in the spread of misinformation. For instance, Pham et al. [73] offer greedy algorithms in a multi-topic setting. He et al. [74] identify two strategies to restraint using various methods and oppose rumor spreading using truth spreading.

Farajtabar et al. [75] define the misinformation spread as a reinforcement learning problem by representing states, mitigation actions, and reward functions to learn. They report their results on the learned mitigation framework.

Hosni et al. [76] explore the spread of rumor during the period of emerging news. They propose a novel multi-rumor propagation model and a Markov Chain based formulation for user opinions.

Finally, it is worth mentioning that the studies on containing misinformation has been intertwined with the ones on *influence maximization*. Li et al. [77] compile a recent and comprehensive survey on influence maximization using deep learning methods.

2.1.4 Game Theoretic Approaches

In this work, we propose a model for misinformation propagation in social networks conforming to a game-theoretic model. Information diffusion on graphs has been studied using game-theoretic models previously. In [78], a framework based on evolutionary game-theoretic models on graphs has been proposed and tested on various synthetic and real networks. Yang et al. [79] propose and analyze an information spread model based on the diffusion of competitive information on graphs. The diffusion of rumor and misinformation based on game-theoretic models has also been studied recently. Kumar et al. [80] create a model for misinformation spread. Their approach is different from ours in that they deal with cooperation as a means to spread misinformation, which is the opposite of our approach. Li et al. [81] describe an evolutionary game with a punishment mechanism and a probabilistic function to update node strategies. Xiao et al. [82] introduce internal and external factors and use them in a rumor propagation model under a rumor/anti-rumor setting. Askarizadeh et al. [83, 84] explore an evolutionary model incorporating factors that affect rumor propagation and its control.

2.1.5 Crowdsourcing and Blockchain Approaches

Crowdsourcing can be an effective way to identify true and fake news through the use of collective intelligence. By gathering the perspectives and expertise of a diverse group of individuals; it is possible to assess the veracity of news stories and determine whether they are genuine or fabricated.

A considerable number of studies in the computer science literature have demonstrated the effectiveness of crowdsourcing in identifying true and fake news. In one of these studies, Kittur et al. [85] show that crowdsourced fact-checking can be more accurate than individual fact-checking, as the collective wisdom of the crowd can often outweigh the biases and limitations of any one individual.

Tacchini et al. [86] use Logistic Regression and crowdsourcing to detect hoaxes in Facebook posts. Kim et al. [87] describe a system that aids the crowdsourcing process to better detect and prevent misinformation in social networks. Chen et al. [88] propose an entropy-based mechanism for a quorum-based fake-news prevention system that uses expert voting on fake news candidates, with Hyperledger as the persistence layer. Avelino and Rocha [89] suggest BlockProof for verifying the authenticity and integrity of web content using a blockchain solution. Their experiments mainly focus on the feasibility of the average response times of the system rather than how and how well the system identifies fake news. In [90], Sengupta et al. propose ProBlock, which is a secure voting system with an emphasis on reviewer privacy.

Pennycook and Rand [91] measure the effectiveness of crowdsourcing to identify credible news sources. Denaux et al. [92] work on identifying the credibility of reviews to increase the reliability of crowdsourcing tasks. Soprano et al. [93] provide an in-depth, multidimensional analysis of crowdsourcing for misinformation assessment, denoting the effectiveness of crowdsourcing for this problem. Teixeira et al. [94] explore the preliminaries of a blockchain-based journalism system.

Shan et al. introduce Poligraph [95], which aims to provide a human review-machine learning combined approach for expert review on blockchain and focuses on latency and byzantine fault tolerance without putting the accuracy of the machine learning approach forward, where our work is reputation-related and designed for the use of the general public; hence, the problem requires solving the issues such as the attacks under uncertainty of user intent. A similar work to ours is ABC-Verify [96], which also utilizes a blockchain and machine learning approach. However, they use machine learning for news content, whereas we use a classifier for users' behavior.

2.2 Simulating Social Networks

Social networks have been vastly studied from many aspects, including but not limited to information diffusion, epidemic spread analysis, and influence projection. The need to model, simulate, analyze and visualize social networks led to the development of computational tools. Researchers either develop their own code or utilize existing tools by fitting the problem at hand to the context of the tool they use. Agent-based model (ABM) simulation tools are one aspect of solutions to this problem. An ABM is a type of simulation in which the behavior of individual agents (i.e., autonomous decision-making entities) is modeled in order to understand and predict the behavior of complex systems.

NetLogo [97] is a programming language and integrated development environment (IDE) used for creating simulations and modeling complex systems. It is particularly well-suited for use in education, but it is also used in other fields of research.

Repast [98] is a suite of tools for building ABMs. Repast is used in a variety of fields, including economics, social science, high performance computing, and biology.

Soil [99] is a modeling platform that is built on top of the programming language Python. It provides a set of tools and libraries that are specifically designed for building agent-based models in the domain of environmental and ecological modeling. Soil is used to build and run simulations of complex systems in the natural world, such as ecosystems or climate models.

Hashkat [100] is an open-source ABM platform developed by the Argonne National Laboratory. It is designed to be used by scientists, engineers, and other researchers to build and run large-scale ABMs.

All of these tools are used to build and run simulations or models of complex systems. They are commonly used in research and education, and they all have a focus on agent-based modeling. *NetLogo* and *Repast* are more general-purpose tools, while *Soil* and *Hashkat* are more specialized for particular types of modeling.

2.3 Trust Management Systems

Jøsang et al. [101] define trust as the willingness of one party to rely on something or someone in a particular situation, despite the potential for negative consequences. Trust is often characterized by a feeling of relative security and can vary in extent depending on the specific circumstances. The risk, on the other hand, points to the situation where the result of an action is essential to an actor, but the result may not be positive. After examining the previous actions of a party, a party may decide to trust that party, incorporating the various risks involved in the decision. This is called a *trust decision* [102]. There exist different interpretations of trust, which we detail in the following.

Evidence or experience-based trust: This type of trust is based on the experience that the truster has had with the trustee [103]. It is founded on the belief that the trustee has consistently behaved in a trustworthy manner in the past and that this behavior is likely to continue in the future. This understanding of trust is often discussed in the context of interpersonal relationships, but it can also apply to trust between organizations or systems.

Application-specific behavior-based trust: In some cases, trust is related to specific behaviors that are relevant to the application or context in which it is being used. An example of this is conversational behavior on social media platforms, where the nature of the conversation between two individuals may influence the level of trust between them. Trust, in this sense, is often dynamic and can change over time as the behavior of the trustee changes [104].

Similarity-based trust: This type of trust is based on the idea that entities that are similar to one another are more likely to trust each other. This understanding of trust is often used in collaborative filtering systems, where the similarity of preferences or interests between users is used to predict which items they will like or trust [105, 106, 107, 108].

Reputation: Reputation is often considered to be a critical factor in facilitating trust. Reputation systems track the past behavior of individuals or organizations and use this information to assign them a reputation score. This score can then be used to predict the likelihood that the entity will behave in a trustworthy manner in the future (See Section 2.3.1 for Reputation Systems).

Fuzzy logic based trust: Fuzzy logic is a mathematical approach that allows for the representation of uncertainty in systems. In the context of trust, fuzzy logic has been used to model the uncertainty that can arise when determining the trustworthiness of an entity [109, 110, 111]. This approach allows for the incorporation of multiple factors that may influence trust and can produce more nuanced and realistic predictions of trust compared to binary or deterministic models.

Comprehensive trust: This understanding of trust incorporates a wide range of factors in order to calculate a trust value [112, 113]. This may include evidence or experience-based trust, application-specific behavior-based trust, similarity-based trust, and reputation, as well as other factors that may be relevant to the specific context in which trust is being evaluated. This approach is designed to provide a more comprehensive and realistic assessment of trust but may also be more complex and computationally intensive to implement.

2.3.1 Reputation Systems

The main aim of reputation systems is to create a systematic solution that regulates online interactions between people who do not know each other. According to the Oxford English Dictionary, reputation is “the opinion that people have about what somebody/something is like, based on what has happened in the past”. Therefore, it depends on other people’s views of someone or something. Mui et al. [114] define reputation as “the perception that an agent creates through past actions about its intentions and norms”.

2.3.1.1 Taxonomy

Based on the taxonomy provided in [43], reputation systems are classified into two main categories: *implicit* and *explicit*. Explicit reputation systems are categorized into further dimensions, which we will mention later.

The *implicit* category contains systems in which no direct reputation systems are implemented, yet it is possible to obtain a reputation as maintained by the users. Examples include Facebook and Linkedin, where users can infer the reputation of a given user using - for example - the number of connections of that user. A search engine like Google maintains the reputation of a web page using a ranking of that web page. Users can then argue about the reputation of that page relative to other pages.

The *explicit* category contains the systems which are implemented for the purpose of determining the trustworthiness of users in a system. The dimensions of explicit reputation systems are history, context, collection, representation, aggregation, entities, presence, governance, fabric, interoperability, control, evaluation, data filtering, and data aging. Below there are some summaries according to [43].

History is the stored set of interactions between users and the outcomes of these interactions. *Context* denotes the source of information used by the reputation system to calculate reputation. *Collection* represents the behavior of entities within the system. *Pepresentation* denotes how the reputation of the entity is stored. *Aggregation* denotes the method by which the reputation score is calculated. *Entities* denote the type of entities that can be people or resources. *Presence* means whether the availability of a given entity is required to calculate its reputation. *Governance* describes how the reputation system is managed. *Fabric* is how the reputation is organized. *Interoperability* means how accessible the reputation system is to the outside world, and it can either be open or closed. *Control* is the notion of how the system drives users to act in a certain way. *Evaluation* is the style of how the reputation system conveys the reputation information of an entity. *Data filtering* is whether the system provides the full or set of history to the requestors or note. *Data aging* describes whether there exists a mechanism to incorporate time into the system.

Reputation systems can also be classified by their origin, other than how the reputation system is constructed. To this end, we provide academic and commercial reputation systems.

Academic reputation systems

Regret [115] [116] is a reputation system for an electronic marketplace that considers three dimensions: individual, social, and ontological. The individual dimension consists of a user's interactions, such as the quality of a transaction and the frequency of overcharging or late delivery. The social dimension focuses on the user's social connections, such as the groups and communities they are associated with. The ontological reputation is based on a graph structure and is calculated by weighing and combining the aspects of the individual and social dimensions. The system also has a decay (forget) factor, which enables it to include only recent transactions.

Confidant [117] is a reputation system for ad hoc networks that aims to identify "bad" nodes. Nodes can monitor the behavior of their neighbors in terms of their ability to route, forward, and manipulate information according to the protocol. If a node identifies a "bad" node among its neighbors, it sends an alarm to other nodes to notify them of the bad node. The alarm is trusted based on the reputation of the sender, and if the sender's reputation is deemed sufficient, the alarm is passed on to other nodes. Node ratings are stored locally in lists that can be shared with other nodes. There is a time decay mechanism, meaning that a node's reputation can change from good to bad and vice versa over time.

Xrep [118] is a reputation system for resources and nodes in peer-to-peer networks. Each node maintains a history of resources and nodes, including a binary rating for resources and the number of successful and unsuccessful downloads for other nodes. A node can rate a resource based on ratings given by other peers, and the system allows for the rating of new resources by their originating nodes.

Eigentrust [119] is another peer-to-peer system that is used to keep the reputation of nodes. Each node maintains a history of its interactions with other nodes, including the sum of positive and negative interactions. When deciding on the global reputation of another node, a node receives information about the node from its neighbors and weighs the information based on the reputation of its neighbors, which is kept locally.

PGrid [120] is a peer-to-peer reputation system in which information is distributed among the nodes. The trustworthiness of a node is kept as a binary variable, and nodes can forward complaints about other nodes to their neighbors. When deciding on the trustworthiness of a node, a node receives complaints about the node from other nodes and uses a function to calculate the trustworthiness.

Peertrust [121] is another peer-to-peer reputation system that incorporates two new metrics: community context factor and transaction context factor. Again the reputation of a node is calculated using all the transactions and the feedback for these transactions, but now incorporating the community context factor.

Rateweb [122] is a system for the reputation of web services. In the system, each consumer keeps a history of interaction with each web service. When a consumer wants to calculate the reputation of a web service, it asks other consumers who have interacted with that service. Then, the reputation of the service is calculated by summing over the individual perceptions of consumers with regard to that service and averaging. A decay factor is also utilized to incorporate the time of interaction of other consumers; i.e., recent interactions with that service are weighted more.

R2Trust [123] is a reputation system for unstructured networks. It involves reputation and risk to calculate a trust value. The trust value of a node in the network is calculated using its relationships and the risk involved in these relationships. A node calculates the reputation value for another node using the local trust values of its peers. The system also uses a decay mechanism to calculate decayed trust values.

GRAft Generalized Recommendation Framework [124] is a reputation framework that involves the collection of reputation from online sources. The reputation score of an entity is maintained using profiles. In GRAft, unlike other reputation systems, requesters of information about an entity decide what to do with the information. There is no aggregation. They use policies over profiles to decide whether to trust an entity.

SocialTrust [125] is a reputation system for peer-to-peer file sharing that utilizes social network connections and conventional credit-based reputation systems to provide efficient reputation management. It allows nodes to favor friends for service transactions. The system also rewards and punishes nodes based on their number of friends and reputation. It aims to prevent certain attacks, such as denial of service and collusion. Simulation results show the effectiveness of SocialTrust in a comparison with Eigentrust.

Finally, the recent surveys in [126, 127] overview the research on blockchain-based reputation systems.

Commercial reputation systems

In *Amazon* [128], users can review the product they buy. Other users can then give feedback on those reviews as “helpful” and “not helpful.”

eBay [129] is an auction site. After a transaction, the buyer and the seller can rate each other as “Good,” “Neutral,” and “Negative”, and on other aspects such as accuracy, communication, shipping time, and charges.

Epinions is a product review website where users earn to become a top reviewer and popular author with good reviews. Users can create lists of trusted users [130]. The number of users that trust a user can be viewed in the user profile and can be regarded as the reputation score of that user.

Slashdot [131] incorporates a karma system where users with good karma can become moderators over time.

Stackoverflow is a question-answering site for software developers. Depending on the reputation of the user, which is obtained by various actions such as asking, responding, voting up and down, and commenting, the users are able to do specific tasks such as editing other people’s questions and answers and becoming moderators [132].

Turkopticon allows users of Amazon Mechanical Turk to know about the reputation of work providers in terms of communication, generosity, fairness, and promptness [133].

Reddit users maintain a reputation as karma. Users can gain karma by posting and commenting [134]. Karma acts like the amount of contribution to the site.

Uber and *Lyft* use a rating system on a five-star scale, to evaluate the performance of their drivers [135]. Drivers with high ratings are more likely to receive more ride requests. They also monitor drivers’ acceptance rates, cancellation rates, and other metrics to ensure that they are providing high-quality service to passengers. Drivers with low ratings may be deactivated from the platform.

Airbnb uses a reputation system based on reviews [136], to evaluate the performance of its hosts and guests. Both hosts and guests are able to leave reviews for each other after a stay. These reviews are visible on both profiles and can be used to help other users make decisions about whether or not to interact with them.

2.3.2 Attacks in Reputation Systems

Hoffman et al. [40] identify three aspects of a reputation system where vulnerabilities may occur: formulation, calculation, and dissemination.

The formulation aspect refers to how the system formulates a reputation metric from user input. This process involves collecting and evaluating data from various sources in order to calculate a reputation score for a particular user or entity. It is crucial for the formulation process to be transparent and fair, as any biases or inconsistencies in the data collection and evaluation could affect the accuracy of the reputation score.

The calculation aspect refers to how the reputation metric from the formulation process is actually calculated. This process may involve using algorithms or formulas to process and analyze the collected data in order to generate a reputation score. In some cases, such as in distributed systems, the calculation process may fail to achieve the metric described in the formulation process due to the complexity and decentralized nature of the system.

The dissemination aspect refers to how the calculated scores are conveyed to users or requesting parties. This process involves making the reputation scores accessible to those who need them, such as through a public database or a user's profile page. It is essential for the dissemination process to be transparent and accurate, as any errors or discrepancies in the presentation of the scores could lead to misunderstandings or misuse of the reputation system.

In this section, we list the types of attacks in trust management systems identified by various studies such as [40, 41, 42].

Self-promoting: Attackers falsely change their reputation to trick the system. This type of attack can be done by single or multiple organized entities. The attack can be performed due to a vulnerability in the calculation or dissemination processes of reputation scores. One such vulnerability occurs in systems where data integrity and authentication mechanisms do not exist.

One such attack can be performed by a single real-world entity acquiring multiple identities and using these to generate positive feedback on a single entity (see Sybil attack). This issue can arise due to a lack of a mechanism for proving the legitimacy of interactions. Some of the attackers may also be involved in legitimate transactions in addition to their malicious behavior, making them *moles* [137].

Mitigation techniques include accountability, proving the legitimacy of transactions, and preventing users from having multiple identities. Another mitigation technique would be to find attackers that cooperate with each other and form groups that generate positive feedback for themselves.

Slandering: Attackers falsely reduce the reputation of other nodes by reporting false information. This kind of attack can happen in systems where feedback authentication is exploitable. Systems that do not authenticate negative feedback or do not have a limiting capability of such or where negative feedback is more impactful in calculating the reputation score are susceptible to slandering attacks. There can be sophisticated attacks even when we allow only reputable entities to have negative feedback capability. This kind of entity does not exhibit malicious behavior most of the time but rather occasionally targets a single or a group of entities, using their good reputation.

Mitigation techniques involve suitable authentication mechanisms that maintain legitimate transactions. Systems that use indirect features to calculate reputation may be susceptible to attacks since each of these features may be exploited to manipulate the reputation score. For example, a system where the number of followers is used to calculate reputation can be deceived by having fake entities that are made to follow the adversary, increasing its reputation.

Whitewashing attack: This attack involves attackers willing to sacrifice their reputation to perform various actions and then use a vulnerability in the system to false restore their reputation score. This can be achieved, for example, by reentering the system with a new identity or using a vulnerability in the calculation method of reputation or using bots to increase own reputation.

Reputation systems where only a negative feedback mechanism is used are quite suitable for performing this type of attack because new users are not at a disadvantage. Another such vulnerability is not incorporating the freshness of interactions in the calculation of reputation. This enables potential attackers with an overall good reputation to perform short-lived attacks and get by undetected. These are called *traitors* by [138].

Defense techniques against this attack include having a reputation system where the freshness of interactions is included, new users do not have the same reputation as long-term users, and disallowing multiple identities.

Orchestrated attack: Attackers combine multiple strategies to perform attacks. There are multiple parties and groups involved. One such example is *oscillation attack* by [139], where groups of entities change their behavior over time. For example, one group of entities may be involved in increasing their reputation, whereas the other tries to exploit the bad reputation. The first group may be using self-promoting techniques and help the second group decrease their reputation faster. When the switch happens, the first group may use their well-built reputation to slander other victim entities until their good reputation becomes bad, and vice versa.

Mitigating orchestrated attacks is highly difficult since it requires dedicated effort to detect not one but at least two different strategies simultaneously. One approach would be to detect malicious strategies separately and come to a conclusion about an orchestrated attack later.

Denial of service: Attackers disable the system from calculating the reputation in the system. Mitigation techniques include redundancy and avoiding central mechanisms.

Misleading feedback attack: Misleading feedback attack, or badmouthing attack, which is similar to the slandering attack, aims to reduce the reputation of an entity or increase the reputation of an adversary.

Mitigating techniques include recommender trust-based mechanisms, detection-based mechanisms, and incentive-based mechanisms.

In recommender trust-based mechanisms, the recommendation from entities with high recommender trust value has more impactful propagations on the system [140].

Detection-based methods try to detect malicious entities using different techniques such as clustering [141, 142].

Incentive-based methods provide indirect ways to maintain good ratings through the encouragement of good transactions (for example, the seller selling good products, grouping of good buyers and sellers) [143].

Discrimination attack: In a setting of service providers and service users, service providers may choose to provide good service to some users and bad service to others. From the user's standpoint, users may give honest ratings, but the reputation of the service provider will be contradictory. There are no defense mechanisms for this type of attack, as reported in [41].

On-off attack: In this type of attack, malicious entities build a good reputation over time, then use this reputation to act maliciously without getting detected [140]. Defense mechanisms include a forgetting factor. However, if it is a long forgetting factor, recent behavior is not reflected, and if the forgetting is short, the malicious agent could build its reputation back again in a shorter period of time. Therefore determining this factor requires a deep understanding of the particular situation and setup.

Sybil attack: In Sybil attacks, multiple identities are forged to create the desired effect where it is reducing the reputation of a target entity or increasing one amplifying the effect with multiple fake entities. Defense mechanisms include preventing fake entities, increasing the legitimacy of transactions, and limiting them.

Newcomer attack: In this attack, a malicious entity may re-register as a new user and thus gets rid of its previous bad reputation [144]. Authenticating users and interactions are crucial to defending against this type of attack.

Value imbalance exploitation: In a service provider, service user setting, a service provider may bombard many high-quality services with low value and thus profit by the numbers [144].

Naive attack: This attack simply means the dishonest recommendations of an attacker without considering the inner workings of the reputation system. Systems, in general, should be resilient enough to mitigate this type of attack by not taking into account untrustworthy users' recommendations.

Traitor attack: In this attack, attackers build a reputation for a period of time and then act maliciously for a shorter time window. This is also called the on-off attack. Defense mechanisms from that also apply.

Collusion attack: Collusion attack can involve multiple entities, groups, and attack strategies, thus difficult to detect and defend against (see Orchestrated attack). The main idea revolves around attackers changing their behavior over time. Sun et al. [145] create a three-component based defense mechanism where they first detect a change in user behavior and correlate related users, and identify groups with malicious users.

2.4 Blockchain

After the invention of the infamous cryptocurrency Bitcoin in 2009, the technology behind it - "blockchain" - has received much attention because of its potential usage in other applications. The definition of Blockchain is simple; it consists of some blocks of data, and each block is linked to the previously created block. The whole linked blocks of data are called Blockchain.

2.4.1 Bitcoin

Bitcoin may have started a new era in the ways we define the economy and the digital world. While we enjoyed online shopping, we did that with credit cards that are built upon our regular currencies, such as Dollar and Euro. Visa is such a system that offers credit in dollars, for instance, to replace regular paper money. Paypal was another payment method where people deposited their dollars and transferred money between PayPal accounts. Bitcoin is a cryptographic method that tries to bring a transaction system that is distributed and kept collectively [146]. When a person transfers a certain amount of Bitcoin from one wallet to another, it gets validated across all the nodes, and it becomes immutable where one cannot change the amount of money.

2.4.2 The Mechanics of Blockchain

Although you can store any information in a block, it is required for each block to hold some essential information to be part of the blockchain. Therefore, each block consists of an index, data, previous hash value, a unique random number, and a generated hash [146].

Each block holds an index in the order of its creation time. The data it holds can be anything. For instance, in cryptocurrencies, data is transactions.

For example, user 1K3P-ABCD sends 0.005 Bitcoin to user 8KLM-EFGH in one transaction with hash 5AA2-3858 with a certain amount of fee [146].

Hashing is a way to map real data to some encrypted, much smaller data [147]. In the blockchain, the usage of hashing is for verification and chaining blocks; if you send data with its hash value, you can check whether the original information is changed by a malicious attack by recalculating the hash value from the real data and then comparing it to the received hash value. If you calculate a hash value from the received data and it is the same as the received hash value, then the received information is correct.

There are three essential aspects of hashing. First, the critical aspect of hashing is that you cannot reproduce the original data from a hash value, i.e., it is a *one-way* algorithm. Second, each single character change in the block changes the generated hash value arbitrarily. Third, the hash size is not affected by the length of the data. For instance, in SHA256 hashing [147], you can create a piece of information that has one alphabetic character, and it creates a hash value that has 64 characters, or you can create information with a million characters, and it still creates a hash value with the size of 64 characters.

In the blockchain, each block stores the hash value from the previous block, and it artificially creates chains and links each block to its previous block. Without the hash value from the previous block, your information will always be invalid because the hash value of the previous block is used to generate a hash for the current block. Because all blocks are connected to the previous, if data in the middle of the chain changes, the whole blockchain after the changed block becomes unusable because hash values become invalid [146]. Commonly, other computers in the system detect this issue and disapprove of this change, so the change becomes meaningless. Therefore, it is nearly impossible to cheat as long as more than 50 percent approve this change (which is sometimes called a *51% attack*).

All the information in the block is used for hashing, but there needs to be one more variable called *nonce* [146]. A nonce is a unique random number such that if you add a nonce to all data in the block, generated hash value starts with some unique set of characters, like 5 zero values. All hashes in all blocks start with the same pattern so that all computers in the system know whether the block is valid.

In any blockchain system, all computers connected to the network act as judges, and the information in the blocks are verified by the majority. Therefore as long as the majority is not malicious, their decision remains truthful. These computers are more commonly known as miners. More commonly, miners are known to generate money from thin air, but their most important role is to make blockchain work and make it safe from malicious attacks.

Miners' role is to find a random number *nonce*, and the easiest way to find it is to try each integer value starting from zero to infinite [146]. The first miner that finds this unique number creates a valid hash value that starts with a unique pattern, like five zeros. After a hash value is found, the miner sends it to every other computer in the network. If the hash value holds the data of the block, other computers approve adding the block to the blockchain.

Mining costs a lot of computational power and electricity. Therefore they need to be rewarded, or else it is unreasonable for them to verify transactions in the system voluntarily. At the end of each block, the fastest founder of the correct "nonce" number adds a transaction to its account. As of July 2019, its reward is 12.5 Bitcoin. The critical thing to notice is this transaction does not include the sender, and it is generated from thin air. This is the only possible way to generate a Bitcoin; other transactions must always be sent from someone. In other blockchain systems, other cryptocurrencies may be used.

2.4.3 Alternative Blockchains - Ethereum

There are thousands of cryptocurrencies, but only special ones have real value. For example, Ethereum is another type of cryptocurrency, and its particular property is its smart contracts [148].

A smart contract is a computer program that checks whether specific conditions are met. After that, it automatically makes the transaction. Other computers in the network act as witnesses. Smart contracts can be programmed by a computer language named Solidity [148] which is similar to JavaScript.

Smart contracts do not necessarily need to be used just for money transfers. Recently, a peace declaration between North and South Korea is recorded on an Ethereum block [149]. It can be used to store information of any kind. By uploading an agreement on an Ethereum block, the agreement is set out to be eternal.

2.4.4 Future of Blockchain

Blockchain technology has the potential to impact the way we conduct transactions in the future significantly. Its secure, transparent, and decentralized nature offers numerous advantages, such as the reduced risk of fraud, increased efficiency, and resistance to censorship and tampering. These benefits could lead to the disruption of traditional business models that rely on transaction fees, resulting in increased profits for sellers and reduced costs for buyers. However, the adoption of blockchain technology also comes with its own set of challenges and limitations. For example, implementing it may require significant changes to existing infrastructure and systems, and there still needs to be more understanding and standardization surrounding its use. Additionally, its decentralized and privacy-oriented nature may make it more challenging to regulate and reach a consensus on specific issues.

Despite these challenges, the potential opportunities for blockchain technology are vast. For example, it could improve supply chain management and logistics, enhance the security and transparency of voting systems, and be used in peer-to-peer applications such as ride-sharing, home rentals, and exchange of digital properties. Several companies have already begun using blockchain technology as their foundation, offering alternatives to traditional marketplaces, offering digital ownership via non-fungible tokens (NFTs), ride-sharing services, and music streaming platforms. Many industries, including finance, healthcare, and real estate, are also exploring the potential uses of blockchain technology and conducting pilot projects to test its feasibility. As more and more industries begin to adopt blockchain technology, it has the potential to powerfully shape the way we do business in the future.

Chapter 3

Crowd: a Social Network Simulation Framework

3.1 Introduction

In our work, we approach the problem of modeling, simulating, and analyzing social networks from a novel perspective. We deliver the notion of "network as code" inspired by "infrastructure as code", a term from cloud computing where one can define infrastructure using configuration files and code. "Network as code" defines the social network, agents, actions, interactions, parameters, and how it will be simulated, analyzed, and visualized using configuration files and code. This approach enables researchers to quickly model the given social network along with the flexibility of scripting the entire behavior in the network (see the comparison with existing tools in Table 3.1).

In this chapter, we first explain the design of *Crowd*, a social network simulation framework. We then explain how it can be used. We finally share various social network analysis scenarios that can be achieved with the framework.

Table 3.1: Comparison with existing tools

Tool	Focus	Analysis	Agent	Configuration Style	Processing	Software Type	Code Language
NetLogo [97]	Broad Agent-Based	Predefined and user-defined code	Predefined and user-defined code	Model-based	Synchronous	Tool	-
Repast [98]	Broad Agent-Based	-	-	-	Synchronous	Tool	Java
Soil [99]	Social Networks	Predefined	Predefined	Conf File	Synchronous	Tool	Python
Hashkat [100]	Social Networks (Twitter)	Predefined	Conf File	Predefined Twitter based	Synchronous	Tool	C++
Crowd	Broad Agent-Based	Predefined and user-defined code	Predefined and user-defined code	Network as code	Synchronous (current)	Library	Python

3.2 Architecture

We developed Crowd as an extensible framework in terms of preprocessing, visualizing, and running various networks with nodes and actions. The overall architecture is given in Figure 3.1. We describe each part in the following and give a simplified version of how a simulation is run utilizing each of these parts in Algorithm 1.

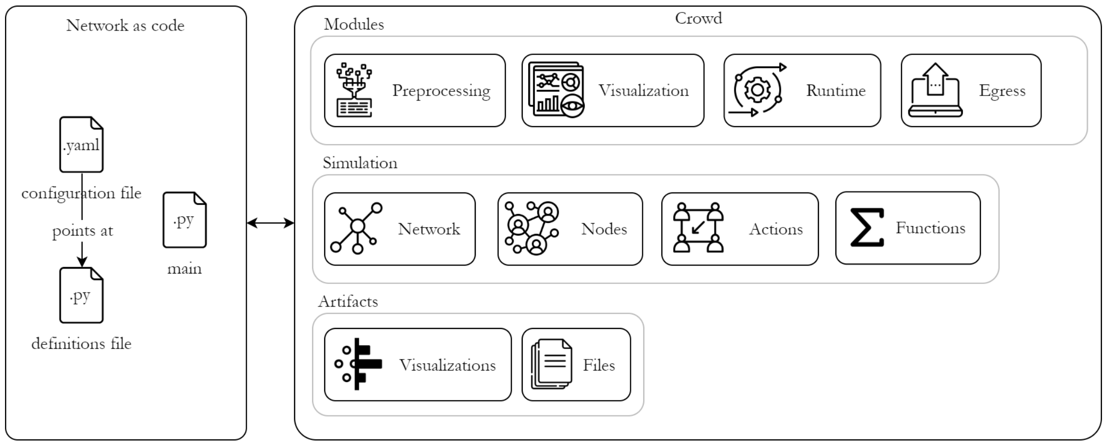


Figure 3.1: Architecture of Crowd[1]

[1]Icons are from flaticon.com

Algorithm 1 Simulation Run Algorithm

```
Require: $epochs$: number of epochs to run
Require: $visualizers$: list of visualizers to use
Require: $snapshot\_period$: number of epochs between snapshots
Require: $agility$: agility parameter
Require: $egress$: egress object
Require: $G$: Graph
function RUN($epochs$, $visualizers$, $snapshot\_period$, $agility$, $digress$)
    $count \leftarrow conf.count$
    if $agility = 0$ then
        $selection\_count \leftarrow 1$
    else
        $selection\_count \leftarrow count * agility$
    end if
    for $epoch \in [0, epochs)$ do
        $selected\_nodes\_for\_action \leftarrow$ SELECT_NODES_FOR_ACTION($selection\_count$)
        for $nodenum \in selected\_nodes\_for\_action$ do
            $actions \leftarrow G.nodes[nodenum]$. SELECT_ACTIONS
            for $action \in actions$ do
                EXECUTE_ACTION($nodenum, G.nodes[nodenum], action$)
            end for
        end for
        if ($epoch$ mod $snapshot\_period = 0$) or ($epoch = epochs - 1$) then
            if $visualizers \neq None$ then
                for $visualizer \in visualizers$ do
                    VISUALIZER.DRAW($epoch$)
                end for
            end if
            if $digress \neq None$ then
                EGRESS.SAVE
            end if
        end if
    end for
    if $visualizers \neq None$ then
        for $visualizer \in visualizers$ do
            VISUALIZER.ANIMATE
        end for
    end if
end function
```

Configuration A network can be defined in a YAML file. The file basically contains the definition of the network. This may include the types of nodes, types of node actions, types of network actions, functions, information, and network structure. In addition, this file points to a definition file that contains the code on how this network is going to be run and how nodes will interact within the network.

User code defines the logic behind the network. Developers can define the capabilities of the nodes and what they do according to the specific network they design.

The library currently supports parametric networks such as random, Barabasi-Albert, Watts-Strogatz (i.e., types of networks supported by networkx. See Section 3.3. Dependencies) and from a file. In addition, the library has a mechanism to check the structural correctness of the configuration file as some of the options and nested-options are optional. This mechanism also makes sure that the definitions file contains the functionality described in the configuration file.

Preprocessing Crowd enables users to give various preprocessing options to run before the simulations. The preprocessing can be defined by the user by extending the Preprocessing interface in the library. There exists a built-in community detection algorithm based on Louvain's method.

Visualization Visualization is important for any simulation framework. Visualization in Crowd takes the values in each epoch resulting from the functions defined by the user in the configuration file as statistical functions. Currently, there exists a basic visualizer that draws the state of the network in association with the custom node types defined in the configuration file. Furthermore, the statistical visualizer draws various charts using the values defined by the statistical functions. These visualizers can also generate an animated GIF combining all epochs if prompted. Finally, an HTML visualizer draws using the D3.js Javascript framework.

3.3 Dependencies

Crowd depends on a set of well-known python libraries. As most of these libraries continue to be developed, the functionalities of Crowd are also expected to increase. We list the most important dependencies below.

- `Networkx` is a Python library for working with complex networks. It provides tools for creating, manipulating, and analyzing networks, as well as for reading and writing network data to and from files. NetworkX is useful for analyzing and modeling large-scale complex systems that can be represented as networks. It is used in various fields, including computer science, physics, biology, and social sciences.

  NetworkX provides many tools for working with complex networks, including functions for creating and adding nodes and edges to a network, calculating various graph-theoretic properties (such as degree centrality or betweenness centrality), and visualizing networks using various layout algorithms. It also supports reading and writing graph data from and to various file formats, such as GraphML, GEXF, and GML.

- `ndlib` is a Python library for analyzing complex networks. It is built on top of NetworkX. ndlib provides several tools for analyzing and modeling the dynamics of complex systems represented as networks. This includes functions for computing network structure and topology measures, such as centrality measures and community detection algorithms. It also includes several tools for simulating the spread of epidemics, information, or other phenomena on networks, using models such as SIR (Susceptible-Infected-Removed) or SI (Susceptible-Infected).

- `imageio` is a Python library for reading and writing a wide range of image, video, and audio file formats. It provides a simple, easy-to-use interface for reading and writing files and a number of advanced features for handling more complex cases.

imageio supports a wide range of file formats, including popular image formats such as JPEG, PNG, and GIF, as well as less common formats like DICOM (a format used for medical images) and HDF5 (a format for storing large, complex datasets). It can also read and write video and audio files in various formats, including MP4, AVI, and WAV.

imageio is designed to be easy to use and flexible, making it suitable for a wide range of applications. It is often used for tasks such as reading and writing image and video files for data analysis and machine learning, as well as for more general-purpose file I/O tasks.

- `d3.js` (Data-Driven Documents) is a JavaScript library for creating interactive visualizations for the web. It is particularly well-suited for structured data visualization and is widely used for creating charts, graphs, and other types of visualizations driven by data.

  d3.js is built on web standards such as HTML, CSS, and SVG, making it easy to integrate with web-based projects and workflows and providing many features and capabilities for creating dynamic, interactive visualizations, including support for animations, transitions, and data binding.

- `matplotlib` is a data visualization and plotting library for Python. It allows users to create a variety of static, animated, and interactive visualizations and is particularly well-suited for visualizing large, complex datasets. It is widely used in data analysis, scientific computing, and machine learning. It is known for its ability to create a wide range of visualizations and easy customization.

3.4 Case Studies

This section describes how various scenarios can be implemented using Crowd. The first scenario includes how we used the framework for a part of our work in Chapter 4. The second one is based on a simulation scenario of Covid-19 pandemic data.

3.4.1 Game Theory Network

In this scenario defined in "`gtnetwork.yaml`" (shown in Listing 1), we create various types of 1000 nodes called Cooperator, Defector, and Neutral, each assigned with a specific color on a random regular network with a degree of 4, with variables w, b, c, community detection applied to Cooperator and Defector type of nodes. At each iteration, a function "`blue_red_ratio`" is called, and nodes can "`update`" their type with the given "`nodetype`." The configuration file points at the definition file of "`gtnetwork.py`" given in Listing 2, which implements what the network and nodes will do in the simulation. These files are then used in Listing 3 to perform the simulation. Basic and Stats Visualizers are used to visualize the epochs. Various results of simulations similar to this can be found in Chapter 4.

Listing 1 `gtnetwork.yaml`: Configuration File to describe properties such as definition, information, node types, structure, preprocessing, function, statistical functions, and node actions

```yaml
name : gtnetwork
definitions : gtnetwork
info:
    total_count : 1000
    w : 0.01
    b : 25
    c : 2
node types:
    Cooperator:
        color : blue
    Defector:
        color : red
    Neutral:
        color : grey

structure :
    random:
        degree: 4
preprocessing:
    communitydetection:
        - Cooperator
        - Defector
functions:
    - get_total_count
    - blue_red_ratio
statfunctions:
    - blue_red_ratio
node_actions:
    - update:
        -args:
            - nodetype

```

Listing 2 `gtnetwork.py`: User-written code to describe the network, nodes and actions

```
from crowd import node as n
import random

class gtnetwork:

    def get_total_count():
        ...
        return count

    def blue_red_ratio(network):
        ...
        return all_blue_count/all_red_count

    def calculate_fitness(network, node):
        ...
        return fitness

    def get_neighbors_and_color_counts(network, node):
        ...

    def randomly_choose_precise(network):
        ...

    def update(node, *argv):
       ..
    def changetype(node, *argv):
        ...

        return
    def select_nodes(network):
        ..
        return nodes

class Cooperator(n.Node):
    def select_actions(self, actions):
        ...
        return selected_actions

class Defector(n.Node):
    def select_actions(self, actions):
        ...
        return selected_actions

class Neutral(n.Node):
    def select_actions(self, actions):
        ...
        return selected_actions
```

Listing 3 Network visualization test code

```
from crowd import network as n
from crowd.visualization import basic as bv
from crowd.visualization import statsvisualizer as sv

mynetwork = n.Network("gtnetwork.yaml")
visualizer = bv.Basic("artifacts")
visualizer2 = sv.StatsVisualizer("artifacts")
mynetwork.run(30000, [visualizer,visualizer2], agility=0)
```

3.4.2 Covid-19

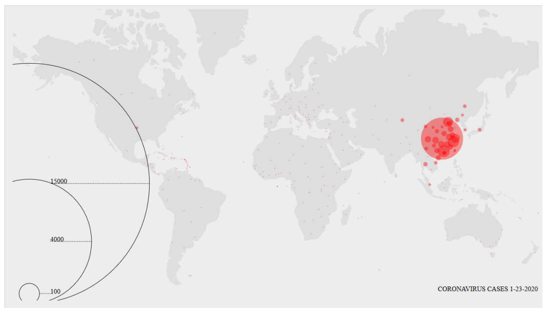
In this scenario, we use existing data on Covid-19 to demonstrate the minimal usage of Crowd without writing any code for the health domain. The configuration file in Listing 4 uses the “`network.tsv`” file, which is a “`nodes only`” network consisting of world countries, ignoring its header. In Listing 5, we use the configuration file to generate the network and run it against “`states.tsv`” which contains daily Covid-19 cases per node (country, in this case) in “`network.tsv`” file. We use a bubble map visualizer that draws the states of the nodes across runs. The results are given in Figure 3.2.

Listing 4 `coronavirus.yaml`: Coronavirus configuration file

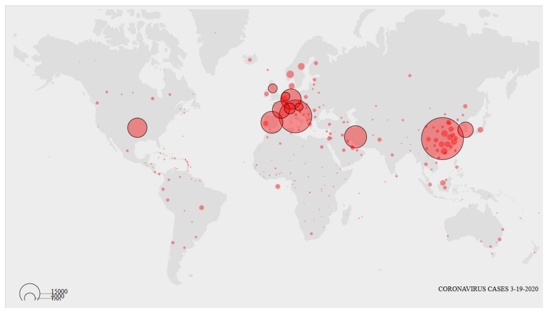
```
name : coronavirus
info:
    total_count: 1000
structure:
  file:
    type: nodes_only
    path: network.tsv
    header: true
```

Listing 5 Coronavirus visualization test code

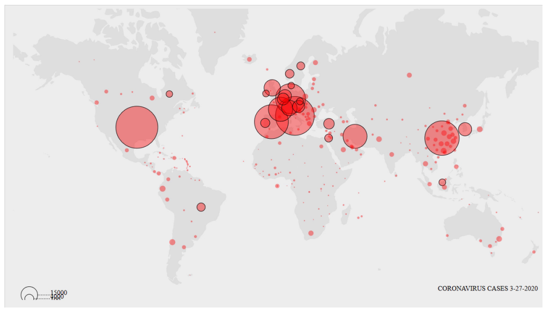
```
from crowd import network as n
from crowd.visualization import bubblemapvisualizer as bmv

def test():

    mynetwork = n.Network("coronavirus.yaml")
    visualizer = bmv.BubbleMapVisualizer("artifacts")
    mynetwork.run_states("states.tsv", [visualizer])

test()
```



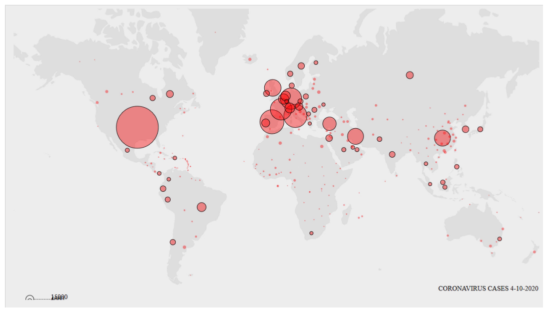
(a) Daily cases on 23 February 2020



(b) Daily cases on 19 March 2020



(c) Daily cases on 27 March 2020



(d) Daily cases on 10 April 2020

Figure 3.2: Covid-19 spread visualized using Crowd

3.5 Roadmap

In this chapter, we approached the problem of modeling, simulating, and analyzing social networks from a novel perspective. We delivered the notion of "network as code" inspired by "infrastructure as code," a term from cloud computing where one could define infrastructure using configuration files and code. "Network as code" defined the social network, agents, actions, interactions, parameters, and how it would be simulated, analyzed, and visualized using configuration files and code. This approach enables researchers to quickly model the given social network along with the flexibility of scripting the entire behavior in the network.

Our initial goal includes making Crowd open-source so that it could be used in the field and contributions such as feedback and code could be received. As we described in the comparison of various tools in the field, all the tools utilized a synchronous approach in running simulations. To diversify this and increase simulation performance, we intend to implement multi-threading and GPU support.

We also intend to add generation capabilities for more types of networks that imitate the network structure of existing networks. Finally, we plan to add more visualizations by directly integrating d3.js and other visualization libraries.

Chapter 4

Game Theoretic and Reinforcement Learning Approaches

4.1 Introduction

In this chapter, we explain our methodology on the gamification of misinformation propagation in online social networks. This chapter is based on our work in [150]. Our contributions are as follows:

- We explore the misinformation propagation problem as a cooperative game on graphs and construct the payoff matrix where we define the benefit and costs associated with certain actions of the players (the nodes).
- We run various simulations on random regular networks and a real network based on Facebook to explore how the game evolves concerning the benefit and cost ratio.

- We define another game at the network level where there are agents that try to maximize the number of nodes affected on their behalf during their misinformation and counter-misinformation campaigns.
- We propose a deep reinforcement learning-based method for the selections of the actions of the agent in the network level game, perform experiments and compare the results with other node selection techniques.

Section 4.2 describes how the misinformation game can be modeled at the node level. Section 4.3 gives the details of the network level game. In Section 4.4, we discuss our findings and in Section 4.5, we summarize this chapter along with future directions.

4.2 Misinformation Game at the Node Level

4.2.1 Evolution of Cooperation on Graphs

Prisoner's Dilemma is a well-known game in the field of game theory with two accomplices presented with two strategies: Cooperate (Silence) and Defect (Betrayal) [36]. A typical payoff matrix for the cooperation game inspired by the Prisoner's Dilemma is shown in Table 4.1. In this scenario, although the mutual cooperation $q = (-1, -1)$ is the most favorable overall outcome, the individually best choices force the players to defect: $t = (-2, -2)$.

Cooperation within communities has also been studied [151], including the iterated form of the game in a networked scenario to identify the evolutionary properties of cooperation [152]. In such a game, people are nodes on a graph and connected with edges representing a contextual relationship. A cooperator emits a benefit b to all its neighbors at the cost of c. A defector does not emit any benefit but benefits from the cooperators it neighbors. The payoff matrix of such a game is given in Table 4.2.

Table 4.1: The payoff matrix of Prisoners' Dilemma

	Cooperate	Defect
Cooperate	$q = (-1, -1)$	$r = (-3, 0)$
Defect	$s = (0, -3)$	$t = (-2, -2)$

Table 4.2: The payoff matrix of the cooperation game

	Cooperate	Defect
Cooperate	$(b - c)$	$(-c)$
Defect	(b)	(0)

In this networked setting, with benefit and cost presented as the game parameters, the evolution mechanism is layout through updating nodes with respect to certain mechanisms. At a time instant, a node is randomly chosen to be updated. There are three such update strategies [38].

1. Death-birth updating strategy where a node is chosen to die and cooperators and defectors compete over it without considering its original fitness.
2. Birth-death updating where a node is chosen for reproduction and its child replaces one of the neighbors.
3. Imitation updating where a node is chosen to replace its strategy with respect to its neighbors. Its fitness is taken into account.

We will specifically visit death-birth updating and imitation updating strategies since these two can provide intuitive applications in today's online social networks: Death-birth updating can simulate setting the strategy of a node only by its neighboring strategies, while imitation updating can provide an analogy for the update of a node while also considering its own fitness in the final decision.

Here, we will describe the strategies using the payoffs in Table 4.1. Under death-birth updating, let us assume that a node is to be updated. Let k represent the number of neighbors of the node, and a C player is a cooperator, and a D is a defector. In this sense, k_C represents the number of C neighbors whereas

k_D represents the number of D neighbors with $k = k_C + k_D$. Then, the node's strategy is set as C based on the probability:

$$\frac{k_C f_C}{k_C f_C + k_D f_D} \tag{4.1}$$

where $k_C f_C$ and $k_D f_D$ denote the total fitness of the neighboring nodes with C and D strategies, respectively. The fitness of a C player and a D player is described in the following equations if the selected node was a D player:

$$f_C = 1 - w + w[[(k-1)p_{C|C}]q + [(k-1)p_{D|C} + 1]r] \tag{4.2}$$

$$f_D = 1 - w + w[[(k-1)p_{C|D}]s + [(k-1)p_{D|D} + 1]t] \tag{4.3}$$

In these two equations, we describe the fitness of a node in terms of its neighbors. $p_{C|C}$ is the probability of finding a C node as a neighbor of a C node, whereas $p_{D|C}$ is the probability of finding a D node as a neighbor of a C node. $(k-1)p_{C|C}q$ represents the total contribution from C neighbors. Calculating these probabilities is relatively cost-effective in large graphs. w is the selection parameter that provides a linear combination within the game dynamics. We talk about a strong selection when w is close to 1; a weak selection when it is very small.

For imitation updating, we need to describe the fitness of a C node f_0 that is about to be updated.

$$f_0 = 1 - w + w(k_C s + k_D t) \tag{4.4}$$

The node chooses strategy D with respect to the following probability:

$$\frac{k_D f_D}{k_C f_C + k_D f_D + f_0} \tag{4.5}$$

According to the game plan where payoffs are set as in 4.2, cooperators are favoured if $b/c > k + 2$ under imitation updating and $b/c > k$ under death-birth updating [38].

4.2.2 Information Propagation as an Evolutionary Game on Graphs

We visit similar games from the literature towards an analogy for the misinformation game. "Closed-bag exchange" is a game where two people exchange bags containing money and goods. In the game, players either honor the deal (cooperate) or deliver an empty bag (defect). "Peace war game" is another example where making peace (cooperate) is mutually beneficial; the one-sided "war" (defect) strategy brings more benefit in the game. Finally, the lying game has been modeled in numerous research [153]. Strategizing for the interest of the individual - as evident in these games - may lead to a notion called the tragedy of the commons for shared resources. The problem has been translated to the digital world as the tragedy of the digital commons, and the lack of regulatory systems causes pollution in digital resources also associated with misinformation [154, 155].

In an online social network, some nodes may be inclined or even work for misinforming their proximity (i.e., their followers, connections, friends) on purpose. This deliberate version of misinformation is called disinformation, and the nodes are responsible for their actions and thus can be included in a game-theoretic environment. After being misinformed, previously indifferent nodes can relay this information to other nodes, making them a part of the game. There may exist other nodes that work on the opposite side of the disinformers. These nodes have to work harder than the latter since it is harder to convince the other nodes about the truth, while false information is generally more catchy and sticky, or interesting. The research confirms that false information spreads faster [22, 156].

First, let us define an information exchange game between two parties. Assume there are two strategies: cooperation and defection, where cooperation is sending correct information while defection is sending a false one. A player receives benefit b if the other player chooses to share correct information and there is a cost of correct information to the sender, while false information provides no benefit to the receiver. The payoffs are defined exactly like the cooperation game defined in Table 4.2. The Nash equilibrium of this game is with the outcome (0,0) where

Table 4.3: The payoff matrix of the misinformation game

	Believe	Don't Believe
Correct	$(b_1 - c_1, b_2) \approx (b - c, b - c)$	$(-c_1, b_3) \approx (-c, b)$
False	$(b_4 - c_2, -c_3) \approx (b, -c)$	$(-c_2, 0) \approx (0, 0)$

both players choose to disinform. However, there was a better outcome for them $(b - c, b - c)$ if they could both choose to relay the correct information. This game can also be intuitively connected to the famous closed-bag exchange game. If the game is set up around this description, the studies about the evolution of cooperation on graphs can be easily applied and the findings are expected to be in parallel. Although this type of setting for the game enables observations on the adaptation of cooperative or defective strategies over the population, it does not closely simulate the properties of propagation through iterative rounds of games over time. Hence, further modifications are required.

In this modified game, the possible strategies for Player 2 are different. Player 1, again, shares either correct information or a false one while Player 2 either accepts the information as correct, i.e., believes it, or does not accept it. Since this is an iterated game, Player 2 can then become a spreader in the later rounds. In this new setting, it is possible to define fine-grained values for benefit and cost for each player. The utility of the correct information to the sender is b_1 with a cost of c_1. The utility of receiving correct information to the receiver is b_2 if the receiver believes and b_3 if the receiver does not believe. That is because there is an intrinsic value in the correct information. However, the sender does not receive any benefit for the latter. The utility of the false information to the sender is b_4 if the receiver believes it, and 0 otherwise. The cost of false information is c_2. Believing false information has a cost of c_3. Although the information exchange game we previously introduced had the same payoff matrix, we need to introduce some assumptions for the misinformation game in its current form. First, we assume c_1 and c_3 as equal and denote it as c. We assume that $c_2 << c_1$ and disregard c_2. We also take all benefits values as equal, except for b_2. We think that it is upper-bounded by $b - c$ since the utility of a receiver cannot be larger than the sender's if the sent information is correct. The simplified payoff matrix is given in Table 3. The Nash equilibrium of this simplified game is

(0,0), to disseminate “False” information for Player 1 and “Do not Believe” the information for Player 2. However, the scenario of sending “Correct” information and “Believing” provides a better and mutually beneficial outcome for players 1 and 2, respectively. Hence, the resulting non-zero-sum game displays the same characteristics of the cooperative games and the Prisoner’s Dilemma in particular, under the previously listed assumptions on the benefit and cost values.

In addition, while the Nash equilibrium provides a general solution for games in the traditional setup, we may need other measures of evolutionary dominance under evolutionary settings. Evolutionarily Stable Strategy (ESS) [157] is a modification of the Nash equilibrium, which states that a strategy is said to be evolutionarily stable if adopted by a population in an evolutionary environment, and cannot be replaced by another strategy. Given $E(I, J)$ as the payoff of selecting strategy I against T, for the strategy I to be an ESS, two conditions should be considered [157]:

1. $E(I, I) > E(J, I)$ or
2. $E(I, I) = E(J, I)$ and $E(I, J) > E(J, J)$

According to this definition, the defection strategy is evolutionarily stable in the designed misinformation game. However, it has been shown that the evolution of cooperation is possible in the case of $b/c > k$ [38] and small k or large w are the two factors that affect the outcome in favor of cooperation in the iterated cooperation game on graphs [158].

A combined strategy for the misinformation game

Algorithm 2 Node Update Algorithm

```
Require: G: Graph
Require: B: Set of Cooperator Nodes (Blue)
Require: R: Set of Defector Nodes (Red)
Require: Gr: Set of Neutral Nodes (Gray)
 1: procedure UpdateNode($G$,$B$, $R$, $Gr$)
 2:     $S_B \leftarrow 0$                                        ▷ Blue Fitness Sum
 3:     $S_R \leftarrow 0$                                        ▷ Red Fitness Sum
 4:     $v, V \leftarrow$ randomlychoose($G, B, R, Gr$)
 5:     for $\hat{v} \in V$ do
 6:         $f_{\hat{v}} \leftarrow$ calculatefitness($G, \hat{v}, B, R$)
 7:         if $\hat{v} \in B$ then
 8:             $S_B \leftarrow S_B + f_{\hat{v}}$
 9:         else if $\hat{v} \in R$ then
10:             $S_R \leftarrow S_R + f_{\hat{v}}$
11:         end if
12:     end for
13:     $f_v \leftarrow$ calculatefitness($G, v, B, R$)
14:     if $v \in B$ then
15:         $S_B \leftarrow S_B + f_v$
16:     else if $v \in R$ then
17:         $S_R \leftarrow S_R + f_v$
18:     end if
19:     if $S_B > S_R$ then
20:         $B \leftarrow B \cup v$
21:     else if $S_B < S_R$ then
22:         $R \leftarrow R \cup v$
23:     end if
24:     return $v$
25: end procedure
```

When the misinformation game is played out on the network, there exist three types of actors: Defectors, Cooperators, and Neutral nodes. In a social network, these correspond to the misinformers, correctors, and neutral nodes (red, blue, and grey nodes, respectively). In epidemiology as well as information diffusion theory, there exist two main types of notions that describe the state of nodes in a network: Susceptible, Infected and Recovered Model (SIR) [159] and Susceptible, Infected, Susceptible (SIS) Model. In the SIR model, a node can be susceptible, infected, or recovered without ever getting infected again. In the SIS model, however, a node can be reinfected. With this analogy, a susceptible, hence neutral node, can be infected or misinformed and become a spreader. It can be recovered with correct information to become a corrector. In this work, we chose the SIS model; thus, it is possible that a node can change its state from a grey node to a red or blue node, and a blue or red node can invert its position to become a red or blue node. To reflect this strategy, we need to accommodate two different strategies: one for grey nodes to select a new position and one for nodes with an existing stance to change their type. The first corresponds to a death-birth updating strategy where the fitness of the node to be updated is not taken into consideration. The latter is when a node updates its type based on its neighbors and its own fitness.

We adopt the updating algorithm given as Algorithm 1. This algorithm denotes a mix of death-birth and imitation updating strategies. The algorithm is described as follows. During the simulation, a node v is randomly selected to be updated, along with its set of neighbors V. For each neighbor $\hat{v} \in V$, a fitness value $f_{\hat{v}}$ is calculated and it is added to a cumulative sum; S_B for cooperators (Blue) or S_R for defectors (Red), according to the strategy (Blue or Red) of $\hat{v}$. Then, the fitness of the selected node f_v is calculated. After the addition of f_v to the cumulative sum of its original strategy, the strategy of v is updated with strategy B or R with the larger cumulative sum (S_B or S_R).

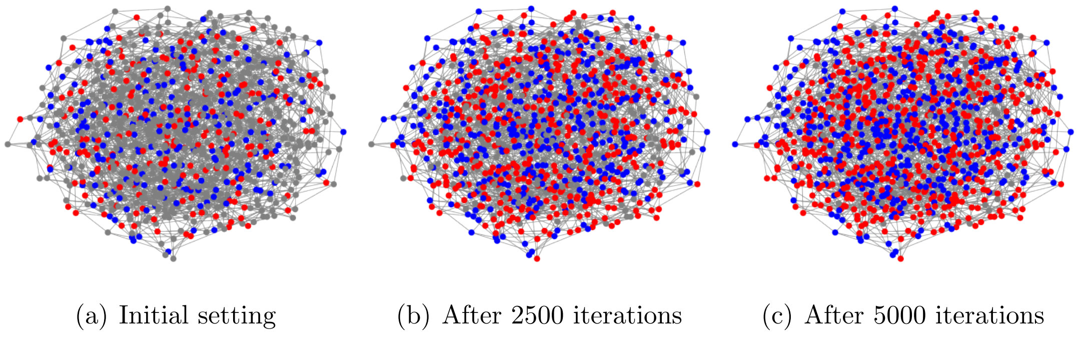

(a) Initial setting (b) After 2500 iterations (c) After 5000 iterations

Figure 4.1: Random network experiment 1: $k = 4$ and $b/c = 2$

4.2.3 Simulations

In this section, to observe whether the misinformation game described in this work is similar to the cooperation game, we run various simulations. The particular point we are after is the inequality of $b/c > k$. We want to see whether the graph is to be dominated by misinformation when the inequality fails, and correct information holds when the inequality is met.

4.2.3.1 Random Regular Networks

To test with the changing number of average neighbors, we choose to experiment with random regular graphs. Furthermore, we apply a community detection algorithm to create groups of nodes. These will serve as the set of competing nodes over unassigned nodes. In all our experiments, red nodes describe misinformers, and blue nodes describe correctors. Gray nodes are not assigned. We also pay attention to the sizes of the blue and red groups. We do not want one group to have a larger upstart advantage over the other.

In the first experiment, k is chosen as 4 and b/c is 2. Figure 4.1 shows the state of the network, and we see that the misinformer strategy increases its population over the cooperator strategy. In the second experiment, k is chosen as 4 and b/c is 20. We see that blue nodes end up with a slightly larger population than the red nodes (Figure 4.2).

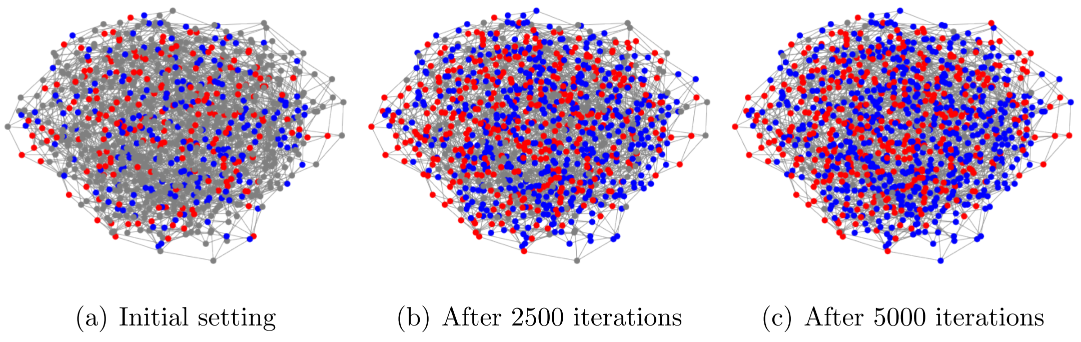

(a) Initial setting (b) After 2500 iterations (c) After 5000 iterations

Figure 4.2: Random network experiment 2: $k = 4$ and $b/c = 20$

Figure 4.3 shows the change of blue/red ratio over time for different values of benefit and cost when $k = 4$. According to the experiments, the dominance of cooperators is possible if $b/c > k$.

4.2.3.2 Facebook Network

We also run simulations on a real data set based on a set of Facebook users published in [160] (around 4000 nodes, 80000 edges). Similarly, we first choose 2 communities and label these as Blue or Red. Then we run the same algorithm. In this graph $k = 43$. In the first experiment with this data, we choose $b/c = 5$. The results in Figure 4.4 show that there is red dominance.

In the second experiment, we exaggerate the ratio of b/c to see its effect. Figure 4.5 shows that the blue strategy dominates the red strategy with this setting.

We repeat the experiment for various b/c combinations, and again, according to our observations, as shown in Figure 4.6, the cooperators are favored if $b/c > k$.

4.2.3.3 Strategy Dominance Probabilities

In evolution and evolutionary game theory, as well as in [38], the term fixation corresponds to the state of a network where a network is completely covered in

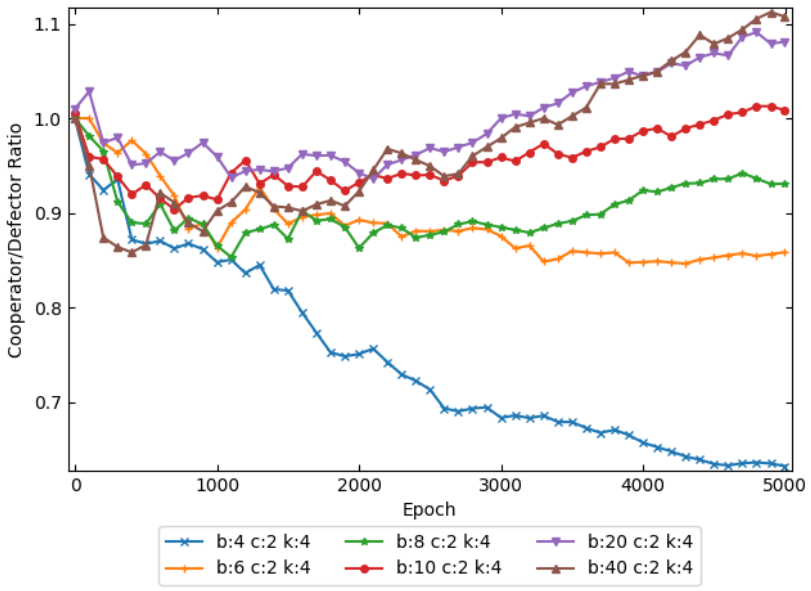


Figure 4.3: Change of blue/red ratio during the simulation of random network experiment with various benefit and cost values

one of the types of nodes. In our simulations, at this moment of our research, we use the "win" probability, which we obtain by the ratio of blue wins over red over a finite number of iterations. In Figure 4.7(a), we show that winning probability increases as b/c increases. The graph shows data for small (N=100), medium (N=1000), and large (N=10000) networks, each with 5N epochs and 50 iterations. k was chosen as 4.

4.2.3.4 The Effect of Initial Node Distribution on the Dominance

In our previous simulations, the initial network setting was the random distribution of nodes. However, in real-world scenarios, this may not be the case. In Figure 4.7(b), we start the network after calculating two same-sized clusters for opposing sides using community detection where most of the network is neutral. We use the Louvain method [161] for community detection for its wide acceptance and accessibility. However, more recent methods such as [162] and [163] could also be used. Our initial results show that the win probability is dramatically increased in this network structure.

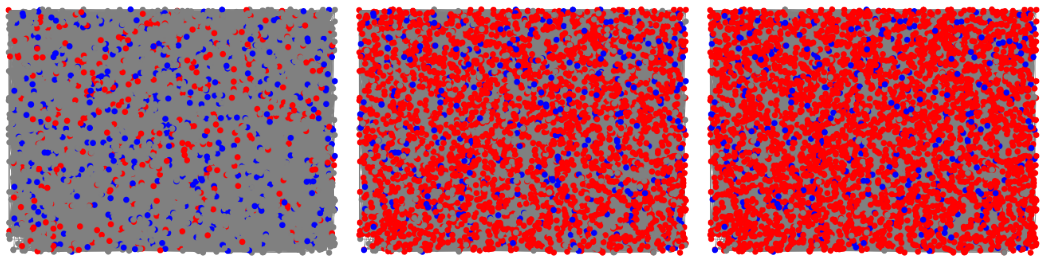

(a) Initial setting (b) After 10000 iterations (c) After 20000 iterations

Figure 4.4: Facebook experiment: $k = 43$ and $b/c = 5$

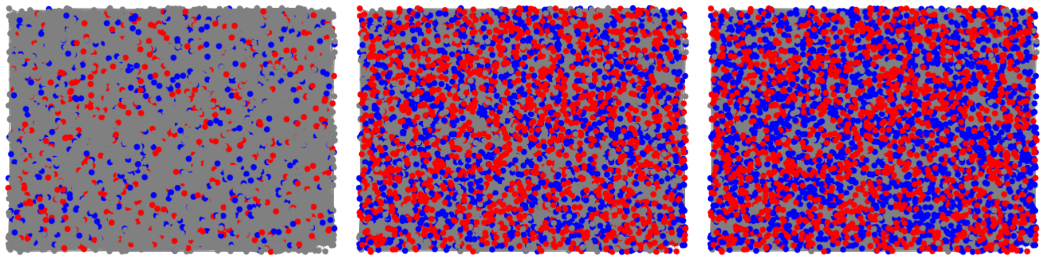

(a) Initial setting (b) After 10000 iterations (c) After 20000 iterations

Figure 4.5: Facebook experiment $k = 43$ and $b/c = 70$

In the community setting, the connectivity between the clusters is low (local k is low) at the very beginning. Since red nodes require blue nodes to benefit, initially, the spread rate of red nodes is low; only when the connectivity between red nodes and blue nodes is high (local k is high) then the red nodes are advantageous.

4.2.4 Limitations of a Node-level Game

Studying the misinformation game where the players are the nodes within a social network may enable a better theoretical understanding of how nodes change their strategies under an evolutionary setting. On the other hand, establishing such a game in contemporary social networks on the web in a holistic manner is difficult. This is because the users are actual people with different aspirations

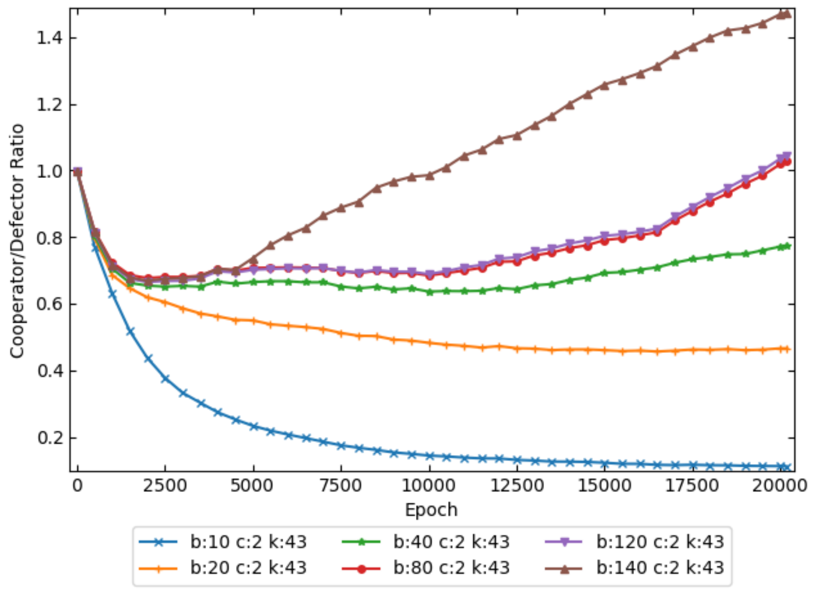
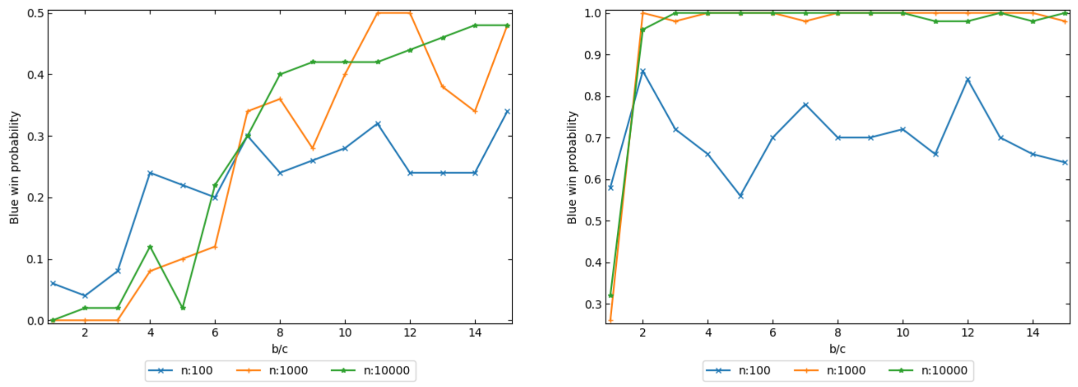


Figure 4.6: Change of blue/red ratio during the simulation of Facebook network with various benefit and cost values

and have different motives for using such networks, and they are exchanging and influenced by different information simultaneously. There may be individual benefit and cost values for each interaction rather than static network-wide values for them. In addition, determining benefit and cost as discrete variables that could simulate or offer analogies for the real-life benefits and costs of the said misinformation mechanism is also difficult; hence as was in our study, it leads to making assumptions.

However, it may be beneficial to list some of the concepts we considered for the values of benefit and cost while doing the study and their limitations. One such example would be to associate benefit with reputation. In this context, sending correct information would yield some benefit in the form of reputation, while cost means preparing such information. The problem with this is that it is not intuitive to represent the value of information in terms of reputation, and vice versa, so that we can calculate the payoff, not to mention the hardness of deciding what reputation is. It may be possible to associate the said cost with the expected loss of reputation if we ignore the value of information. Yet, it may be possible



(a) Blue win probability with changing b/c

(b) Blue win probability with changing b/c, network partitioned using community detection

Figure 4.7: Strategy dominance probabilities with different network sizes and b/c

to incorporate the value of information into the expected reputation. However, we would still need to differentiate between correct and false information.

We showed that the average number of neighbors k is indeed a factor in how the evolution of strategies among nodes occurs. In addition, we showed through simulations that prior predispositions such as existing communities (e.g., cliques or clusters) would also indicate different evolution characteristics. Prior studies on spatial evolutionary games also show such results regarding the effects of network structure [164, 165]. This motivates future work on a fine-grain analysis of the effects of connectivity, such as the size and the number of cliques, echo chambers, and types of relationships. For instance, if the value of benefit and cost were dynamic, as previously said, the feasibility of a partitioning algorithm based on the node-wise values of $b/c > k$ could be studied.

4.3 A Game between Network-level Players

In today's online social networks, there exist intrinsic actors above the node level, i.e., outside the network, with different motives such as politics, advertisement, and reputation who try to spread various information to affect people's minds using various techniques. One such technique is to control or influence a set of nodes that serve for the benefit of the actor during an information spread campaign. These nodes can be maintained by real people (sometimes called trolls) or could be bot accounts [166]. A vast amount of research exists on identifying and mitigating fake accounts in online social networks, and a recent review is provided by [167]. While dealing with misinformation through means of identifying such ingenuine accounts provides relief for the real people to be notified about such accounts and help regulate the social network, the broader problem specification is mainly associated with the area of influence maximization which deals with identifying the parameters that lead to maximal influence for various agents over the nodes of social networks. Influence maximization has been identified as an NP-Hard problem [168].

Given a social network, which is a graph with nodes representing users and edges (directed or undirected) representing a relationship (such as friendship, follow, connection), there is at least one player that controls or influences directly some of the nodes to start spreading some information. The scenario becomes misinformation propagation if the information spread falls into the misinformation category. The purpose of the player is to maximize the number of nodes affected.

4.3.1 Mechanism Design

We set the environment for the game to be the network. For the sake of simplicity, there exist two players, each given a set of randomly selected nodes. While player one spreads misinformation, the other player opposes the misinformation campaign. In each time step, each of the players utilizes one of the nodes as the seed for misinformation. As the propagation mechanism, we chose the SIR model. The reward for the players is the number of affected nodes after the game is ended. In this work, we utilize various well-known node selection algorithms and propose another one based on deep reinforcement learning using the Multi-Agent Deep Deterministic Policy Gradient (MADDPG) algorithm. The nodes that are selected by the algorithms out of a randomly selected (same for each) set of nodes then are used in the information propagation game. Below, we describe the specifics of the baseline algorithms and the proposed method.

4.3.1.1 Node-Centrality

Centrality is a measure of a node's location in the network and is generally used to identify the importance of the node. There are various techniques for calculating the value, such as the number of in-degrees and out-degrees, eigenvector centrality, Katz-centrality, and others. In this work, we experimented with various node-centrality measures and opted for the degree-centrality method as we observed that the results of those measures appear to be quite similar.

4.3.1.2 Page-Rank

Page - Rank was introduced by the Google Search Engine to find out the importance of web pages. Today it has been modified and used in many areas, including social network analysis as a measure of node importance.

4.3.1.3 Greedy

The greedy algorithm was proposed by [168]. It takes a network with n nodes and computes the spread value until it finds k nodes with maximal marginal spread. Its complexity is $O(kn)$ multiplied by the time required by the spread. Theoretical guarantees exist, mentioning that the algorithm achieves at least 63% of the spread resulting from the optimal set.

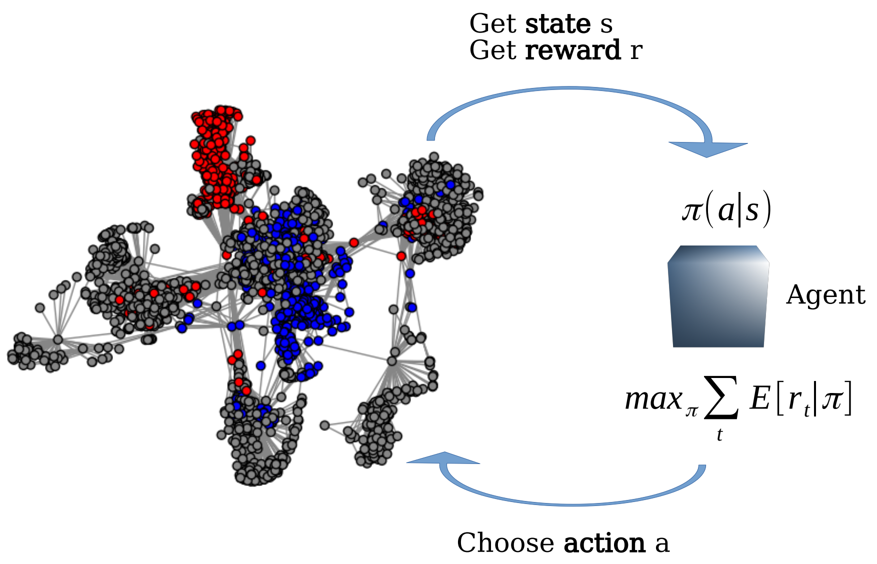


Figure 4.8: Reinforcement learning problem over a social network

4.3.1.4 Cost Effective Lazy Forward (CELF)

CELF [169] is a modification of the greedy algorithm that achieves the same results with less computation using an optimization technique called lazy-forwarding.

4.3.1.5 Proposed Method based on MADDPG

We approach the selection of nodes for misinformation or countering it from a reinforcement learning (RL) perspective. The problem statement is as follows: The social network is an environment consisting of states, actions, and rewards. At any point in time, the state s is a list of node stances, the actions are a list of selected nodes, and the rewards are the number of nodes affected by the actions. Is it possible to learn a policy π that could maximize the expected reward over time? ($\sum_t E[r_t|\pi]$) (see Figure 4.8)

An RL problem can often be described as a Markov Decision Process (MDP), which contains the transition function that encapsulates the state-to-state transition probabilities and the reward function that outputs the value of the reward given the current state. In such a context, the transition and the reward functions can be thought of as the model of the environment and provide a basis for a subset of RL algorithms called the "model-based" algorithms which utilize the said model to find an optimal policy that gives the maximum expected reward.

However, in some cases, the definition, the transition probabilities, and the associated reward functions of an MDP are unknown for various reasons, such as the complexity of the environment or purely design choices. The RL algorithms that are specifically designed to learn in such environments are called the "model-free" RL algorithms. These do not utilize the transition and reward functions but rather often have a way to learn a value for the current state of the environment explicitly by interacting with the environment. This value function can then be used to determine a policy.

Q-Learning can be recognized as the starting point of such approaches that are based on trial-error; however, as the action/observation spaces grow exponentially with respect to the complexity of the environments, the need for deep neural layers introduced other methods such as DQN instead of keeping track of every action-state tuple [170] in a Q-table. There are also Policy Gradient-based algorithms [171] which are used with continuous action spaces where a policy is a parametric distribution, and these parameters are adjusted using gradient descent. These algorithms led to actor-critic methods, Deterministic Policy Gradient (DPG) algorithms [172], and an algorithm called Deep Deterministic Policy Gradient (DDPG) [173]. Two possible problems related to stability arise in the use of DPG algorithms. The first is related to the method being "on-policy" - which means that the critic evaluates the value of actions based on the same policy - creating possible bias [174], by disabling the utilization of a stabilization mechanism such as the experience replay buffer in DQN. The second issue is the sample complexity problem that is related to the required number of samples for efficient learning [175, 176]. In DDPG, there is a single agent with actor and critic networks where the actor-network chooses an action based on the state of the agent, and the critic network determines the value of that selection. To reduce the previously stated stability problems, DDPG first uses an experience replay buffer to store past transitions to operate “off-policy”. Second, it employs target networks associated with the actor and critic networks combined with a soft-update mechanism to increase stability. [177] iterates the possible failures and problems in DDPG that may result in poor learning.

Multi-Agent Deep Deterministic Policy Gradient (MADDPG) [178] was offered as an extension to DDPG for multiple agents. In MADDPG, all agents again have their actor and critic networks; however, critic networks have full access to the environment. In addition, MADDPG utilizes a mechanism called the policy ensembles for more robustness, along with the inherited mechanisms from DDPG. Instead of relying on a single policy per agent, an ensemble of policies exists to sample from. In this work, we chose MADDPG as it reportedly outperformed various other methods [178] previously, and it supports continuous action spaces in multi-agent environments. Also, the agents can see the actions of other agents (even if partially), which is suitable for the scenario in the scope of this work.

The MADDPG architecture is comprised of actor and critic networks along with their target networks. The environment is the set of n nodes, and the observations are the states of the nodes. Each node can have one of the 3 states, infected, neutral, or recovered (i.e., under-misinformation, neutral, or correctly informed). We set up the network to take the n node states as input, and the number of outputs is set out as the number of seed nodes s. The outputs are continuous values and are sorted at the end. The network chooses the output with the largest value in a sense. The other methods are also given the seed s nodes as the input and choose a subset of k nodes to be the originator nodes. This means that CELF, for instance, which is an algorithm that selects the best nodes in the network, is now modified to select the k best nodes from a subset of s nodes. This could potentially undermine the theoretical guarantees mentioned earlier. However, in real networks, the actors cannot choose nodes at will from the entire network but have to work with what they have. Nonetheless, the brute force algorithm still requires ${}^{s}\mathrm{P}_k = s!/(s-k)!$ number of cascades.

Table 4.4: The hyper-parameters for the MADDPG architecture

Hyper-parameter	Value	Hyper-parameter	Value
Actor Network	1024x512	Actor Optimizer	Adam
Critic Network	1024x512	Critic Optimizer	Adam
Actor Learning Rate α	0.0005	Tau	0.01
Critic Learning Rate β	0.0005	Gamma	0.95
Batch Size	128		

Table 4.5: Data sets

Network	Type	Nodes	Edges	Reference
Facebook EgoNet	Undirected	4039	88234	[179]
Twitter EgoNet	Directed	81306	1768149	[179]
Facebook MIT	Undirected	6402	251230	[180]
Epinions	Directed	26588	100120	[181]

The hyper-parameters for the MADDPG architecture are given in Table 4.4.

4.3.2 Experimental Results

In our experiments, we use four fairly large networks, two from Facebook, one from Twitter, and one from the Epinions.com dataset. The details are given in Table 4.5. In our experiments, we randomly select $s = 100$ nodes per game as the pool for selection for the algorithms. The algorithms then select top $k = 20$ nodes as originators. We then run one spread iteration per originator node and observe the total spread. We continue until all k nodes are exhausted. The spread dynamic is chosen as SIR (Susceptible - Infected - Removed), and we utilize the following transitions probabilities: $P(I|S) = 0.02$, $P(R|I) = 0.01$, $P(R|S) = 0.01$.

During the training, we incorporated two mechanisms for improving the agent networks. The first one is to introduce noise to facilitate learning. We found that the addition of noise is critical for exploration. While experimenting with various noise mechanisms, we decided on a noise function based on an Ornstein-Uhlenbeck process, also called the Vasicek model (Eq. 4.6). The first part defines

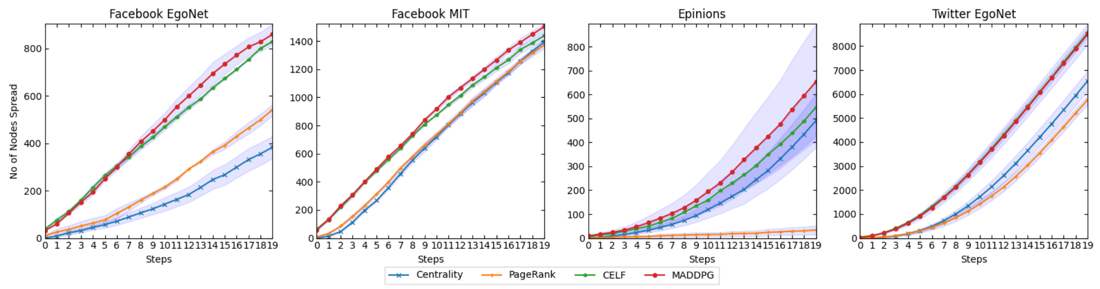


Figure 4.9: Agent 1 - cumulative spread

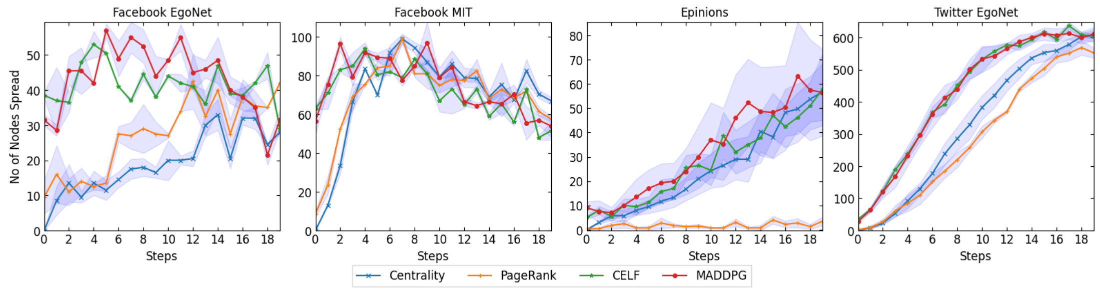


Figure 4.10: Agent 1 - spread at each step

the drift over X where $|X| = s$. μ defines the long-term mean, and θ is the mean reversion speed. The second part - dW_t is the discrete form of a Wiener process (Eq. 4.7) at time step dt, where W is is a random variable between $[0, T]$ and σ is the scale of randomness, i.e., volatility. Given $W_0 = 0$; for $0 < s < t < u < v < T$, $W_t - W_s$ and $W_v - W_u$ are independent increments and these increments follow a Gaussian N distribution with zero mean and unit variance. The noise, then, is sampled at time step t and added as $X_t + dX_t$.

$$dX_t = \theta(\mu - X_t)dt + \sigma dW_t \tag{4.6}$$

$$dW_t \sim \sqrt{dt}N(0, 1) \tag{4.7}$$

The second improvement is to decide when to save checkpoints during the training by using a sliding window of size k for the past rewards. Here, there were many available options, such as sum, mean, rolling sum, etc., but we used the area under the curve (AUC). If the window has a larger AUC than the previous best, we save the checkpoint.

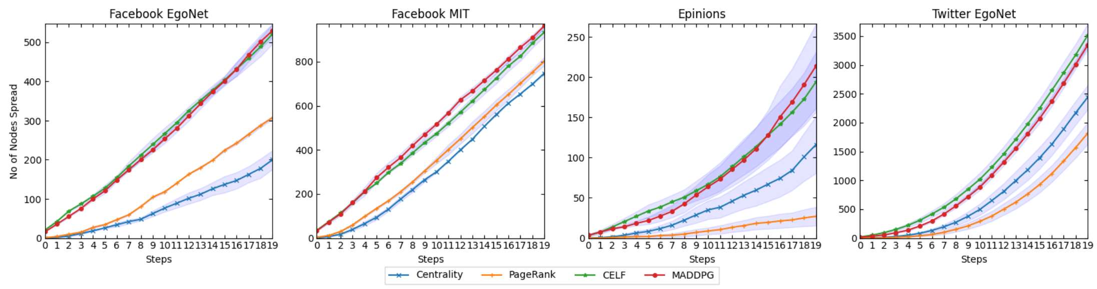


Figure 4.11: Agent 2 - cumulative spread

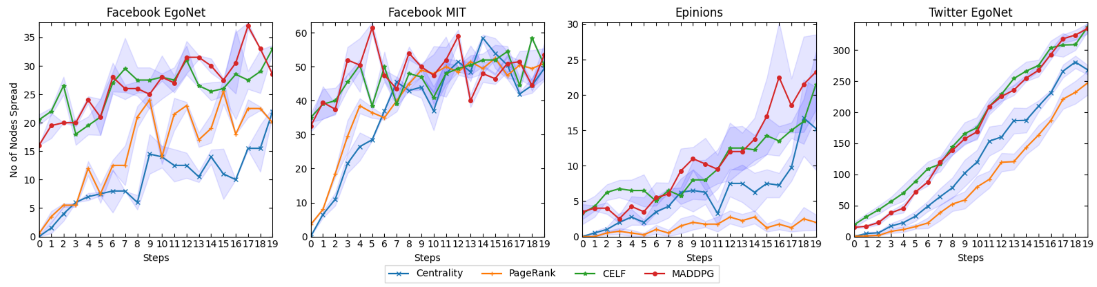


Figure 4.12: Agent 2 - spread at each step

We report the results of Agent 1 - who tries to maximize the spread of misinformation, and Agent 2 - who tries to minimize it. We experimented with various combinations of s and k, and as the results were similar, we only report the results in the mentioned setting. The MADDPG agents were trained a maximum of 1000 times per game. We played 100 games for the results. The results are given as the mean curves and the 95% confidence interval was also plotted. We omitted the results for the Greedy algorithm as the results coincide with the CELF algorithm, as previously expected.

Figure 4.9 contains the results for Agent 1 and the cumulative spread. The results show that the agent effectively learns the set of influential nodes as compared to other algorithms, and even outperforms an established algorithm - CELF in most cases. Figure 4.10 gives the spread at each step. Figure 4.11 and Figure 4.12 give the cumulative and step-by-step spreads of Agent 2, respectively. The results are similar. We see that the improvement experienced by Agent 1 is also experienced by Agent 2. We see that the undirected networks show different characteristics than the directed ones. For directed graphs diffusion happens much faster considering the number of nodes spread. This may be due to the size difference between those networks and connectivity (e.g., the average number of neighbors) inside the network. We also notice that the spread amount for Agent 2 is around half of Agent 1 for directed graphs, which is expected as we set up the transition probabilities of the SIR model that way. However, Agent 2 seems less successful in directed graphs - Epinions and Twitter - than the undirected graphs, the cumulative spread not reaching half of Agent 1 for these networks. It should also be noted that the classical methods - Centrality and Page-Rank - still seem practical choices for node selection tasks.

4.4 Discussion

One of the main issues of using a deep neural network is the interpretability of results, i.e., making sense of its choices. This also remains an issue in our work to be explored in the future.

In this work, we did not utilize any node representation techniques such as an adjacency matrix, convolutionary graphs nodes or a learned representation such as node2vec [182] or a network representation scheme such as averaging over node2vec embeddings, DeepWalk [183] or anonymous walks [184]. This situation creates two immediate limitations. First, it takes longer to train the network if we do not provide the node representations. Second, the trained agents cannot be generalized/transferred to work for other social networks but instead work for the trained network only. However, there are also opportunities in the approach. As the agents learn from the bare states of the nodes, the resulting actions could be used as embeddings - a new vectorized representation for the network states and the ranked significance of nodes. These embeddings can be used in various research tasks in different areas, such as the vaccination problem, node-blocking, cloud computing, etc.

4.5 Summary

In this chapter, we tackled the problem of misinformation propagation in online social networks from a game perspective. First, we approached the problem from the node-level point of view, where nodes were the actual players. We illustrated that the misinformation game constructed as a cooperative game on graphs displays the same characteristics that were explored in the literature. On the other hand, it has practical drawbacks, such as determining real values for variables such as benefit and cost described within the game dynamics. On the other hand, a more practical approach is possible with network-level players. We showed that a deep reinforcement learning algorithm based on MADDPG can select an influential set of nodes in terms of misinformation propagation, and it gives promising results against various well-known algorithms such as CELF, Page-Rank, and Node-Centrality.

In future work, the explainability of the selections by the neural nets of RL agents could be studied to understand and implement better defense scenarios to stop misinformation dissemination. In addition, the behavior of the RL agents could be investigated in different types of networks and different tasks, domains, and different spread characteristics that are associated with the applications of node importance such as epidemiology, vaccination, cloud computing, and Internet of Things.

Chapter 5

A Blockchain - Deep Learning Hybrid Approach for Crowdsourced Detection

5.1 Introduction

In this chapter we motivate the use of blockchain and describe our methodology on how blockchain can be utilized to address the misiformation problem. Our contributions in this chapter are as follows:

- We identify the critical factors in deception by referencing the Interpersonal Deception Theory and provide a model for misinformation in online social networks. We describe how blockchains are one of the candidates that can provide a better solution.
- We describe and implement a crowdsourcing mechanism on the blockchain, specifically on Ethereum, using smart contracts towards identifying true and false stories.

- We add a deep learning system that takes part in identifying the malicious actions in the crowdsourcing mechanism and the final decision on the truthfulness of stories.
- We train and test the system under various types of attacks and provide the results.
- We provide a case study on Twitter Birdwatch data and compare our results with another study on the same data set.

Section 5.2 gives our interpretation on how misinformation occurs on social networks. Section 5.3 describes our methodolody. Section 5.4 gives the details on our experimental evaluation. In Section 5.5, we discuss the effectiveness of the system and how it can be improved to address the misinformation problem with future research directions. Finally in Section 5.6, we summarize our work in this chapter.

5.2 Modeling the Misinformation Process

The Oxford English dictionary defines misinformation as "the act of giving wrong information about something". Online misinformation, in this case, defines a form of misinformation that is transmitted through online tools such as websites and social networks. In various studies, disinformation is considered as a subset of misinformation, while some studies separate disinformation as the intentional form of misinformation [28]. Although studies that try to distinguish disinformation and misinformation may yield the original intent of the spreader, in our model, we emphasize that there does not exist a practical difference affecting its propagation, since the intent of the spreader will be a hidden variable.

We define misinformation as a framework between senders and receivers on a platform that enables a publish/subscribe social network setting. This is the most generic form of a social network. We identify three intuitive mechanisms; Message Preparation Mechanism, Medium, and Message Propagation Mechanism. We

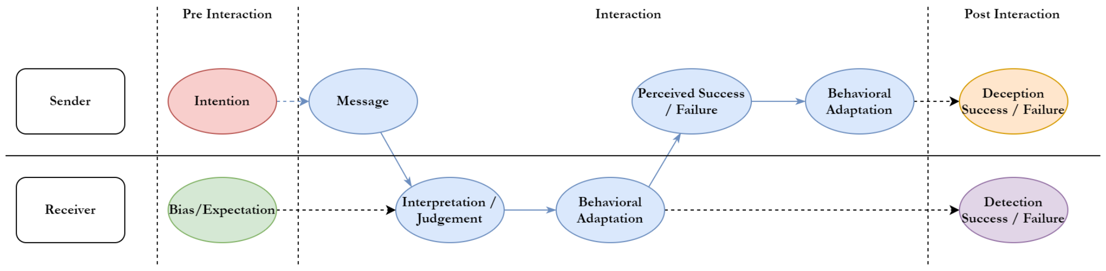


Figure 5.1: An adapted diagram of Interpersonal Deception Theory

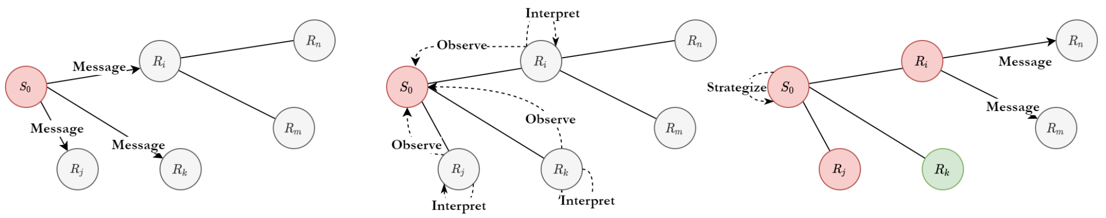


Figure 5.2: A networked model of misinformation propagation

further emphasize various latent variables that emerge with the introduction of Interpersonal Deception Theory. We argue that vulnerabilities in the mentioned mechanisms and variables worsen the spread of misinformation; hence, solutions that aim to stop misinformation should consider targeting these.

5.2.1 Interpersonal Deception Theory

Interpersonal Deception Theory (IDT) was established by Buller and Burgoon [185]. It uncovers the dynamics of deception in interpersonal communication. According to the theory, communication occurs between a sender and a receiver in a repeated fashion based on some communication variables and sender/receiver properties which can affect the success of deception or detection. It is basically about face-to-face communication between individuals. A diagram of communication-based on IDT is presented in Figure 5.1.

In online communication, face-to-face clues are non-existent, but linguistic clues and cognitive dynamics are still in play, and they happen in a networked setting, which could potentially augment or affect communication dynamics. We adopt the misinformation model depicted in Figure 5.2. In a scenario where

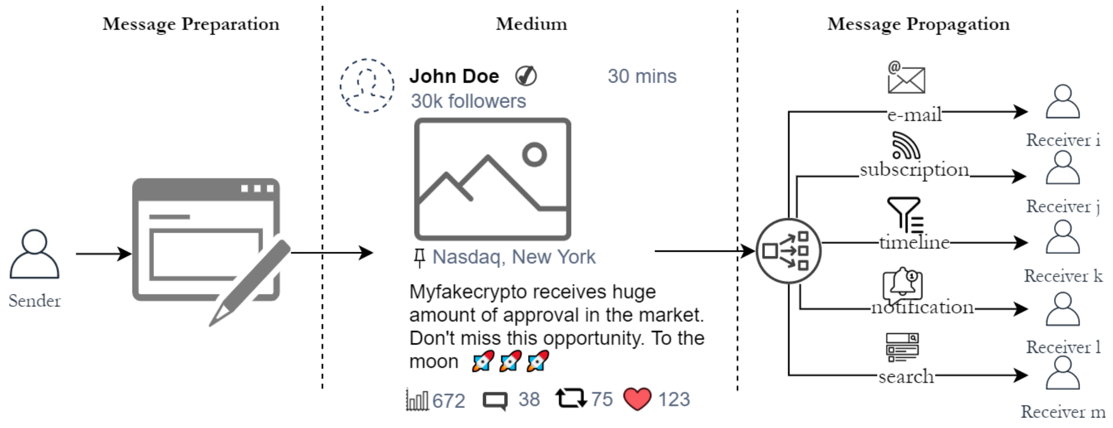


Figure 5.3: A generic model of online misinformation

people follow each other and there are mechanisms for sending messages and sharing content (e.g., Tweets, Retweet on Twitter, Sharing other platforms), misinformation can be sent via a message from an original sender who may or may not have malicious intent to influence other people in its proximity, which could, in turn, be adopted by its followers and spread to their neighbors. According to the IDT, the original sender with malicious intent can observe the reactions (via e.g., Share, Retweet, Like, Dislike, etc.) to adjust its behavior for future messages (transactions). A person with no malicious intent can also help spread the misinformation and indirectly serve the malicious intent, where the content of the message would not yield the original intent. Although people are rational beings and say that they can identify deception easily, the research argues against it [186] with only 54% can identify deception, which is slightly better than tossing a coin. In addition, before even seeing the message, people would already have a perception of the subject of the topic; hence, this should also be taken into consideration [28].

However, a more generic model is needed to represent and identify certain parts of online communication that play a critical role in the dissemination or suppression of misinformation. We first explain each part and associated vulnerabilities.

5.2.2 Message Preparation Mechanism

This part represents the initial part of the misinformation spread. A message is created by the sender and submitted in this stage. The online platforms generally enable this with a submission form, which could be decorated by the platform with various features such as location, tags of people, etc. This part is vital to stop misinformation even before it is introduced. This part is under the control of the online platform. The features of the message and the sender could hint at whether misinformation is present.

Message

A message can be generalized as a set of propositions. The propositions are either direct or indirect which are achieved by the lexical, syntactic, and semantic features aided with various other features such as videos, images, and other digital content. Once the message is delivered, it is interpreted by the receiver, not necessarily yielding the same propositions as they are created. The intent, however, is an intrinsic variable that could help shape the message and explain the rationale for the message being created. However, the intent is not present in the message since a person with no malicious intent could unintentionally spread a message with malicious intent, achieving the same results, even worse [28, 187]. A message can be described as a set of one or more propositions that are communicated via textual or other digital content. While it is possible to argue that each of these propositions should be true for the message to be true, it is still possible that even that is not enough.

Other methods may include fuzzy logic-based approaches where the message has a degree of its truthfulness.

In this stage, various precautions could be implemented to prevent misinformation even before it is created, for instance, detecting a possibility of misinformation by analyzing the content of the message using various machine learning techniques. Recent studies are summarized in [188].

Sender

There are various features such as previous history, profile, and the tendency of the sender that could indicate the existence of misinformation. One of the most hidden features of the sender is the intention [187], which is hard to tell. For instance, while the original poster of a fake story most likely has the intention to deceive, some other users may simply believe the original post and re-post the story. Possible vulnerabilities include weaknesses in the authorization and authentication of the user, such as the creation of Sybil or bot accounts [68].

5.2.3 Medium

The medium is the basic functionality of the online platform. It connects people and allows interaction with posts with various means (e.g., likes, shares, retweets). It enables the people in the network to observe the statistics about the posts of other people or their own. It is when you see how many people watched a video or added a comment to it. This part is also managed by the online platform. Every textual, visual, and auditory clue provided by the medium could potentially augment and diminish the effects of misinformation attempts.

5.2.4 Message Propagation Mechanism

This part represents how the initial message is delivered to the other nodes using the medium of the platform.

Delivery

The delivery mechanism could be a simple publish-subscribe method with which subscribers are notified when a publisher publishes a message. However, the order that they receive the message could also be very important [189]. The order, again, is under the control of the online platform, which is proprietary in many cases. Online platforms employ various ranking algorithms to make some posts more visible and with various other external tools such as push notifications, newsletters, and e-mails. Transparency in this part cannot be guaranteed unless all the corresponding algorithms are made public and it is shown that there is no bias or temperament of visibility.

Receiver

Most of the features of the Receiver are the same as the Sender but for the misinformation to be successful, the interpretation of the receiver is the most important factor. Expectations, knowledge level, and bias towards the message content or the sender are the features to consider. Failing to authenticate the validity of the receiver is also a problem for the dissemination of the information, as Sybil and Bot accounts can interact with a correct message in a malicious way. The reverse is also through where these accounts could be commanded to act in favor of an adversarial sender.

5.3 A Generic Blockchain Approach

5.3.1 Background

Blockchain is a distributed ledger technology that allows for the secure and transparent storage of data. It is a decentralized system, meaning that it is not controlled by any single entity, but rather is maintained by a network of participating nodes. Blockchain was developed as a way to enable secure and transparent online transactions without the need for a central authority or third-party intermediary.

Ethereum is a blockchain platform that was developed to enable the creation and execution of decentralized applications (dapps). These dapps are built using smart contracts, which are self-executing contracts with the terms of the agreement between buyer and seller being directly written into lines of code. The use of smart contracts allows for the automation of complex processes and can help reduce the costs and risks associated with traditional contracts.

Solidity is the programming language used to write smart contracts on the Ethereum platform. It is a high-level language, similar to JavaScript, that is designed to be easy to write and compile down to the low-level code that can be executed on the Ethereum Virtual Machine (EVM). Solidity allows developers to create smart contracts that are both functional and secure.

Ganache is a local Ethereum blockchain that can be used for testing and development purposes. It allows users to create a private blockchain where they can deploy and run smart contracts without having to connect to the main Ethereum network. This makes it a useful tool for experimenting with smart contract development and for testing the functionality of dapps before deploying them to the main network.

5.3.2 Proposed Approach

In social networks such as Twitter, Facebook, and Reddit, we see users creating posts and other users interacting with those posts through various means such as liking, retweeting, and sharing. Each of these platforms has its ways with which the posters accumulate reputation-like standing within the social network. This can either be implicit (e.g., number of followers, number of retweets) or explicit (e.g., Karma system in Reddit). Yet, there does not exist a metric to identify the validity and the reliability of the posts in terms of truthfulness, in addition to the lack of responsibility for the posters as well as the other users interacting with the posts. Therefore, the social platform needs to accommodate the validity, immutability, and non-repudiation of any transaction regarding posts to facilitate more misinformation-free communication.

A blockchain can deliver an environment of such transactions where validity, immutability, and non-repudiation are ensured while also making sure that no single entity determines how the transactions are handled. Let us define an environment where there exist users who are capable of posting stories and voting on stories posted by other users. Votes denote the personal judgment of each user regarding the truthfulness of the story. This environment aims to exploit the benefits of collective intelligence provided by crowdsourcing to identify the validity of the information.

We implement the crowdsourcing mechanism using Ethereum-based smart contracts in the Solidity programming language. We use the Ganache framework to create, compile, deploy and test the smart contracts on a local development environment. Each user has a unique identifier on the blockchain. The smart contract enables users to post stories and vote on stories that are not their own. Both of these actions are called transactions and are immutably stored on the blockchain. Extra features of the users, such as their reputation, are also stored on the blockchain.

Once a story is posted, users start upvoting (+1) or downvoting (-1) the story, indicating their judgment that the story is true or false (fake), respectively. For determining when to terminate voting for a particular story, we experimented with mechanisms such as calculating the momentum of the incoming votes, the entropy of the votes, the entropy change via Kullback-Leibler divergence, monitoring for a particular portion to vote on the story or simply monitoring for a maximum number of votes. However, we decided to use a combined approach to determine the equilibrium for a particular story that involves a threshold mechanism determined by a classifier, entropy, vote count, and Lyapunov exponents which we describe in detail in the following section (See Algorithm 3). This is also the time we distribute the rewards and punishment in the blockchain involving this particular story.

5.3.2.0.1 Lyapunov Exponents Lyapunov exponents are often used in the study of chaotic systems, as they provide a quantitative measure of the degree of chaos in a system. They are also used in the study of complex systems, as they provide a way to understand the behavior of a system at a macroscopic level, even if the system is made up of many interacting parts.

In general, a positive Lyapunov exponent indicates that the system is chaotic and sensitive to initial conditions, while a negative Lyapunov exponent indicates that the system is stable and insensitive to initial conditions. A system with zero Lyapunov exponents is considered to be periodic. If some of the Lyapunov exponents are positive and some are negative, then the system is somewhere between stable and chaotic.

First, let us define the variables that we will be using:

- $\mathbf{x}_t$: The state of the system at time t.
- $\mathbf{F}(\mathbf{x}_t, t)$: The dynamics of the system, which describes how the state of the system changes over time.
- $\mathbf{V}_t$: The Jacobian matrix of the system at time t.
- Δt: The time step size.

With these definitions, we can express the Lyapunov exponents as follows:

$$\mathbf{V}_{t+\Delta t} = \mathbf{V}_t + \frac{\partial \mathbf{F}}{\partial \mathbf{x}_t} \mathbf{V}_t \Delta t$$

This equation describes how the Jacobian matrix changes over time. To calculate the Lyapunov exponents, we can iterate this equation over a series of time steps, starting from an initial time t_0 and ending at a final time t_1.

$$\begin{aligned} \mathbf{V}_{t_0} &= \mathbf{I} \\ \mathbf{V}_{t+\Delta t} &= \mathbf{V}_t + \frac{\partial \mathbf{F}}{\partial \mathbf{x}_t} \mathbf{V}_t \Delta t \end{aligned}$$

$$\mathbf{V}_{t+2\Delta t} = \mathbf{V}_{t+\Delta t} + \frac{\partial \mathbf{F}}{\partial \mathbf{x}_{t+\Delta t}} \mathbf{V}_{t+\Delta t} \Delta t$$
$$\vdots$$
$$\mathbf{V}_{t_1} = \mathbf{V}_{t_1-\Delta t} + \frac{\partial \mathbf{F}}{\partial \mathbf{x}_{t_1-\Delta t}} \mathbf{V}_{t_1-\Delta t} \Delta t$$

Once we have calculated the final Jacobian matrix $\mathbf{V}_{t_1}$, we can use it to calculate the Lyapunov exponents as follows:

$$\mathbf{L} = \text{eig}(\mathbf{V}_{t_1}) \tag{5.1}$$

Here, *eig* is a function that calculates the eigenvalues of a matrix, and **L** is the list of Lyapunov exponents.

5.3.2.0.2 Shannon Entropy

$$H = -\sum_{i=1}^{n} p_i \log_2 p_i \tag{5.2}$$

where n is the number of different transactions, and p_i is the probability of transaction i.

To calculate the probabilities p_i, we need to count the number of times each transaction occurs in the data. Let N_i be the number of times transaction i occurs in the data, and let N be the total number of transactions. Then, the probability of transaction i is $p_i = \frac{N_i}{N}$.

5.3.2.0.3 Vote Count As both Lyapunov exponents and entropy work better with more data, we need to handle the case where there exist a low number of votes on a story. To this purpose, we add another metric called "vote count", which means that the number of votes should exceed a minimum amount C_{min} to be able to terminate voting on a story.

Algorithm 3 Decide whether a system of transactions has reached equilibrium

```
Require: $\mathbf{L}$: Lyapunov Exponents
Require: $\mathbf{H}$: Shannon Entropy
Require: $\tau$ Entropy Threshold
Require: $\mathbf{C}$ : Count of transactions (Votes on story)
Require: $\mathbf{C_{min}}$: Minimum number of transactions
 1: procedure Equilibrium($\mathbf{L}, \mathbf{H}, \tau, \mathbf{C}, \mathbf{C_{min}}$)
 2:     $LEquilibrium \leftarrow True$
 3:     for $\lambda$ in L do
 4:         if $\lambda \geq 0$ then
 5:             $LEquilibrium \leftarrow False$
 6:             break
 7:         end if
 8:     end for
 9:     if $LEquilibrium$ and $H < \tau$ and $C >= C_{min}$ then
10:         return True                  ▷ "The story is stable and in equilibrium."
11:     else
12:         return False             ▷ "The story is chaotic or unstable and not in
        equilibrium."
13:     end if
14: end procedure
```

5.3.3 Learning Problem

As we previously mentioned, a crowdsourcing system should be resilient against malicious behavior. This requires further support from sophisticated algorithms. While it is possible to create a rule-based behavior detection system, the complications would yield the need for a more automated system that is capable of learning over time and detecting more sophisticated adversarial scenarios that are unknown at the time. As there exists a rising concern about the use of bots in the literature [190, 191], the system should be more intelligent than those bots. To this purpose, we employ a two-stage learning architecture combining two different classifiers learning two different models, as depicted in Figure 5.4. The first one - *"Action Classifier"*, learns to distinguish malicious behavior. For this, it is fed time-series data consisting of quadruples in the form (user, story, type, vote).

We consider two key aspects of representing users, stories, and user votes to solve the learning problem. The first aspect is the sequential form of the data,

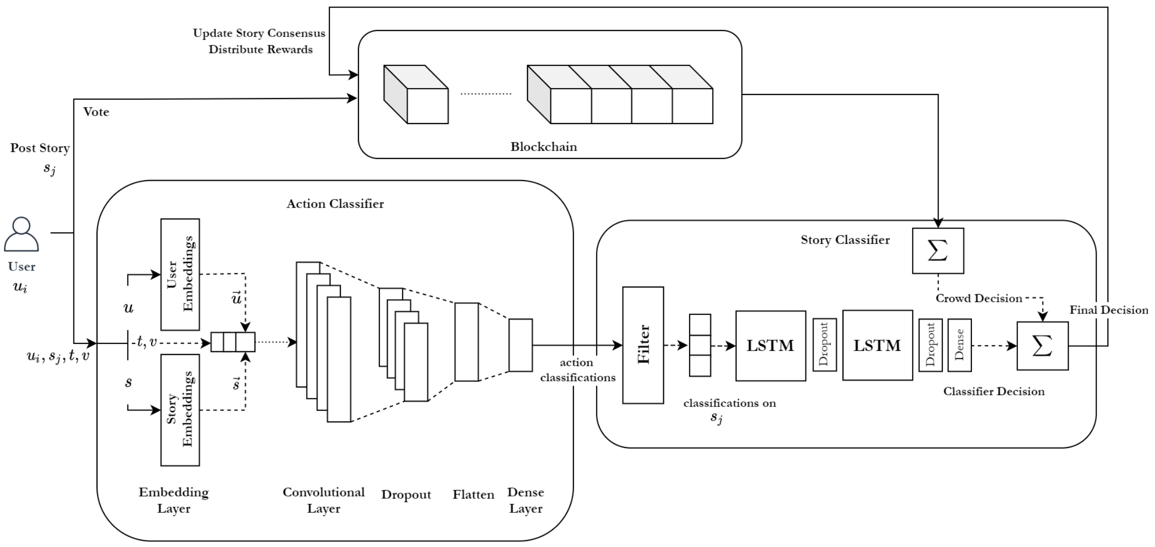


Figure 5.4: Overall architecture of the solution

which captures patterns in user behavior. The second aspect is the need to maintain user-user and story-story similarities. To address these considerations, we propose representing actions as triples of (user, story, vote), which preserves the sequential dimension but fails to provide user-user or user-story similarity. Another approach is to represent actions as a user-story matrix with votes as values, which preserves general similarity but lacks the sequential dimension and suffers from sparsity. Ultimately, we propose combining both approaches to fully capture both aspects. We employ two separate embedding layers that learn representative vectors of users and stories of sizes $log_2|U|$, $log_2|S|$, respectively, where $|U|$, $|S|$ represent the maximum number of users and stories, respectively. We then concatenate those with their corresponding type and vote values. These are then fed into a convolutional layer that utilizes an additional max-pooling layer. A dropout layer is used to prevent overfitting, and finally, a dense layer outputs the classification on stories using a $tanh$ activation function.

To classify the story based on the previous behavior of the users and corresponding classifications of the *Action Classifier*, we implement the *Story Classifier* that utilizes two LSTM layers with a dropout layer after each and a final dense layer. The input data is the time-series classifications of the first classifier. Once we predict the label for the story, it is then combined with the crowd decision as the final decision. If this value is greater than a threshold, it is marked as consensus and the story is updated on the blockchain, and the voting is terminated. After this, a reward is distributed to users that correctly identify the consensus label, and punishment is distributed otherwise. The poster is also rewarded if the posted story is true, and punished otherwise. All the cumulative rewards and punishments constitute the reputation score of the user. The rewards and punishments are kept very straightforward; the user that correctly identifies a false story or a correct one receives 1 unit of reputation, otherwise punished by -2 units. The poster of a correct story is rewarded 2 units of reputation and punished by -4 units otherwise.

5.4 Experimental Evaluation

The experimental setup contains a customizable environment for various attack scenarios and a variable number of users and transactions (stories and votes). By varying the parameters, we measure and analyze the behavior of the system.

We define users who can distinguish true and false stories as "normal". These users do not have any other agenda than making a good judgment.

We also define a user type called "trolls". These users downvote a true story and upvote a false one. Even though these do not have specific targets, they can be quite disruptive. This is sometimes called the "naive" attack.

In another type of attack, malicious users build a good reputation over time, then use this reputation to perform short-lived sets of malicious actions [140]. This attack is called the "on-off" or the "oscillation" attack. The related users are called the "traitors" by [138], which we implemented in the system as well.

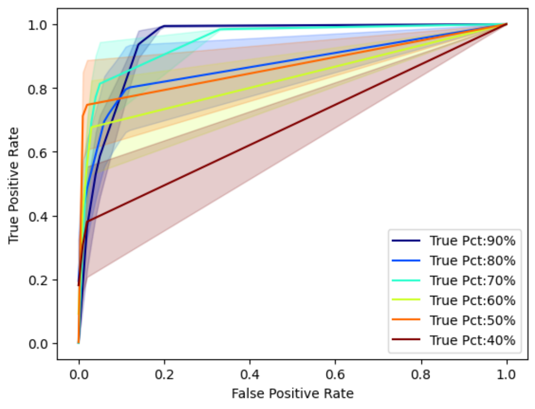
Random behavior is one of the hardest attacks to countermeasure in reputation systems [127]. Users with this behavior act randomly, regardless of the story.

Slandering attack or badmouthing is the act of downvoting specific users but acting normally against others. The reverse of this is called “whitewashing” or “promoting”, which is the act of upvoting certain users’ posts even if they are fake.

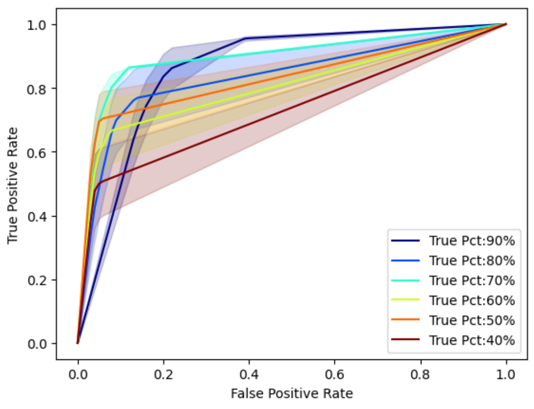
When there are groups of users involved in more sophisticated types of attacks, these are called “orchestrated” or “coalitional” attacks. There are many ways user groups can attack. In this study, we implemented “orchestrated slandering”, in which a group of users slander a set of target users, and “orchestrated whitewashing”, in which they upvote a set of target users. There can be many other attacks, such as one version of the oscillation attack but performed by groups of users. To summarize, there exist the following malicious user types in our simulations in addition to the non-malicious user type normal: troll, random, traitor, and, orchestrated. There are also target users for slandering and whitewashing who act as normal users.

In our simulations, we use varying ratios of true and false stories. To understand the effect of initial story distribution on the final outcome of misinformation detection, we perform various tests. By keeping the number of users, the ratio for various types of users, and the number of stories the same, we perform a test only by varying the true/false ratio of stories. Figure 5.5 gives the receiver operator curves (ROCs) along with their 95% confidence intervals, as we performed the test 5 times for each. According to the curves on the train and test sets, we can observe that the classification performance is affected by the initial distribution, especially the true positive rate.

To understand how the attack types affect the classification success of the system, we utilize Ordinary Least Square (OLS) Regression analysis by using Precision, Recall, F1, and Accuracy values from 1000 runs as our dependent variables. In the tests, the percentage of normal users ranges from 30 to 70, while the remaining is distributed to the malicious types where there exist at least 5 percent of each type. The results are given in Table 5.1. If we look at the coefficients, the normal type of users contributes positively to the metrics, as expected, and the other types of users contribute negatively. We can generalize



(a) Train



(b) Test

Figure 5.5: ROCs for train and test datasets where there exist $n = 100$ users, $k = 500$ stories, $v = 2500$ votes. Percentages of the user types: normal:70, troll:5, random:5, traitor:5, orchestrated:5, slandering targets:5, whitewash targets:5

that troll attacks have the largest effect on the outcome negatively, followed by traitors, random and orchestrated, in this particular order. This is a surprising finding because trolling is not a particularly sophisticated adversarial strategy.

Table 5.1: OLS regression results for various attack types: *1000 runs, normal type 30-70%, other types distributed and at least >5%*

	Precision			$R^2 = 0.621$	Recall			$R^2 = 0.705$
Type	Coef	Std Err	t	$P > \|t\|$	Coef	Std Err	t	$P > \|t\|$
normal	-2.945e-05	7.02e-05	-0.419	0.675	0.0003	0.000	2.315	0.021
troll	-0.0044	0.000	-37.270	0.000	-0.0097	0.000	-42.495	0.000
random	-0.0018	0.000	-15.585	0.000	-0.0044	0.000	-20.085	0.000
traitor	-0.0033	0.000	-29.777	0.000	-0.0074	0.000	-34.168	0.000
orchestrated	-0.0002	0.000	-1.873	0.061	-0.0002	0.000	-0.712	0.476
	F1			$R^2 = 0.736$	Accuracy			$R^2 = 0.705$
normal	0.0002	9.86e-05	1.630	0.103	0.0003	0.000	2.315	0.021
troll	-0.0078	0.000	-46.832	0.000	-0.0097	0.000	-42.495	0.000
random	-0.0034	0.000	-21.230	0.000	-0.0044	0.000	-20.085	0.000
traitor	-0.0059	0.000	-37.403	0.000	-0.0074	0.000	-34.168	0.000
orchestrated	-0.0002	0.000	-1.117	0.264	-0.0002	0.000	-0.712	0.476

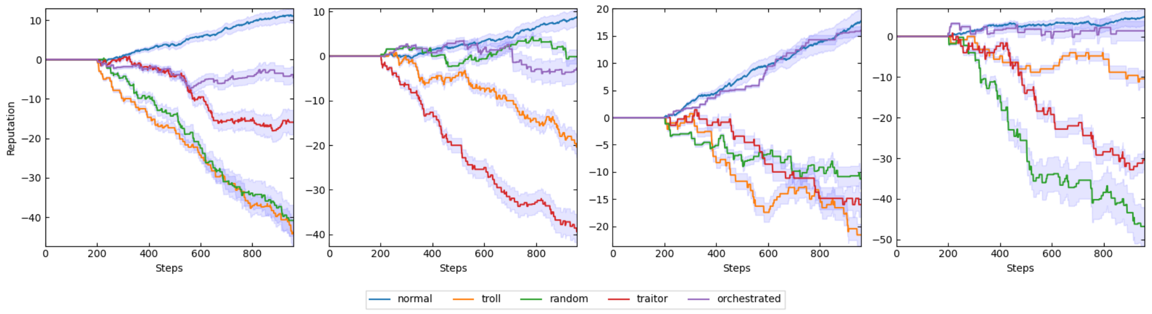


Figure 5.6: Reputation means of user types over time, $n = 100, s = 200, v = 1000$

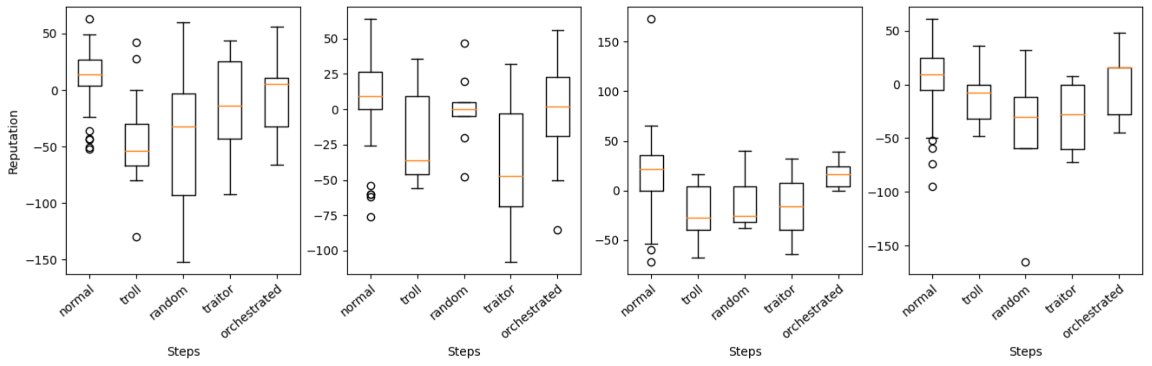


Figure 5.7: Final reputation distributions of user types, $n = 100, s = 200, v = 1000$

While most of the results are significant, the effect of orchestrated attacks is not shown to be p¡0.05, which means either we have to run more tests or the attack actually does not have a significant impact on the overall outcome, while benefiting the attackers. It should be noted that R^2 values also show relatively good models.

To understand how the reputation changes over time and how it is distributed for various user types, we perform various experiments. In these experiments, we set up various User-Story-Count triples and vary the percentage of initial user type distribution. In these experiments, the normal user type varies from 42% to 70%, target user types are fixed at 5%, and malicious user types are equally distributed. For instance, for 100 users, 200 stories, and 1000 votes, Figure 5.6 gives the results for varying ratios of user types. Figure 5.7 shows the final distributions of reputations for these scenarios. Figure 5.8 and Figure 5.9 show

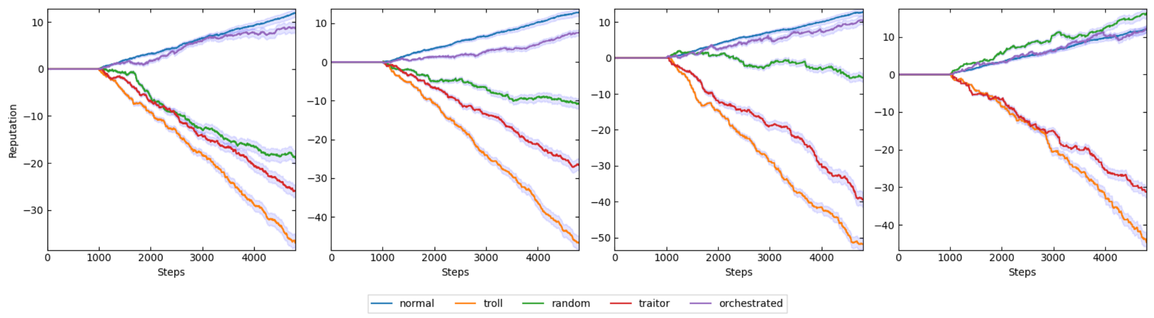


Figure 5.8: Reputation means of user types over time, $n = 500, s = 1000, v = 5000$

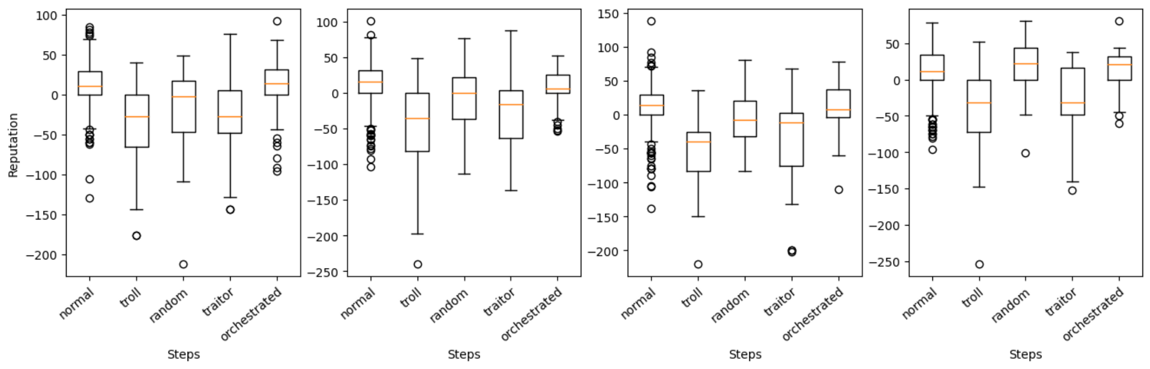


Figure 5.9: Final reputation distributions of user types, $n = 500, s = 1000, v = 5000$

the results for 500 users, 1000 stories, and 2000 votes, while Figure 5.10 and Figure 5.11 present the results for 1000 users, 2000 stories, and 10000 votes.

We provide the Precision, Recall, F1, and Accuracy metrics as well as the ratio of malicious detection of various attacks in Table 5.2. The results indicate that the system can identify the attacks successfully, but both detection and management of reputation for the orchestrated type attacks require more work. The malicious user types have less reputation overall than the non-malicious (normal) ones, and the system performs as described in the results under malicious behavior.

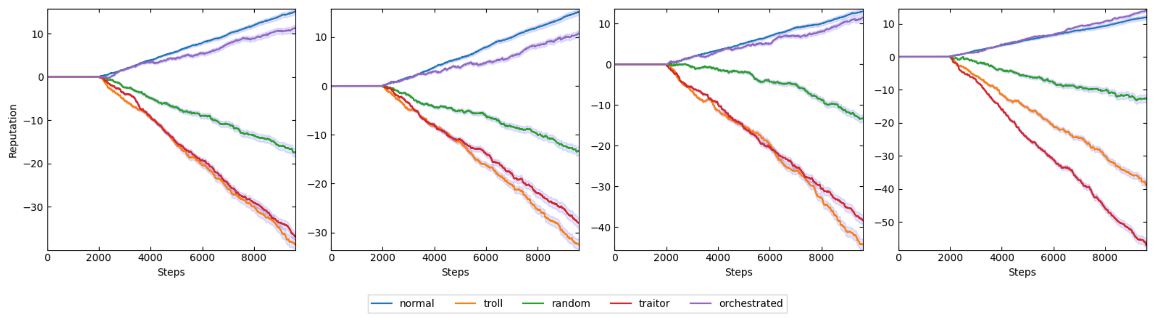


Figure 5.10: Reputation means of user types over time, $n = 1000, s = 2000, v = 10000$

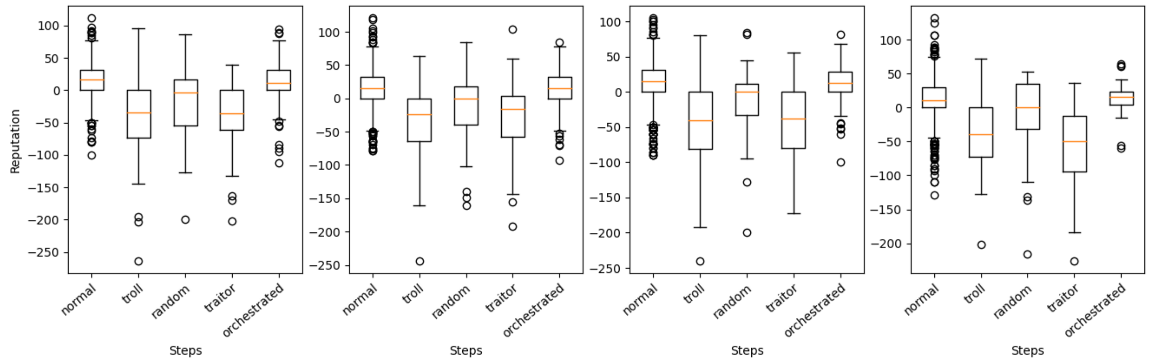


Figure 5.11: Final reputation distributions of user types, $n = 1000, s = 2000, v = 10000$

5.4.1 Twitter Case Study

Birdwatch is a new feature that is being developed by Twitter as a way to combat misinformation on the platform. The idea behind Birdwatch is to allow users to contribute to a community-driven approach to addressing misinformation. Rather than relying solely on Twitter's algorithms and moderators to identify and flag misinformation, Birdwatch allows users to identify and add notes to tweets that they believe are misleading or false. These notes will be visible to other users, providing additional context and information that can help users make more informed decisions about the content they see on Twitter.

Birdwatch is currently in the early stages of development and is being tested in a limited pilot program. It is not yet available to all users on the platform. Twitter has emphasized that the goal of Birdwatch is to create a system that

Table 5.2: Precision, Recall, F1, and Accuracy Metrics for 3 different user-story-vote (U-S-V) distribution: Dist 1: 100 users 200 stories 1000 votes; Dist 2: 500 users 1000 stories, 5000 votes; Dist 3: 1000 users, 2000 stories, 10000 votes. The user type distribution per U-S-V distribution: 42-12-12-12-12-targets, 50-10-10-10-10-targets, 62-7-7-7-7-targets, 70-5-5-5-5-targets.

	Metrics				Malicious Detection			
Run	Precision	Recall	F1	Accuracy	Troll	Random	Traitor	Orchestrated
U-S-V Dist. 1	0.81	0.68	0.74	0.68	0.96	0.96	0.85	0.33
	0.86	0.79	0.82	0.79	1.00	0.96	0.68	0.00
	0.89	0.83	0.86	0.83	0.82	0.8	0.83	1.00
	0.91	0.91	0.91	0.91	0.80	0.67	0.57	0.00
U-S-V Dist. 2	0.83	0.70	0.76	0.70	0.90	0.94	0.92	0.57
	0.85	0.74	0.79	0.75	0.89	0.93	0.83	0.18
	0.89	0.84	0.87	0.84	0.83	0.86	0.76	0.54
	0.92	0.90	0.91	0.90	0.91	0.88	0.82	0.10
U-S-V Dist. 3	0.84	0.70	0.76	0.70	0.90	0.84	0.88	0.42
	0.87	0.77	0.82	0.77	0.84	0.92	0.89	0.25
	0.89	0.84	0.86	0.84	0.81	0.85	0.85	0.50
	0.91	0.88	0.89	0.88	0.91	0.88	0.84	0.17

is transparent, open, and accountable, and that allows for a diverse range of perspectives and voices. The company has also stated that it will use a variety of mechanisms to prevent abuse and manipulation of the system. It is not yet clear when or how Birdwatch will be rolled out to the broader Twitter community.

In [192], the authors investigate Birdwatch and show that it is vulnerable to adversarial activity. They propose different metrics as criteria that need to be satisfied for the quality of tweets, users, and notes. They annotate a dataset of 500 tweets included in the Birdwatch data from January - April 2021 and show that their reputation system *HawkEye* outperforms Birdwatch in classifying tweets accurately.

To understand how the system described in our work performs against a human-curated dataset, we adopt the dataset provided in the study of Hawkeye. We treat notes on tweets and the ratings on notes as votes by counting agreeing ratings on notes as the same value as their corresponding note and disagreeing notes as the opposite value of the note. We observe that there exist 2.8% duplicates in the original data set and eliminate them, and perform the simulation on this new set. In this dataset, we are unable to identify malicious

Table 5.3: Comparison of Hawkeye and our system on the Twitter Birdwatch dataset

Metric	HawkEye *Supervised*	HawkEye *Unsupervised*	Our Work *Train*	Our Work *Test*
Precision	85	79	85	85
Recall	74	78	77	75
F1	76	78	78	77

actions as we are unsure of the intent of the users. We denote only the false-story posting as malicious. Notwithstanding, our system performs as accurately and in some cases better than Hawkeye, considering the duplicate elimination and the time-series approach in our work compared to Hawkeye, which used the entire dataset and cross-validation. The results are provided in Table 5.3. We give the results of the train and test sets, and list Hawkeye's results from their original paper. We employ 80%-20% train and test split, where the train set is the earliest portion of the dataset.

5.5 Discussion

In this section, we discuss some important aspects regarding the deployment and success of blockchain and machine learning-based solutions proposed to effectively deal with the misinformation problem.

First of all, the positioning of the blockchain is quite important. While the distributed nature of the blockchain provides fault tolerance, it is only possible for such a blockchain to be run without any specific owner only if there exists an incentive-based mechanism to run a blockchain node. Additionally, the blockchain should provide an API for the integration of existing social networks or news-providing services. Such a scenario would possibly include the creation of a user in the blockchain with a public key and a wallet, that is pointed by a user profile in an existing social network. Each of the stories posted on the social network would have to be inserted into the blockchain as well.

With users, stories, and related votes each having a representation in the blockchain would create possibilities to solve existing research problems as well since the existing network topologies would be flattened out in the blockchain, hinting at the bypassing of the computationally difficult problems.

For instance, if the blockchain should have the possibility of storing which user has seen a story, we would solve the problem of determining which users were affected by a post. We would simply filter the database, instead of using various algorithms to traverse the network. The blockchain would have such an API `AddUserStorySeen (userid, storyid)`, and the social network would call this API when a user visits a story.

It should also be noted that if the blockchain would be able to serve multiple social networks and news-sharing platforms, then this feature could be used to reduce the effect of the same information being copied and spread over different domains. In addition, we would have a better opportunity to identify the originality of the posts [193].

On the other hand, the identification and prevention of Sybil accounts and solving the multiple identities have not been solved in social networks without trading off the privacy issue. Solving this problem in a blockchain-based solution, such as the one in this work, is of utmost importance in addition to the anonymity problem.

One other issue that needs to be considered is to also have the machine learning approach distributed to solve the misinformation problem. Having a single authority host, the classifier would jeopardize the transparency we describe in this work. The technical solution would require hosting the model on different nodes and verifying model correctness.

In this work, we used two different classifiers. The main advantage of this is that we can teach the first classifier different attack forms and this would increase the explainability of the classifier compared to having a single box.

It should also be noted that we have implemented known attacks in a limited setting. As a future work, we consider using reinforcement learning to generate different attacks, analyze and try to encounter them.

5.6 Summary

In this chapter, we identified the critical factors in deception by referencing the Interpersonal Deception Theory and provided a model for misinformation in online social networks. We described how blockchains are one of the candidates that can provide a better solution in terms of transparency, decentralization, validity, and immutability. We described and implemented a crowdsourcing mechanism on the blockchain, specifically on Ethereum, using smart contracts towards identifying true and false stories. We added a deep learning system that took part in identifying the malicious actions in the crowdsourcing mechanism and the final decision on the truthfulness of stories. We trained and tested the system under various types of attacks and provided the results. We provided a case study on Twitter Birdwatch data and compared our results with another study on the same data set.

Chapter 6

Conclusion

The proliferation of information in our daily lives has highlighted the importance of accessing accurate and reliable information. However, the fluid nature of the internet makes it difficult to control the flow of information, allowing some individuals to disseminate false or misleading information quickly. Researchers have attempted to address this problem by developing methods to prevent the spread of misinformation.

In this study, we have contributed to understanding misinformation on social networks in three ways. We first developed a social network simulation framework called Crowd that allowed us to model, simulate, visualize and analyze various social network scenarios.

Secondly, we constructed a misinformation propagation game based on evolutionary game theory and observed that the game follows the characteristics of cooperative games. We then designed a game on the network level where network-level actors determine the actions of nodes. We proposed a deep reinforcement learning-based approach using the Multi-Agent Deep Deterministic Policy Gradients algorithm. Our results demonstrate that our methods are effective in defending against misinformation and outperform well-known node-selection algorithms on various social networks. We discussed the advantages, limitations, and future research directions in detail.

Solutions in the literature that aim to detect and mitigate misinformation, such as manual labeling of news by small groups of people or machine learning systems, may need improvement on the principles of transparency, immutability, validity, and decentralization and require large amounts of data. Blockchains can provide such principles by design as they are decentralized and transparent systems to facilitate crowd-sourcing. In the final part of our work, we proposed a simple way to score news articles alongside various attack scenarios. Our case study, including a comparison with a current study on Twitter, further supports the validity of our approach. After presenting our findings, we extensively discussed how this system could be utilized and its limitations, and listed possible research directions.

Bibliography

[1] N. Newman, R. Fletcher, A. Kalogeropoulos, D. Levy, and R. K. Nielsen, “Reuters institute digital news report 2017,” 2017.

[2] S. Flaxman, S. Goel, and J. M. Rao, “Filter bubbles, echo chambers, and online news consumption,” *Public opinion quarterly*, vol. 80, no. S1, pp. 298–320, 2016.

[3] H. Allcott and M. Gentzkow, “Social media and fake news in the 2016 election,” *Journal of Economic Perspectives*, vol. 31, pp. 211–36, May 2017.

[4] C. F. Greer and D. A. Ferguson, “Using twitter for promotion and branding: A content analysis of local television twitter sites,” *Journal of Broadcasting & Electronic Media*, vol. 55, no. 2, pp. 198–214, 2011.

[5] G. Pressgrove, B. W. McKeever, and S. M. Jang, “What is contagious? exploring why content goes viral on twitter: A case study of the als ice bucket challenge,” *International Journal of Nonprofit and Voluntary Sector Marketing*, vol. 23, no. 1, p. e1586, 2018.

[6] J. Yin, S. Karimi, A. Lampert, M. Cameron, B. Robinson, and R. Power, “Using social media to enhance emergency situation awareness,” in *Twenty-Fourth International Joint Conference on Artificial Intelligence*, IJCAI’15, p. 4234–4238, AAAI Press, 2015.

[7] K. Starbird and L. Palen, “Voluntweeters: Self-organizing by digital volunteers in times of crisis,” in *Proceedings of the SIGCHI conference on human factors in computing systems*, pp. 1071–1080, ACM, 2011.

[8] A. L. Schmidt, F. Zollo, M. Del Vicario, A. Bessi, A. Scala, G. Caldarelli, H. E. Stanley, and W. Quattrociocchi, "Anatomy of news consumption on facebook," *Proceedings of the National Academy of Sciences*, vol. 114, no. 12, pp. 3035–3039, 2017.

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

[10] O. D. Apuke and B. Omar, "Fake news and covid-19: modelling the predictors of fake news sharing among social media users," *Telematics and Informatics*, vol. 56, p. 101475, 2021.

[11] A. Gupta, H. Lamba, P. Kumaraguru, and A. Joshi, "Faking sandy: characterizing and identifying fake images on twitter during hurricane sandy," in *Proceedings of the 22nd international conference on World Wide Web*, pp. 729–736, 2013.

[12] S. S. Ho, A. S. F. Chuah, N. Kim, and E. C. Tandoc Jr., "Fake news, real risks: How online discussion and sources of fact-check influence public risk perceptions toward nuclear energy," *Risk Analysis*, vol. 42, no. 11, pp. 2569–2583, 2022.

[13] J. A. Piazza, "Fake news: the effects of social media disinformation on domestic terrorism," *Dynamics of Asymmetric Conflict*, vol. 15, no. 1, pp. 55–77, 2022.

[14] I. Khaldarova and M. Pantti, "Fake news: The narrative battle over the ukrainian conflict," in *The Future of Journalism: Risks, Threats and Opportunities*, pp. 228–238, Routledge, 2020.

[15] J. Shin and K. Thorson, "Partisan selective sharing: The biased diffusion of fact-checking messages on social media," *Journal of Communication*, vol. 67, no. 2, pp. 233–255, 2017.

[16] M. Babaei, J. Kulshrestha, A. Chakraborty, E. M. Redmiles, M. Cha, and K. P. Gummadi, "Analyzing biases in perception of truth in news stories and

their implications for fact checking," *IEEE Transactions on Computational Social Systems*, vol. 9, no. 3, pp. 839–850, 2022.

[17] T. Draws, D. La Barbera, M. Soprano, K. Roitero, D. Ceolin, A. Checco, and S. Mizzaro, "The effects of crowd worker biases in fact-checking tasks," in *2022 ACM Conference on Fairness, Accountability, and Transparency*, pp. 2114–2124, 2022.

[18] E. Thorson, "Belief echoes: The persistent effects of corrected misinformation," *Political Communication*, vol. 33, no. 3, pp. 460–480, 2016.

[19] Z. Guo, M. Schlichtkrull, and A. Vlachos, "A survey on automated fact-checking," *Transactions of the Association for Computational Linguistics*, vol. 10, pp. 178–206, 2022.

[20] N. Higdon, *The anatomy of fake news: A critical news literacy education.* University of California Press, 2020.

[21] S. Kumar and N. Shah, "False information on web and social media: A survey," *CoRR*, vol. abs/1804.08559, 2018.

[22] S. Vosoughi, D. Roy, and S. Aral, "The spread of true and false news online," *Science*, vol. 359, no. 6380, pp. 1146–1151, 2018.

[23] M. Del Vicario, A. Bessi, F. Zollo, F. Petroni, A. Scala, G. Caldarelli, H. E. Stanley, and W. Quattrociocchi, "The spreading of misinformation online," *Proceedings of the National Academy of Sciences*, vol. 113, no. 3, pp. 554–559, 2016.

[24] V. Qazvinian, E. Rosengren, D. R. Radev, and Q. Mei, "Rumor has it: Identifying misinformation in microblogs," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, (Stroudsburg, PA, USA), pp. 1589–1599, Association for Computational Linguistics, 2011.

[25] G. Shrivastava, P. Kumar, R. P. Ojha, P. K. Srivastava, S. Mohan, and G. Srivastava, "Defensive modeling of fake news through online social networks," *IEEE Transactions on Computational Social Systems*, vol. 7, no. 5, pp. 1159–1167, 2020.

[26] L. Li, Q. Zhang, X. Wang, J. Zhang, T. Wang, T.-L. Gao, W. Duan, K. K.-f. Tsoi, and F.-Y. Wang, "Characterizing the propagation of situational information in social media during covid-19 epidemic: A case study on weibo," *IEEE Transactions on computational social systems*, vol. 7, no. 2, pp. 556–562, 2020.

[27] L. Hu, S. Wei, Z. Zhao, and B. Wu, "Deep learning for fake news detection: A comprehensive survey," *AI Open*, vol. 3, pp. 133–155, 2022.

[28] X. Zhou and R. Zafarani, "A survey of fake news: Fundamental theories, detection methods, and opportunities," *ACM Computing Surveys (CSUR)*, vol. 53, no. 5, pp. 1–40, 2020.

[29] X. Zhang and A. A. Ghorbani, "An overview of online fake news: Characterization, detection, and discussion," *Information Processing & Management*, vol. 57, no. 2, p. 102025, 2020.

[30] S. Mohseni, E. Ragan, and X. Hu, "Open issues in combating fake news: Interpretability as an opportunity," *arXiv preprint arXiv:1904.03016*, 2019.

[31] H. Seo, A. Xiong, and D. Lee, "Trust it or not: Effects of machine-learning warnings in helping individuals mitigate misinformation," in *Proceedings of the 10th ACM Conference on Web Science*, pp. 265–274, 2019.

[32] A. T. Nguyen, A. Kharosekar, S. Krishnan, S. Krishnan, E. Tate, B. C. Wallace, and M. Lease, "Believe it or not: Designing a human-ai partnership for mixed-initiative fact-checking," in *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*, pp. 189–199, 2018.

[33] N. Wingfield, M. Isaac, and K. Benner, "Google and facebook take aim at fake news sites," *The New York Times*, vol. 11, p. 12, 2016.

[34] N. Bartley, A. Abeliuk, E. Ferrara, and K. Lerman, "Auditing algorithmic bias on twitter," in *13th ACM Web Science Conference 2021*, pp. 65–73, 2021.

[35] J. Bandy and N. Diakopoulos, "Curating quality? how twitter's timeline algorithm treats different types of news," *Social Media+ Society*, vol. 7, no. 3, p. 20563051211041648, 2021.

[36] G. Szabó and G. Fath, "Evolutionary games on graphs," *Physics reports*, vol. 446, no. 4, pp. 97–216, 2007.

[37] E. Lieberman, C. Hauert, and M. A. Nowak, "Evolutionary dynamics on graphs," *Nature*, vol. 433, no. 7023, pp. 312–316, 2005.

[38] H. Ohtsuki, C. Hauert, E. Lieberman, and M. A. Nowak, "A simple rule for the evolution of cooperation on graphs and social networks," *Nature*, vol. 441, no. 7092, pp. 502–505, 2006.

[39] R. Fletcher and S. Park, "The impact of trust in the news media on online news consumption and participation," *Digital Journalism*, vol. 5, no. 10, pp. 1281–1299, 2017.

[40] K. Hoffman, D. Zage, and C. Nita-Rotaru, "A survey of attack and defense techniques for reputation systems," *ACM Comput. Surv.*, vol. 42, pp. 1:1–1:31, Dec. 2009.

[41] D. Wang, T. Muller, Y. Liu, and J. Zhang, "Towards robust and effective trust management for security: A survey," in *2014 IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications*, pp. 511–518, Sep. 2014.

[42] Y. Ruan and A. Durresi, "A survey of trust management systems for online social communities–trust modeling, trust inference and attacks," *Knowledge-Based Systems*, vol. 106, pp. 150–163, 2016.

[43] F. Hendrikx, K. Bubendorfer, and R. Chard, "Reputation systems: A survey and taxonomy," *Journal of Parallel and Distributed Computing*, vol. 75, pp. 184–197, 2015.

[44] V. L. Rubin, Y. Chen, and N. J. Conroy, "Deception detection for news: Three types of fakes," in *Proceedings of the 78th ASIS&T Annual Meeting: Information Science with Impact: Research in and for the Community*, ASIST '15, (Silver Springs, MD, USA), pp. 83:1–83:4, American Society for Information Science, 2015.

[45] L. Wu, F. Morstatter, K. M. Carley, and H. Liu, "Misinformation in social media: definition, manipulation, and detection," *ACM SIGKDD Explorations Newsletter*, vol. 21, no. 2, pp. 80–90, 2019.

[46] S. Zannettou, M. Sirivianos, J. Blackburn, and N. Kourtellis, "The web of false information: Rumors, fake news, hoaxes, clickbait, and various other shenanigans," *Journal of Data and Information Quality (JDIQ)*, vol. 11, no. 3, pp. 1–37, 2019.

[47] K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu, "Fake news detection on social media: A data mining perspective," *SIGKDD Explor. Newsl.*, vol. 19, pp. 22–36, Sept. 2017.

[48] N. J. Conroy, V. L. Rubin, and Y. Chen, "Automatic deception detection: Methods for finding fake news," *Proceedings of the Association for Information Science and Technology*, vol. 52, no. 1, pp. 1–4, 2015.

[49] A. Gupta, H. Lamba, P. Kumaraguru, and A. Joshi, "Faking sandy: Characterizing and identifying fake images on twitter during hurricane sandy," in *Proceedings of the 22Nd International Conference on World Wide Web*, WWW '13 Companion, (New York, NY, USA), pp. 729–736, ACM, 2013.

[50] F. Yang, Y. Liu, X. Yu, and M. Yang, "Automatic detection of rumor on sina weibo," in *Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics*, MDS '12, (New York, NY, USA), pp. 13:1–13:7, ACM, 2012.

[51] V. Pérez-Rosas, B. Kleinberg, A. Lefevre, and R. Mihalcea, "Automatic detection of fake news," *CoRR*, vol. abs/1708.07104, 2017.

[52] S. Volkova, K. Shaffer, J. Y. Jang, and N. Hodas, "Separating facts from fiction: Linguistic models to classify suspicious and trusted news posts on twitter," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, (Vancouver, Canada), pp. 647–653, Association for Computational Linguistics, July 2017.

[53] S. Kwon, M. Cha, and K. Jung, "Rumor detection over varying time windows," *PloS one*, vol. 12, no. 1, p. e0168344, 2017.

[54] V. L. Rubin and T. Lukoianova, "Truth and deception at the rhetorical structure level," *Journal of the Association for Information Science and Technology*, vol. 66, no. 5, pp. 905–917, 2015.

[55] V. L. Rubin, N. J. Conroy, and Y. Chen, "Towards news verification: Deception detection methods for news discourse," in *Hawaii International Conference on System Sciences*, 2015.

[56] Z. Jin, J. Cao, Y. Zhang, and J. Luo, "News verification by exploiting conflicting social viewpoints in microblogs," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, p. 2972–2978, AAAI Press, 2016.

[57] N. Ruchansky, S. Seo, and Y. Liu, "Csi: A hybrid deep model for fake news detection," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, CIKM '17, (New York, NY, USA), pp. 797–806, ACM, 2017.

[58] L. Wu and H. Liu, "Tracing fake-news footprints: Characterizing social media messages by how they propagate," in *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, WSDM '18, (New York, NY, USA), pp. 637–645, ACM, 2018.

[59] J. Ma, W. Gao, and K.-F. Wong, "Detect rumors in microblog posts using propagation structure via kernel learning," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1:*

Long Papers), (Vancouver, Canada), pp. 708–717, Association for Computational Linguistics, Association for Computational Linguistics, July 2017.

[60] E. Ferrara, O. Varol, C. Davis, F. Menczer, and A. Flammini, "The rise of social bots," *Commun. ACM*, vol. 59, pp. 96–104, June 2016.

[61] Q. Cao, M. Sirivianos, X. Yang, and T. Pregueiro, "Aiding the detection of fake accounts in large scale social online services," in *Proceedings of the 9th USENIX Conference on Networked Systems Design and Implementation*, NSDI'12, (Berkeley, CA, USA), pp. 15–15, USENIX Association, 2012.

[62] G. Wang, M. Mohanlal, C. Wilson, X. Wang, M. J. Metzger, H. Zheng, and B. Y. Zhao, "Social turing tests: Crowdsourcing sybil detection," *CoRR*, vol. abs/1205.3856, 2012.

[63] A. Ghosh, S. Kale, and P. McAfee, "Who moderates the moderators?: Crowdsourcing abuse detection in user-generated content," in *Proceedings of the 12th ACM Conference on Electronic Commerce*, EC '11, (New York, NY, USA), pp. 167–176, ACM, 2011.

[64] O. Varol, E. Ferrara, C. Davis, F. Menczer, and A. Flammini, "Online human-bot interactions: Detection, estimation, and characterization," in *Proceedings of the international AAAI conference on web and social media(ICWSM)*, pp. 280–289, 2017.

[65] G. Wang, T. Konolige, C. Wilson, X. Wang, H. Zheng, and B. Y. Zhao, "You are how you click: Clickstream analysis for sybil detection," in *Presented as part of the 22nd {USENIX} Security Symposium ({USENIX} Security 13)*, pp. 241–256, 2013.

[66] S. Cresci, "A decade of social bot detection," *Communications of the ACM*, vol. 63, no. 10, pp. 72–83, 2020.

[67] M. Latah, "Detection of malicious social bots: A survey and a refined taxonomy," *Expert Systems with Applications*, vol. 151, p. 113383, 2020.

[68] A. Alharbi, H. Dong, X. Yi, Z. Tari, and I. Khalil, “Social media identity deception detection: A survey,” *ACM Computing Surveys (CSUR)*, vol. 54, no. 3, pp. 1–35, 2021.

[69] C. Budak, D. Agrawal, and A. El Abbadi, “Limiting the spread of misinformation in social networks,” in *Proceedings of the 20th International Conference on World Wide Web*, WWW ’11, (New York, NY, USA), pp. 665–674, ACM, 2011.

[70] A. Gupta, P. Kumaraguru, C. Castillo, and P. Meier, “Tweetcred: Real-time credibility assessment of content on twitter,” in *International Conference on Social Informatics*, pp. 228–243, Springer, 2014.

[71] N. P. Nguyen, G. Yan, M. T. Thai, and S. Eidenbenz, “Containment of misinformation spread in online social networks,” in *Proceedings of the 4th Annual ACM Web Science Conference*, WebSci ’12, (New York, NY, USA), pp. 213–222, ACM, 2012.

[72] N. P. Nguyen, G. Yan, and M. T. Thai, “Analysis of misinformation containment in online social networks,” *Computer Networks*, vol. 57, no. 10, pp. 2133–2146, 2013. Towards a Science of Cyber Security Security and Identity Architecture for the Future Internet.

[73] D. V. Pham, G. L. Nguyen, T. N. Nguyen, C. V. Pham, and A. V. Nguyen, “Multi-topic misinformation blocking with budget constraint on online social networks,” *IEEE Access*, vol. 8, pp. 78879–78889, 2020.

[74] Z. He, Z. Cai, J. Yu, X. Wang, Y. Sun, and Y. Li, “Cost-efficient strategies for restraining rumor spreading in mobile social networks,” *IEEE Transactions on Vehicular Technology*, vol. 66, no. 3, pp. 2789–2800, 2016.

[75] M. Farajtabar, J. Yang, X. Ye, H. Xu, R. Trivedi, E. Khalil, S. Li, L. Song, and H. Zha, “Fake news mitigation via point process based intervention,” in *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML’17, pp. 1097–1106, PMLR, JMLR.org, 2017.

[76] A. I. E. Hosni and K. Li, “Minimizing the influence of rumors during breaking news events in online social networks,” *Knowledge-Based Systems*, vol. 193, p. 105452, 2020.

[77] Y. Li, H. Gao, Y. Gao, J. Guo, and W. Wu, “A survey on influence maximization: From an ml-based combinatorial optimization,” *arXiv preprint arXiv:2211.03074*, 2022.

[78] C. Jiang, Y. Chen, and K. R. Liu, “Graphical evolutionary game for information diffusion over social networks,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 4, pp. 524–536, 2014.

[79] X. Yang, Z. Zhu, H. Yu, Y. Zhao, and L. Guo, “Evolutionary game dynamics of the competitive information propagation on social networks,” *Complexity*, vol. 2019, 2019.

[80] K. K. Kumar and G. Geethakumari, “Information diffusion model for spread of misinformation in online social networks,” in *2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 1172–1177, IEEE, 2013.

[81] D. Li, J. Ma, Z. Tian, and H. Zhu, “An evolutionary game for the diffusion of rumor in complex networks,” *Physica A: Statistical Mechanics and its Applications*, vol. 433, pp. 51–58, 2015.

[82] Y. Xiao, D. Chen, S. Wei, Q. Li, H. Wang, and M. Xu, “Rumor propagation dynamic model based on evolutionary game and anti-rumor,” *Nonlinear Dynamics*, vol. 95, no. 1, pp. 523–539, 2019.

[83] M. Askarizadeh, B. T. Ladani, and M. H. Manshaei, “An evolutionary game model for analysis of rumor propagation and control in social networks,” *Physica A: statistical mechanics and its applications*, vol. 523, pp. 21–39, 2019.

[84] M. Askarizadeh and B. T. Ladani, “Soft rumor control in social networks: Modeling and analysis,” *Engineering Applications of Artificial Intelligence*, vol. 100, p. 104198, 2021.

[85] A. Kittur, E. H. Chi, and B. Suh, "Crowdsourcing user studies with mechanical turk," in *Proceedings of the SIGCHI conference on human factors in computing systems*, pp. 453–456, 2008.

[86] E. Tacchini, G. Ballarin, M. L. D. Vedova, S. Moret, and L. de Alfaro, "Some like it hoax: Automated fake news detection in social networks," *CoRR*, vol. abs/1704.07506, 2017.

[87] J. Kim, B. Tabibian, A. Oh, B. Schölkopf, and M. Gomez-Rodriguez, "Leveraging the crowd to detect and reduce the spread of fake news and misinformation," in *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, WSDM '18, (New York, NY, USA), pp. 324–332, ACM, 2018.

[88] C.-C. Chen, Y. Du, R. Peter, and W. Golab, "An implementation of fake news prevention by blockchain and entropy-based incentive mechanism," *Social Network Analysis and Mining*, vol. 12, no. 1, pp. 1–21, 2022.

[89] M. Avelino and A. A. d. A. Rocha, "Blockproof: A framework for verifying authenticity and integrity of web content," *Sensors*, vol. 22, no. 3, p. 1165, 2022.

[90] E. Sengupta, R. Nagpal, D. Mehrotra, and G. Srivastava, "Problock: a novel approach for fake news detection," *Cluster Computing*, vol. 24, no. 4, pp. 3779–3795, 2021.

[91] G. Pennycook and D. G. Rand, "Fighting misinformation on social media using crowdsourced judgments of news source quality," *Proceedings of the National Academy of Sciences*, vol. 116, no. 7, pp. 2521–2526, 2019.

[92] R. Denaux, F. Merenda, and J. M. Gómez-Pérez, "Towards crowdsourcing tasks for accurate misinformation detection.," in *ASLD@ ISWC*, pp. 159–167, 2020.

[93] M. Soprano, K. Roitero, D. La Barbera, D. Ceolin, D. Spina, S. Mizzaro, and G. Demartini, "The many dimensions of truthfulness: Crowdsourcing misinformation assessments on a multidimensional scale," *Information Processing & Management*, vol. 58, no. 6, p. 102710, 2021.

[94] L. Teixeira, I. Amorim, A. U. Silva, J. C. Lopes, and V. Filipe, "A new approach to crowd journalism using a blockchain-based infrastructure," in *Proceedings of the 18th International Conference on Advances in Mobile Computing & Multimedia*, pp. 170–178, 2020.

[95] G. Shan, B. Zhao, J. R. Clavin, H. Zhang, and S. Duan, "Poligraph: Intrusion-tolerant and distributed fake news detection system," *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 28–41, 2021.

[96] T. H. Y. Zen, C. B. Hong, P. M. Mohan, and V. Balachandran, "Abc-verify: Ai-blockchain integrated framework for tweet misinformation detection," in *2021 IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI)*, pp. 1–5, IEEE, 2021.

[97] S. Tisue and U. Wilensky, "Netlogo: A simple environment for modeling complexity," in *International conference on complex systems*, vol. 21, pp. 16–21, Boston, MA, 2004.

[98] M. J. North, N. T. Collier, J. Ozik, E. R. Tatara, C. M. Macal, M. Bragen, and P. Sydelko, "Complex adaptive systems modeling with repast simphony," *Complex adaptive systems modeling*, vol. 1, no. 1, pp. 1–26, 2013.

[99] J. M. Sánchez, C. A. Iglesias, and J. F. Sánchez-Rada, "Soil: An agent-based social simulator in python for modelling and simulation of social networks," in *International Conference on Practical Applications of Agents and Multi-Agent Systems*, pp. 234–245, Springer, 2017.

[100] K. Ryczko, A. Domurad, N. Buhagiar, and I. Tamblyn, "Hashkat: large-scale simulations of online social networks," *Social Network Analysis and Mining*, vol. 7, no. 1, pp. 1–13, 2017.

[101] A. Jøsang, R. Ismail, and C. Boyd, "A survey of trust and reputation systems for online service provision," *Decision Support Systems*, vol. 43, no. 2, pp. 618–644, 2007. Emerging Issues in Collaborative Commerce.

[102] E. Koutrouli and A. Tsalgatidou, "Taxonomy of attacks and defense mechanisms in p2p reputation systems—lessons for reputation system designers," *Computer Science Review*, vol. 6, no. 2, pp. 47–70, 2012.

[103] G. Diego, "Can we trust trust?," *Trust: Making and breaking cooperative relations*, vol. 213, p. 214, 1988.

[104] M. Tavakolifard, K. C. Almeroth, and J. A. Gulla, "Does social contact matter?: Modelling the hidden web of trust underlying twitter," in *Proceedings of the 22Nd International Conference on World Wide Web*, WWW '13 Companion, (New York, NY, USA), pp. 981–988, ACM, 2013.

[105] Q. Shambour and J. Lu, "A hybrid trust-enhanced collaborative filtering recommendation approach for personalized government-to-business e-services," *International Journal of Intelligent Systems*, vol. 26, no. 9, pp. 814–843, 2011.

[106] Q. Shambour and J. Lu, "A trust-semantic fusion-based recommendation approach for e-business applications," *Decision Support Systems*, vol. 54, no. 1, pp. 768–780, 2012.

[107] T. Bhuiyan, *Trust for intelligent recommendation.* Springer, 2013.

[108] J. Golbeck, "Trust and nuanced profile similarity in online social networks," *ACM Trans. Web*, vol. 3, pp. 12:1–12:33, Sept. 2009.

[109] R. Falcone, G. Pezzulo, and C. Castelfranchi, "A fuzzy approach to a belief-based trust computation," in *Trust, Reputation, and Security: Theories and Practice* (R. Falcone, S. Barber, L. Korba, and M. Singh, eds.), (Berlin, Heidelberg), pp. 73–86, Springer Berlin Heidelberg, 2003.

[110] V. Kant and K. K. Bharadwaj, "Fuzzy computational models of trust and distrust for enhanced recommendations," *International Journal of Intelligent Systems*, vol. 28, no. 4, pp. 332–365, 2013.

[111] K. W. Nafi, T. S. Kar, M. A. Hossain, and M. M. A. Hashem, "A fuzzy logic based certain trust model for e-commerce," in *2013 International Conference on Informatics, Electronics and Vision (ICIEV)*, pp. 1–6, May 2013.

[112] X. Liu, A. Datta, and E.-P. Lim, *Computational trust models and machine learning*. Chapman and Hall/CRC, 2014.

[113] J. Zhan and X. Fang, “A novel trust computing system for social networks,” in *2011 IEEE Third International Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third International Conference on Social Computing*, pp. 1284–1289, Oct 2011.

[114] L. Mui, M. Mohtashemi, and A. Halberstadt, “A computational model of trust and reputation,” in *Proceedings of the 35th Annual Hawaii International Conference on System Sciences*, pp. 2431–2439, Jan 2002.

[115] J. Sabater and C. Sierra, “Social regret, a reputation model based on social relations,” *SIGecom Exch.*, vol. 3, pp. 44–56, Dec. 2001.

[116] J. Sabater and C. Sierra, “Regret: Reputation in gregarious societies,” in *Proceedings of the Fifth International Conference on Autonomous Agents*, AGENTS ’01, (New York, NY, USA), pp. 194–195, ACM, 2001.

[117] S. Buchegger and J.-Y. Le Boudec, “Performance analysis of the confidant protocol,” in *Proceedings of the 3rd ACM International Symposium on Mobile Ad Hoc Networking &Amp; Computing*, MobiHoc ’02, (New York, NY, USA), pp. 226–236, ACM, 2002.

[118] E. Damiani, D. C. di Vimercati, S. Paraboschi, P. Samarati, and F. Violante, “A reputation-based approach for choosing reliable resources in peer-to-peer networks,” in *Proceedings of the 9th ACM Conference on Computer and Communications Security*, CCS ’02, (New York, NY, USA), pp. 207–216, ACM, 2002.

[119] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, “The eigentrust algorithm for reputation management in p2p networks,” in *Proceedings of the 12th International Conference on World Wide Web*, WWW ’03, (New York, NY, USA), pp. 640–651, ACM, 2003.

[120] K. Aberer and Z. Despotovic, "Managing trust in a peer-2-peer information system," in *Proceedings of the Tenth International Conference on Information and Knowledge Management*, CIKM '01, (New York, NY, USA), pp. 310–317, ACM, 2001.

[121] L. Xiong and L. Liu, "Peertrust: Supporting reputation-based trust for peer-to-peer electronic communities," *IEEE transactions on Knowledge and Data Engineering*, vol. 16, no. 7, pp. 843–857, 2004.

[122] Z. Malik and A. Bouguettaya, "Rateweb: Reputation assessment for trust establishment among web services," *The VLDB Journal*, vol. 18, pp. 885–911, Aug 2009.

[123] C. Tian and B. Yang, "R2trust, a reputation and risk based trust management framework for large-scale, fully decentralized overlay networks," *Future Generation Computer Systems*, vol. 27, no. 8, pp. 1135–1141, 2011.

[124] F. Hendrikx and K. Bubendorfer, "Malleable access rights to establish and enable scientific collaboration," in *2013 IEEE 9th International Conference on e-Science*, pp. 334–341, Oct 2013.

[125] K. Chen, H. Shen, K. Sapra, and G. Liu, "A social network based reputation system for cooperative p2p file sharing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 8, pp. 2140–2153, 2014.

[126] E. Bellini, Y. Iraqi, and E. Damiani, "Blockchain-based distributed trust and reputation management systems: A survey," *IEEE Access*, vol. 8, pp. 21127–21151, 2020.

[127] O. Hasan, L. Brunie, and E. Bertino, "Privacy-preserving reputation systems based on blockchain and other cryptographic building blocks: A survey," *ACM Computing Surveys (CSUR)*, vol. 55, no. 2, pp. 1–37, 2022.

[128] L. Liu and M. Munro, "Systematic analysis of centralized online reputation systems," *Decision Support Systems*, vol. 52, no. 2, pp. 438–449, 2012.

[129] P. Resnick and R. Zeckhauser, "Trust among strangers in internet transactions: Empirical analysis of ebay's reputation system," in *The Economics*

of the Internet and E-commerce, pp. 127–157, Emerald Group Publishing Limited, 2002.

[130] H. Liu, E.-P. Lim, H. W. Lauw, M.-T. Le, A. Sun, J. Srivastava, and Y. A. Kim, "Predicting trusts among users of online communities: an epinions case study," in *Proceedings of the 9th ACM Conference on Electronic Commerce*, pp. 310–319, 2008.

[131] N. Poor, "Mechanisms of an online public sphere: The website slashdot," *Journal of computer-mediated communication*, vol. 10, no. 2, p. JCMC1028, 2005.

[132] D. Movshovitz-Attias, Y. Movshovitz-Attias, P. Steenkiste, and C. Faloutsos, "Analysis of the reputation system and user contributions on a question answering website: Stackoverflow," in *2013 IEEE/ACM international conference on advances in social networks analysis and mining (ASONAM 2013)*, pp. 886–893, IEEE, 2013.

[133] L. C. Irani and M. S. Silberman, "Turkopticon: Interrupting worker invisibility in amazon mechanical turk," in *Proceedings of the SIGCHI conference on human factors in computing systems*, pp. 611–620, 2013.

[134] P. Van Mieghem, "Human psychology of common appraisal: The reddit score," *IEEE Transactions on Multimedia*, vol. 13, no. 6, pp. 1404–1406, 2011.

[135] M. Basili and M. A. Rossi, "Platform-mediated reputation systems in the sharing economy and incentives to provide service quality: the case of ridesharing services," *Electronic Commerce Research and Applications*, vol. 39, p. 100835, 2020.

[136] G. Zervas, D. Proserpio, and J. W. Byers, "A first look at online reputation on airbnb, where every stay is above average," *Marketing Letters*, vol. 32, no. 1, pp. 1–16, 2021.

[137] M. Feldman, K. Lai, I. Stoica, and J. Chuang, "Robust incentive techniques for peer-to-peer networks," in *Proceedings of the 5th ACM Conference on*

Electronic Commerce, EC '04, (New York, NY, USA), pp. 102–111, ACM, 2004.

[138] S. Marti and H. Garcia-Molina, "Taxonomy of trust: Categorizing p2p reputation systems," *Computer Networks*, vol. 50, no. 4, pp. 472–484, 2006. Management in Peer-to-Peer Systems.

[139] M. Srivatsa, L. Xiong, and L. Liu, "Trustguard: Countering vulnerabilities in reputation management for decentralized overlay networks," in *Proceedings of the 14th International Conference on World Wide Web*, WWW '05, (New York, NY, USA), pp. 422–431, ACM, 2005.

[140] Y. L. Sun, Z. Han, W. Yu, and K. R. Liu, "A trust evaluation framework in distributed networks: Vulnerability analysis and defense against attacks," in *Proceedings IEEE INFOCOM 2006. 25TH IEEE International Conference on Computer Communications*, pp. 1–13, IEEE, 2006.

[141] S. Liu, J. Zhang, C. Miao, Y.-L. Theng, and A. C. Kot, "iclub: an integrated clustering-based approach to improve the robustness of reputation systems," in *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 3*, pp. 1151–1152, International Foundation for Autonomous Agents and Multiagent Systems, 2011.

[142] A. A. Irissappane, S. Jiang, and J. Zhang, "A biclustering-based approach to filter dishonest advisors in multi-criteria e-marketplaces," in *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*, pp. 1385–1386, International Foundation for Autonomous Agents and Multiagent Systems, 2014.

[143] J. Zhang, R. Cohen, and K. Larson, "Combining trust modeling and mechanism design for promoting honesty in e-marketplaces," *Computational Intelligence*, vol. 28, no. 4, pp. 549–578, 2012.

[144] A. Jøsang and J. Golbeck, "Challenges for robust trust and reputation systems," in *Proceedings of the 5th International Workshop on Security and Trust Management (SMT 2009), Saint Malo, France*, p. 52, Citeseer, 2009.

[145] Y. Sun and Y. Liu, "Security of online reputation systems: The evolution of attacks and defenses," *IEEE Signal Processing Magazine*, vol. 29, no. 2, pp. 87–97, 2012.

[146] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Decentralized Business Review*, p. 21260, 2008.

[147] A. J. Menezes, S. A. Vanstone, and P. C. V. Oorschot, *Handbook of Applied Cryptography.* Boca Raton, FL, USA: CRC Press, Inc., 1st ed., 1996.

[148] V. Buterin *et al.*, "A next-generation smart contract and decentralized application platform," *white paper*, vol. 3, no. 37, pp. 2–1, 2014.

[149] W. Zhao, "Historic korean peace declaration recorded on ethereum blockchain." `https://www.coindesk.com/historic-korean-peace-declaration-recorded-on-ethereum-blockchain/`. Accessed: 2018-05-05.

[150] T. Yilmaz and Ö. Ulusoy, "Misinformation propagation in online social networks: Game theoretic and reinforcement learning approaches," *IEEE Transactions on Computational Social Systems*, 2022.

[151] M. Doebeli and C. Hauert, "Models of cooperation based on the prisoner's dilemma and the snowdrift game," *Ecology letters*, vol. 8, no. 7, pp. 748–766, 2005.

[152] M. A. Nowak, "Five rules for the evolution of cooperation," *science*, vol. 314, no. 5805, pp. 1560–1563, 2006.

[153] V. P. Crawford, "Lying for strategic advantage: Rational and boundedly rational misrepresentation of intentions," *American Economic Review*, vol. 93, no. 1, pp. 133–149, 2003.

[154] M. Zimdars and K. McLeod, *Fake news: understanding media and misinformation in the digital age.* MIT Press, 2020.

[155] V. Almeida, F. Filgueiras, and F. Gaetani, "Digital governance and the tragedy of the commons," *IEEE Internet Computing*, vol. 24, no. 4, pp. 41–46, 2020.

[156] M. Glenski, T. Weninger, and S. Volkova, “Propagation from deceptive news sources who shares, how much, how evenly, and how quickly?,” *IEEE Transactions on Computational Social Systems*, vol. 5, no. 4, pp. 1071–1082, 2018.

[157] J. Smith and G. R. Price, “The logic of animal conflict,” *Nature*, vol. 246, no. 5427, pp. 15–18, 1973.

[158] H. Ohtsuki and M. A. Nowak, “Direct reciprocity on graphs,” *Journal of theoretical biology*, vol. 247, no. 3, pp. 462–470, 2007.

[159] W. O. Kermack and A. G. McKendrick, “A contribution to the mathematical theory of epidemics,” *Proceedings of the royal society of london. Series A, Containing papers of a mathematical and physical character*, vol. 115, no. 772, pp. 700–721, 1927.

[160] J. McAuley and J. Leskovec, “Learning to discover social circles in ego networks,” in *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, vol. 25 of *NIPS’12*, (USA), pp. 539–547, Curran Associates Inc., 2012.

[161] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, “Fast unfolding of communities in large networks,” *Journal of statistical mechanics: theory and experiment*, vol. 2008, no. 10, p. P10008, 2008.

[162] U. N. Raghavan, R. Albert, and S. Kumara, “Near linear time algorithm to detect community structures in large-scale networks,” *Physical review E*, vol. 76, no. 3, p. 036106, 2007.

[163] E. D. Raj, G. Manogaran, G. Srivastava, and Y. Wu, “Information granulation-based community detection for social networks,” *IEEE Transactions on Computational Social Systems*, vol. 8, no. 1, pp. 122–133, 2020.

[164] D. Madeo and C. Mocenni, “Game interactions and dynamics on networked populations,” *IEEE Transactions on Automatic Control*, vol. 60, no. 7, pp. 1801–1810, 2014.

[165] G. Como, F. Fagnani, and L. Zino, "Imitation dynamics in population games on community networks," *IEEE Transactions on Control of Network Systems*, vol. 8, no. 1, pp. 65–76, 2020.

[166] S. Rao, A. K. Verma, and T. Bhatia, "A review on social spam detection: Challenges, open issues, and future directions," *Expert Systems with Applications*, vol. 186, p. 115742, 2021.

[167] P. K. Roy and S. Chahar, "Fake profile detection on social networking websites: a comprehensive review," *IEEE Transactions on Artificial Intelligence*, vol. 1, no. 3, pp. 271–285, 2020.

[168] D. Kempe, J. Kleinberg, and É. Tardos, "Maximizing the spread of influence through a social network," in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 137–146, 2003.

[169] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance, "Cost-effective outbreak detection in networks," in *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 420–429, 2007.

[170] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, *et al.*, "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[171] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," in *Proceedings of the 12th International Conference on Neural Information Processing Systems*, NIPS'99, (Cambridge, MA, USA), p. 1057–1063, MIT Press, 1999.

[172] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," in *International conference on machine learning*, pp. 387–395, PMLR, 2014.

[173] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

[174] A. S. Bedi, S. Chakraborty, A. Parayil, B. M. Sadler, P. Tokekar, and A. Koppel, "On the hidden biases of policy mirror ascent in continuous action spaces," in *International Conference on Machine Learning*, pp. 1716–1731, PMLR, 2022.

[175] S. M. Kakade, *On the sample complexity of reinforcement learning*. University of London, University College London (United Kingdom), 2003.

[176] H. Kumar, A. Koppel, and A. Ribeiro, "On the sample complexity of actor-critic method for reinforcement learning with function approximation," *arXiv preprint arXiv:1910.08412*, 2019.

[177] G. Matheron, N. Perrin, and O. Sigaud, "The problem with ddpg: understanding failures in deterministic environments with sparse rewards," *arXiv preprint arXiv:1911.11679*, 2019.

[178] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, (Red Hook, NY, USA), p. 6382–6393, Curran Associates Inc., 2017.

[179] J. McAuley and J. Leskovec, "Learning to discover social circles in ego networks," in *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, (Red Hook, NY, USA), p. 539–547, Curran Associates Inc., 2012.

[180] A. L. Traud, P. J. Mucha, and M. A. Porter, "Social structure of Facebook networks," *Phys. A*, vol. 391, pp. 4165–4180, Aug 2012.

[181] M. Richardson, R. Agrawal, and P. Domingos, "Trust management for the semantic web," in *International semantic Web conference*, pp. 351–368, Springer, 2003.

[182] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 855–864, 2016.

[183] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 701–710, 2014.

[184] S. Ivanov and E. Burnaev, "Anonymous walk embeddings," in *International conference on machine learning*, pp. 2186–2195, PMLR, 2018.

[185] D. B. Buller and J. K. Burgoon, "Interpersonal deception theory," *Communication theory*, vol. 6, no. 3, pp. 203–242, 1996.

[186] C. F. Bond Jr and B. M. DePaulo, "Accuracy of deception judgments," *Personality and social psychology Review*, vol. 10, no. 3, pp. 214–234, 2006.

[187] X. Zhou, K. Shu, V. V. Phoha, H. Liu, and R. Zafarani, ""this is fake! shared it by mistake": Assessing the intent of fake news spreaders," in *Proceedings of the ACM Web Conference 2022*, pp. 3685–3694, 2022.

[188] S. R. Sahoo and B. B. Gupta, "Multiple features based approach for automatic fake news detection on social networks using deep learning," *Applied Soft Computing*, vol. 100, p. 106983, 2021.

[189] J. N. Cohen, "Exploring echo-systems: how algorithms shape immersive media environments.," *Journal of Media Literacy Education*, vol. 10, no. 2, pp. 139–151, 2018.

[190] E. Ferrara, O. Varol, C. Davis, F. Menczer, and A. Flammini, "The rise of social bots," *Communications of the ACM*, vol. 59, no. 7, pp. 96–104, 2016.

[191] B. Ross, L. Pilz, B. Cabrera, F. Brachten, G. Neubaum, and S. Stieglitz, "Are social bots a real threat? an agent-based model of the spiral of silence to analyse the impact of manipulative actors in social networks," *European Journal of Information Systems*, vol. 28, no. 4, pp. 394–412, 2019.

[192] R. Mujumdar and S. Kumar, “Hawkeye: A robust reputation system for community-based counter-misinformation,” in *Proceedings of the 2021 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, ASONAM ’21, (New York, NY, USA), p. 188–192, Association for Computing Machinery, 2022.

[193] S. Huckle and M. White, “Fake news: A technological approach to proving the origins of content, using blockchains,” *Big data*, vol. 5, no. 4, pp. 356–371, 2017.