Understanding and Combating Online Social Deception

Zhen Guo

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Computer Science and Applications

Jin-Hee Cho, Chair
Ing-Ray Chen
Chang-Tien Lu
Kurt Luther
Srijan Sengupta

April 6, 2023
Falls Church, Virginia

Keywords: online social deception, security, defense, detection, phishing attacks, social capital, friending decision, disinformation, game theory, subjective opinion, uncertainty, deep reinforcement learning
Copyright 2023, Zhen Guo
Understanding and Combating Online Social Deception

Zhen Guo

(ABSTRACT)

In today’s world, online communication through social network services (SNSs) has become an essential aspect of people’s daily lives. As social networking sites (SNSs) have become more sophisticated, cyber attackers have found ways to exploit them for harmful activities such as financial fraud, privacy violations, and sexual or labor exploitation. Thus, it is imperative to gain an understanding of these activities and develop effective countermeasures to build SNSs that can be trusted. The existing approaches have focused on discussing detection mechanisms for a particular type of online social deception (OSD) using various artificial intelligence (AI) techniques, including machine/deep learning (ML/DL) or text mining. However, fewer studies exist on the prevention and response (or mitigation) mechanisms for effective defense against OSD attacks. Further, there have been insufficient efforts to investigate the underlying intents and tactics of those OSD attackers through their in-depth understanding. This dissertation is motivated to take defense approaches to combat OSD attacks through the in-depth understanding of the psychological-social behaviors of attackers and potential victims, which can effectively guide us to take more proactive action against OSD attacks which can minimize potential damages to the potential victims as well as be cost-effective by minimizing or saving recovery cost.

In this dissertation, we examine the OSD attacks mainly through two tasks, including understanding their causes and combating them in terms of prevention, detection, and mitigation. In the OSD understanding task, we investigate the intent and tactics of false informers (e.g., fake news spreaders) in propagating fake news or false information. We understand false informers’ intent more accurately based on intent-related phrases from fake news contexts to decide on effective and efficient defenses (or interventions) against them. In the OSD combating task, we develop the defense systems following two sub-tasks: (1) The social capital-based friending recommendation system to guide OSN users to choose trustworthy users to defend against phishing attackers proactively; and (2) The defensive opinion update framework for OSN users to process their opinions by filtering out false information. The schemes proposed for combating OSD attacks contribute to the prevention, detection, and mitigation of OSD attacks.
This Ph.D. dissertation explores the issue of online social deception (OSD) in the context of social networking services (SNSs). With the increasing sophistication of SNSs, cyber attackers have found ways to exploit them for harmful activities, such as financial fraud and privacy violations. While previous studies have focused on detection mechanisms using artificial intelligence (AI) techniques, this dissertation takes a defense approach by investigating the underlying psychological-social behaviors of attackers and potential victims. Through two tasks of understanding OSD causes and combating them through various AI approaches, this dissertation proposes a social capital-based friending recommendation system, a defensive opinion update framework, and a fake news spreaders’ intent analysis framework to guide SNS users in choosing trustworthy users and filtering out phishing attackers or false information. The proposed schemes contribute to the prevention, detection, and mitigation of OSD attacks, potentially minimizing potential damages to potential victims and saving recovery costs.
Dedication

In loving memory of my dear grandmother Chunhua Zhao, 01/26/1936 - 01/28/2021. She was my guiding light and the source of my strength. I am forever grateful for her love and wisdom, which have supported me to overcome every obstacle, especially throughout my toughest times.
Acknowledgments

This dissertation is partly supported by NSF Grant 2107449, 2107450, 2107451, CNS-2141095, and the Army Research Office under Grant Contract Number W91NF-20-2-0140.

I would like to express my sincere gratitude to my advisor, Dr. Cho, for her invaluable support and guidance throughout my academic journey. When I faced the challenge of changing advisors after the third year of my Ph.D. program, Dr. Cho graciously took me under her wing and provided me with the mentorship, directions, and practical plans I needed to succeed. She mentored me to design, write, and present my research and provided the encouragement and necessary support to conduct my independent research. Dr. Cho’s expertise, dedication, and patience have been instrumental in shaping my academic and professional development, and I am deeply grateful for her mentorship.

I would like to extend my heartfelt thanks to the other committee members, Dr. Ing-Ray Chen, Dr. Chang-Tien Lu, Dr. Kurt Luther, and Dr. Srijan Sengupta. I am grateful for their time and effort spent in reading my work, providing constructive criticism, and pushing me to do my best. Each of them brought a unique perspective and expertise to my research, which greatly enhanced the quality and depth. I appreciate their support and encouragement throughout my academic journey, and I am honored to have the opportunity to work with such accomplished scholars. I would like to express my sincere gratitude to my academic advisor, Dr. Cliff Shaffer, for his dedicated efforts and unwavering support throughout my Ph.D. milestones. I also thank Dr. Feng Chen, Dr. Yang Cao, and Dr. You Lu for their invaluable contributions to tackling the challenges in my research projects.

I would like to express my deepest appreciation to the individuals who have had the greatest impact on my life. I thank Sylvia Zhou, her lovely daughter Ally, and her entire family for providing me with a warm and loving family atmosphere when I was far from home. Your emotional support has meant everything to me during each hardship or accomplishment moment. I am also immensely grateful to Dr. Han Cheng for encouraging me to apply for an internship and to rediscover who I was at 15 after years of difficult times. Additionally, I express my appreciation to Ashtaad Dalal, MD, in my medical team, whose dedication and professional care have helped me to maintain my physical and emotional well-being during some of the most challenging times of my life. Finally, I want to pay tribute to my beloved dogs: Bella (Shiba Inu, 10 years), Dori (Shiba Inu, 9 years), and Sora (Siberian Husky, 2 years). You have been my faithful companions over the past ten years through thick and thin, and I am grateful for the love and happiness you have brought into my life.

Lastly, I extend my heartfelt gratitude to those who have contributed to the successful completion of my dissertation by supporting me through each critical decision. I would like to
begin by thanking my parents, Baotai Guo and Minfeng Zhang, for their unwavering emotional and financial support. I also owe a debt of gratitude to Lu Wang, who encouraged me to switch to a computer science program based on my interest, and Dr. Yanjun Li, who generously provided me with a decent financial package for my CS studies. During my MS CS studies, I had unforgettable memories with wonderful friends like Fei Wang, Amy Wang, Zhen Liu, Lisha Zhou, Shaoqing Pan, and Akira Wong, who have continued to support me throughout my Ph.D. program studies. Additionally, I would like to express my appreciation to Dr. Frank Hsu, who provided me with invaluable understanding and support when I had to switch to a new Ph.D. advisor at Virginia Tech. During this challenging time, I was fortunate to receive support from the CS department at Virginia Tech, particularly from Program Coordinators Sharon Kinder-Potter (Blacksburg) and Roxanne Paul (Northern Capital Region). Many of my friends at VT also provided me with endless support, including Dr. You Lu, Dr. Huishan Yang, Dr. Shan Jiang, and Siyu Mi. I am grateful for your unwavering support and companionship.
Contents

List of Figures xii
List of Tables xvi

1 Introduction 1

1.1 Motivation 1

1.2 Research Goal 2

1.3 Research Questions 2

1.4 Key Contributions 3

1.5 Structure of the Dissertation 4

2 Literature Review 6

2.1 Concepts and Characteristics of Deception 6

2.1.1 Multidisciplinary Concept of Deception 6

2.1.2 Types of Deception 9

2.1.3 Taxonomies and Spectrum of Deception 10

2.1.4 Properties of Deception 12

2.2 Types of Online Social Deception Attacks 13

2.2.1 False Information 14

2.2.2 Luring 15

2.2.3 Fake Identity 16

2.2.4 Crowdturfing 16

2.2.5 Human-Targeted Attacks 17

2.2.6 Relationships between Online Social Deception, Social Network Attacks, and Cybercrimes 17

2.2.7 Effect of Online Social Deception Attacks on Security Goals and Safety 19
2.3 Prevention Mechanisms of Online Social Deception

2.3.1 Data-Driven Prevention Mechanisms

2.3.2 Social Honeypots

2.4 Detection Mechanisms of Online Social Deception

2.4.1 User Profile-based Deception Detection Mechanisms

2.4.2 Message Content-based Deception Detection Mechanisms

2.4.3 Network Structure Feature-Based Detection

2.4.4 Hybrid Detection

2.5 Response Mechanisms to Online Social Deception

3 SAFER: Social Capital-based Friend Recommendation to Defend Against Phishing Attacks

3.1 Motivation & Research Goal

3.2 Research Questions

3.3 Key Contributions

3.4 Background & Related Work

3.5 Preliminaries

3.5.1 User Behavior Modeling

3.5.2 Adversary Model

3.6 Proposed Approach: SAFER

3.6.1 Quantification of Social Capital

3.6.2 Phishing Attack Detection

3.6.3 Updating a User’s Status

3.7 Experimental Setup

3.7.1 Datasets

3.7.2 Ethics

3.7.3 Default Parameterization

3.7.4 Experiment Procedures

3.7.5 Comparing Schemes
4.7 Experimental Setup ........................................................................ 80
  4.7.1 Datasets ........................................................................ 80
  4.7.2 Metrics ...................................................................... 80
  4.7.3 Environment Setup ........................................................ 82
4.8 Simulation Results & Analysis ................................................................. 83
  4.8.1 Uncertain Opinions in Disinformation Propagation ............... 84
  4.8.2 Strategy Selection, Payoffs, and Nash Equilibrium .......... 85
  4.8.3 Analysis of Opinion Dynamics Using the SIR Model .......... 88
  4.8.4 Effect of Disinformation Propagation on Polarization ....... 89
  4.8.5 Effect of Disinformation Propagation on Social Capital ..... 91
4.9 Key Findings ........................................................................ 92

5 mudRIA: Multidimensional Uncertainty-Aware Deep Reinforcement Learning-based Intent Analysis of Fake News Spreaders 95
  5.1 Motivation & Research Goal .................................................. 95
  5.2 Research Questions ............................................................. 96
  5.3 Key Contributions ................................................................ 97
  5.4 Background & Related Work .................................................. 98
    5.4.1 Intent Mining .................................................................. 98
    5.4.2 Sentiment Analysis of Fake News .................................. 100
    5.4.3 Deep Reinforcement Learning-based Text Mining ............ 101
  5.5 Proposed Approach: mudRIA .................................................. 103
    5.5.1 Intent-related Text Features for Fake News Spreaders .... 103
    5.5.2 REINFORCE-based Intent Classifier .............................. 106
    5.5.3 Gradient Variance Reduction by Advantage Actor-Critic (A2C) 110
    5.5.4 PG with Multi-dimensional Uncertainty Reward Function .... 112
  5.6 Experimental Setup ................................................................ 115
    5.6.1 Dataset ..................................................................... 115
    5.6.2 Metrics .................................................................... 115
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.6.3</td>
<td>Comparing Schemes</td>
<td>116</td>
</tr>
<tr>
<td>5.7</td>
<td>Experimental Results &amp; Analyses</td>
<td>116</td>
</tr>
<tr>
<td>5.7.1</td>
<td>Multi-Class Intent Prediction Accuracy</td>
<td>116</td>
</tr>
<tr>
<td>5.7.2</td>
<td>Reward based on Model Effectiveness and Efficiency</td>
<td>117</td>
</tr>
<tr>
<td>5.7.3</td>
<td>Sensitivity of Varying Uncertainty Threshold, $\eta$</td>
<td>118</td>
</tr>
<tr>
<td>5.8</td>
<td>Key Findings</td>
<td>121</td>
</tr>
<tr>
<td>6</td>
<td>Conclusions</td>
<td>122</td>
</tr>
<tr>
<td>6.1</td>
<td>Summary of the Key Contributions &amp; Findings</td>
<td>122</td>
</tr>
<tr>
<td>6.2</td>
<td>Publications Accepted, Published, Submitted, and In Preparation</td>
<td>123</td>
</tr>
<tr>
<td>6.3</td>
<td>Future Research</td>
<td>125</td>
</tr>
<tr>
<td></td>
<td>Bibliography</td>
<td>127</td>
</tr>
<tr>
<td></td>
<td>Appendices</td>
<td>153</td>
</tr>
<tr>
<td>A.1</td>
<td>Nash Equilibria Analyses</td>
<td>154</td>
</tr>
<tr>
<td>A.2</td>
<td>Analysis of Strategy Selection of Varying the Ratio of H-DM and Other Users</td>
<td>158</td>
</tr>
<tr>
<td>A.3</td>
<td>Analysis of Varying the Ratio of Attackers</td>
<td>159</td>
</tr>
<tr>
<td>A.4</td>
<td>Additional Results Under the Dataset Cresci15</td>
<td>160</td>
</tr>
</tbody>
</table>
List of Figures

2.1 The key multidisciplinary concepts of deception. ......................... 8
2.2 The spectrum of deception based on the extent of detectability of deception (x-axis); and the extent of good/bad intent of deception and no intent (y-axis). 12
2.3 The number of works that studied different types of online social deception attacks based on five classes of online social deception. ......................... 13
2.4 The relationships between OSN attacks, social deception, and cybercrime. 18
2.5 Three types of agent-based epidemic models. The solid line arrows are transitions from one state to another with probabilities. The dotted line arrows are transactions that may not exist at all times. (a) SIR model: $\beta$ is the infection rate, $\gamma$ is the recovery rate, and $\xi$ is the rate of Recovered to Susceptible. (b) SIHR model: $\alpha$ is a stifling rate, $\beta$ is a refusing rate, $\gamma$ is a spreading rate, $\delta$ is a forgetting rate, $\eta$ is a wakened remembering rate, and $\xi$ is the spontaneous remembering rate. (c) SEIZ model: $\beta$ is an infection rate, $\epsilon$ is a self-adoption rate, $\phi$ is a contact rate, and $\xi$ is a skeptic rate. The details of $p$ and $l$ and the whole model were explained in [128]. 28

3.1 A user’s status update in the SER-SEIR model. .......................... 47
3.2 The overview of the proposed SAFER scheme. .......................... 48
3.3 Probability distributions of relational, cognitive, and human capital ($rc_i$, $cc_i$ and $hi$), namely RC, CC, and HC, using the datasets of Cresci15 [46] (see (a)-(c)) and the 1KS-10KN [295, 296] (see (d)-(f)). The solid red lines and dotted blue lines show the best-fit probability density function curve for attackers and legitimate users. The red and blue bars are the histograms for attackers and legitimate users. ............................................. 53
3.4 Comparison of the FRS schemes regarding the ratio of detected attackers when varying the number of attacks ($NA$) by the same attacker and the fraction of attackers ($Pa$). ............................................. 54
3.5 $S$, $I$, and $R$ comparisons of seven FRSs varying one to five phishing attacks by 20% attackers from the Cresci15 [46] and the 1KS-10KN [295, 296], respectively. 58
3.6 The comparisons of seven FRSs varying the fraction of attackers ($Pa$) from Cresci15 [46] and 1KS-10KN [295, 296], respectively. ......................... 59
4.1 The overview of the proposed opinion update game in an OSN. The top figure illustrates the three subgame examples in one interaction, compared to their related NEs. The bottom-left figure represents the opinion polarization in the last interaction. The bottom-right figure depicts the network status of the bottom-left figure by the epidemic model. ......................................................... 63

4.2 The pairwise interactions of three agent roles, attackers, users, and a defender, in our networked game. ................................................................. 78

4.3 The evolution of SL-based opinions of all legitimate users over 200 interactions in belief (b, blue), disbelief (d, red), uncertainty (u, green), and base rate (a, grey) under the dataset 1KS-10KN [295, 296]. ......................................................... 84

4.4 The histograms of three opinion masses (i.e., (b, d, u)), base rate a, and the projected belief P(b) for all normal users after 200 interactions: Each sub-figure shows the histograms for the five OMs in a given element under the dataset 1KS-10KN [295, 296]. ......................................................... 85

4.5 The probability distributions of the taken strategies by each player type based on NE solutions and the solutions by our proposed game under the dataset 1KS-10KN [295, 296]. ......................................................... 86

4.6 The comparison of the average payoffs of the strategies taken by each player type based on the NE and our proposed games in 1KS-10KN [295, 296]. ......................................................... 87

4.7 The SIR curves of five opinion models in our proposed game, including R_m as the Removed nodes by the defender. ......................................................... 88

4.8 The simulated SIR curves of five opinion models from the ODE parameter optimization. The β and γ values fluctuate at each interaction time. The common ranges from 200 interactions of each β_t and γ_t are shown inside. ......................................................... 89

4.9 The community plots of network topology based on three community detection algorithms, bipartitions, modularity, and label propagation, to show networks before and after disinformation propagation under the five OMs under the dataset 1KS-10KN [295, 296]. ......................................................... 90

4.10 The polarization scores from the four polarization metrics measuring the graph partition of three community algorithms under the dataset 1KS-10KN [295, 296]: Each metric compares the OSN networks before and after disinformation is propagated in 200 user interactions under the five opinion models. ......................................................... 92

4.11 The betweenness scores before and after 200 user interactions and opinions updates with respect to the five opinion models (OMs) under the dataset 1KS-10KN [295, 296]. In (f), we also indicated the mean betweenness of an individual user in the label on the top of each bar. ......................................................... 93
4.12 The trust scores before and after 200 user interactions under the five opinion models (OMs) under the dataset 1KS-10KN [295, 296]. In (f), we also indicated the mean trust of an individual user in the label on each bar.

5.1 Intent classification of spreading fake news.

5.2 The intent mining process from a fake news dataset.

5.3 LSTM pre-training model. Rounded rectangular boxes represent the NN layers. An orange circle highlights the golden intent class, for a news piece example \( x \).

5.4 The overview of REINFORCE-based intent classifier with pre-trained LSTM.

5.5 The uncertainty-aware reward function mUR for DRL models.

5.6 The delayed reward and number of words used from DRL training and testing.

5.7 The effects of varying threshold \( \eta \) in PG and PG+mUR on the prediction accuracy and the length of optimized words (PG is \( \eta = 0 \)).

5.8 The effects of varying threshold \( \eta \) in PGB and PGB+mUR.

5.9 The effects of varying threshold \( \eta \) in A2C and A2C+mUR.

A.1 A normal-form game tree of an attacker with an opinion \((b = 0.11, d = 0.72, u = 0.17, a = 0.13)\) spreading disinformation to a user with opinion \((b = 0.09, d = 0.25, u = 0.66, a = 0.27)\). The attacker and user make simultaneous moves denoted by the dotted lines. The attacker’s strategies are \( a^A_x \)'s, and the user’s ones are \( a^U_y \)'s, respectively. The two utility values under each tree node are the attacker’s utility in Eq. (4.15) and the user’s utility in Eq. (4.21) without using \( p^A_k \).

A.2 A normal-form game tree of a user with an opinion \((b = 0.48, d = 0.24, u = 0.28, a = 0.67)\) interacting with a column player with an opinion \((b = 0.09, d = 0.25, u = 0.66, a = 0.27)\). Nature chooses the type of this column player as an attacker by \( p^A_U \) or a user otherwise. The normal-form trees on the left and right sides are the game tree with a specific type of column player. The two utility values under each tree node are the utilities of the two players, either a user vs. an attacker (i.e., Eq. (4.21) and Eq. (4.15), respectively) on the left tree or a user vs. a user (\( uc^j_{im'} \) in Eq. (4.22)) on the right tree.
A.3 A normal-form game tree of a defender recognizing a player with an opinion
\( (b = 0.09, d = 0.25, u = 0.66, a = 0.27) \) as malicious. Nature chooses the type
of this player as an attacker by a rate \( p_i^A \) or a user otherwise. The normal-
form trees on the left and right sides are the game tree with a specific type
of column player. The two utility values under each tree node are the utility
of the two players, the first is a defender in Eq. (4.18), and the second is an
attacker in Eq. (4.15) on the left tree or user in Eq. (4.21) on the right tree.

A.4 The probability distributions of the best strategies taken by three player roles
during 200 interactions.

A.5 The ratio of \( S, I, \) and \( R \) of five initial percentages of attackers, \( P_{false} \), after
200 interaction times in our proposed game.

A.6 The evolution of SL-based opinions of all legitimate users over 200 interactions
in belief \( (b, \text{blue}) \), disbelief \( (d, \text{red}) \), and uncertainty \( (u, \text{green}) \) under the
dataset Cresci [46].

A.7 The histograms of three opinion masses (i.e., \( (b, d, u) \)), the base rate \( a \), and
the projected belief \( P(b) \) for all legitimate users after 200 interactions: Each
subfigure shows the histograms for the five OMs in a given element under the
dataset Cresci [46].

A.8 The probability distributions of the taken strategies by each player type based
on NE solutions and the solutions by our proposed game under Cresci [46].

A.9 The comparison of the average payoffs of the strategies taken by each player
type based on the NE solutions and our proposed game under Cresci [46].

A.10 The community plots of network topology based on three community de-
tection algorithms, bipartitions, modularity, and label propagation, to show
networks before and after disinformation propagation under the five OMs
under the dataset Cresci [46].

A.11 The polarization scores from the four polarization metrics measuring the
graph partition of three community algorithms under the dataset Cresci [46]:
Each metric compares the OSN networks before and after disinformation is
propagated over 200 user interactions under the five opinion models.

A.12 The betweenness scores before and after 200 user interactions and opinions
updates with respect to the five opinion models (OMs) under the dataset
Cresci [46]. In (f), we also indicated the mean betweenness of an individual
user in the label on the top of each bar.

A.13 The trust scores before and after 200 user interactions under the five opinion
models (OMs) under the dataset Cresci [46]. In (f), we also indicated the
mean trust of an individual user in the label on each bar.
List of Tables

2.1 Goal, intent, and security breach according to a different type of social deception. ................................................. 10
2.2 Classification of Online Social Deception Attacks .................................................. 14
2.3 Impact of Online Social Deception Attacks on Loss of Security Goals and Safety 19
2.4 Defense Mechanisms Surveyed To Deal With Online Social Deception Attacks 36

3.1 Key three dimensions of social capital and their key characteristics .................. 40
3.2 Features used to quantify dimensions of social capital, trust, and friend network in the Cresci15 [46] and 1KS-10KN [295, 296] datasets. ..................... 43
3.3 Design Parameters and Their Default Values ................................................... 50

4.1 Notations of Design Parameters and Their Meanings ...................................... 67
4.2 Input and Output of the Game Model with Three Players ................................. 73
4.3 Key parameters, the explanations, and default values ..................................... 82

5.1 Features of Intent Classes of News Propagators ............................................. 106
5.2 Model Hyper-Parameters Setting ................................................................. 115
5.3 Multi-Class Intent Testing Accuracy ............................................................ 117

A.1 The Expected Payoffs of an Attacker and a U-type User. .............................. 154
A.2 The Expected Payoffs of (U-Type user) vs. (Attacker, U-Type User). ........ 156
A.3 The Expected Payoffs of (Defender) vs. (Attacker, User) ............................ 157
Chapter 1

Introduction

1.1 Motivation

Social media and social network services (SNSs) have become indispensable to people’s everyday lives. In 2020, approximately 82% of Americans reported using social media [138]. This significant surge is due to various benefits users enjoy, such as accessible communication with others, engagement in civic and political activities, searching for jobs, marketing, and sharing information or emotional support. Despite these significant benefits, many people have ambivalent feelings about social media due to privacy concerns or deceptive activities aiming to harm normal, legitimate users [216]. Perpetrators have exploited highly advanced social media technologies as convenient tools for deceiving users [9].

The widespread damage due to online social deception (OSD) attacks has recently increased significantly, with about 25% of people experiencing several types of social deception, such as identity theft, cyberbullying, fraud, or phishing, in 2018 [221]. The severe consequences have led to such OSD attacks being defined as cybercrimes [198] since the early 2000s. The advanced features of SNS technologies have further facilitated the significant increase of serious, sophisticated cybercrimes beyond simple phishing or spamming, such as human trafficking, online consumer fraud, identity cloning, hacking, child pornography, or online stalking [262]. Therefore, we need to deeply understand OSD and consider how to gain an understanding of these activities and develop effective countermeasures to build SNSs that can be trusted.

As shown in Chapter 2 for the in-depth literature review of OSD attacks and their countermeasures, the existing approaches have focused on detection mechanisms for a particular type of OSD by various artificial intelligence (AI) techniques, including machine/deep learning (ML/DL), or text mining. However, fewer studies exist on the prevention and response (or mitigation) mechanisms for effective defense against OSD attacks. Further, there have been insufficient efforts to investigate the underlying intents and tactics of those OSD attackers through their in-depth understanding. In this research, we are motivated to take defense approaches to combat OSD attacks through in-depth understandings of the psychological-social behaviors of attackers and potential victims. Those understandings can effectively guide us to take more proactive actions against OSD attacks which can minimize potential damages to the potential victims and be cost-effective by minimizing or saving recovery costs.
In this dissertation, we examine the OSD attacks mainly through two tasks, including understanding the causes of the OSD attacks and combating them in terms of prevention, detection, and mitigation.

1.2 Research Goal

This dissertation research aims to understand the underlying intents and tactics of the OSD attacks and combat the OSD attacks based on the understanding. To be specific, we execute two main tasks as follows:

[OSD Understanding Task] In this task, we investigate the intent and tactics of false informers (e.g., fake news spreaders) in propagating fake news or false information. We understand false informers’ intent more accurately based on intent-related phrases from fake news contexts to decide on effective and efficient defenses (or interventions) against them.

[OSD Combating Task] In this task, we develop the defense systems for the following two sub-tasks: (1) The social capital-based friending recommendation system to guide online social network (OSN) users to choose trustworthy friends to defend against phishing attackers proactively; and (2) The defensive opinion update framework for OSN users to process their opinions by filtering out false information. The schemes proposed for combating OSD attacks contribute to the prevention, detection, and mitigation of OSD attacks.

1.3 Research Questions

This dissertation research aims to answer the following overarching research questions:

- How much does understanding the behaviors of OSN attackers and potential victims using social-behavioral theories contribute to mitigating the effect of OSD attacks?
  - How much can the social capital of an individual OSN user contribute to defending against OSD attacks (e.g., phishing attacks)?
  - How much can the information processing behavior of an individual OSN user filter out false information and minimize the propagation of false information?
  - How much can the intent analysis of OSD attackers (e.g., fake news spreaders) help minimize false information’s propagation cost-effectively?

- What are the key factors that can mainly influence the responses of OSN users (i.e., potential victims) to OSD attacks (e.g., fake news)?
  - What are the key vulnerability or resilience factors of the OSN users to the OSD attacks?
1.4. Key Contributions

- What environmental aspects can influence the OSN users’ responses to OSD attacks?
- What are the impacts of how to deal with OSD attacks on other aspects of society?
- What are social issues when ODS attacks are not adequately handled regarding the polarization of society, personal and social loss, or unbalanced distribution of social resources?

1.4 Key Contributions

This dissertation has achieved the following key contributions:

- To the best of our knowledge, our work is the first to leverage the concept of multidimensional social capital to model its defense capability against phishing attacks and evaluate its effectiveness. We quantify multiple dimensions of a user’s social capital (i.e., structural, cognitive, and relational capital) based on the user’s behavioral characteristics in OSN contexts derived from two real Twitter datasets where attackers are bots [46] or humans [295, 296].

- We develop a social capital-based friend recommendation system called SAFER and compare its performance with that of existing counterparts, such as social attribute-based [88], topic-based [283, 319], and trust-based [40]. In addition, we conduct extensive system simulation experiments to identify the key dimension of social capital contributing significantly to combating phishing attacks.

- We design a generic game-theoretic opinion framework in an OSN environment against the spread of disinformation. We demonstrate the flexibility of this framework to accommodate various user interactions and opinion models (OMs), including uncertainty, homophily, encounter, herding, and assertion-based updates. In addition, this framework provides the functionality of analyzing the effects of disinformation processing on users’ rational decisions and opinion dynamics and polarization.

- We construct a set of uncertainty-aware payoff equations based on the SL opinion and integrate them into a game theory domain with three agents, i.e., attackers, a defender, and users. Each player can make choices based on uncertainty, observations, and inherent preference to maximize utility when interacting with or accepting other users’ opinions. Furthermore, we solve each player’s preferred strategies by Nash Equilibria (NE), making decisions based on correct beliefs towards the opponents’ moves. Since perfect NE strategies may not be realistic in complex real-world scenarios, we compare the performance between each player’s best strategies chosen under uncertainty and its NE choices to investigate the gap.
• We investigate how different ways of updating opinions can introduce different opinion dynamics and change the social capital of OSN users. We measure an individual user’s social capital based on the levels of bridging and bonding with other users. We measure a user’s bridging by betweenness and bonding by trust, where both metrics have been used to represent a user’s influence or power in a network [274].

• We apply several DRL policy gradient models to predict fake news spreaders’ intent from texts of fake news in the mudRIA framework. We formulate a delayed reward to maximize the intent classification accuracy while optimizing the number of intent-related words. No prior work has combined these two objectives in DRL for intent mining.

• We enhance the limitations of a delayed reward in text classification DRL in that the delayed reward is the same as the final step’s immediate reward. We allow a step to trust a delayed reward with a higher weight if a local critic network-derived Subjective Logic (SL) opinion achieves a high certainty level. Accordingly, we calculate the policy gradients by a new multi-dimensional uncertainty-aware reward function (mUR) and demonstrate an increased accuracy for intent prediction.

• We reduce gradient variances in text classification DRL by Advantage Actor-Critic (A2C) with an episode-level advantage from a pre-trained critic network. Since an episode-level advantage is identical for each step, we have combined the strengths of gradient reduction and a local certainty level to accomplish our intent prediction goals.

1.5 Structure of the Dissertation

This Ph.D. dissertation has the following structure:

• Chapter 2 provides a literature review regarding the concepts, characteristics, and types of online social deception (OSD) and the existing prevention, detection, and response mechanisms to combat OSD. This chapter has been published in a survey paper [89].

• Chapter 3 provides the SAFER framework to quantitatively model three dimensions of social capital and investigate the effects of resistance to phishing attackers by social capital on a friend recommendation system. This SAFER framework has been published in a top-tier conference paper [91].

• Chapter 4 provides a game-theoretic opinion interactions framework to simulate the disinformation propagation dynamics in an OSN and shows the opinion polarization and network communities by five opinion models based on our published papers [90, 92, 93].
1.5. Structure of the Dissertation

- Chapter 5 develops a framework mudRIA for intent classification of the online fake news spreaders by multi-dimensional uncertainty-aware deep reinforcement learning (DRL). This work investigated a series of policy gradient-based DRL models based on our accepted paper [96] and papers in review [97, 98].

- Chapter 6 summarizes the key contributions, findings, and publication list achieved in my Ph.D. timeline.

- Appendix A summarizes the additional experiments and their results for the research project addressed in Chapter 4.
Chapter 2

Literature Review

In this chapter, we provide the related literature review regarding the concepts, characteris-
tics, and types of online social deception and the existing prevention, detection, and response
mechanisms to combat OSD. This chapter has been published in a survey paper [89].

2.1 Concepts and Characteristics of Deception

The concept of deception is highly multidisciplinary and has been studied in various domains.
In this section, we discuss the root definitions of deception and the fundamental properties
of deception which have been applied in launching OSD attacks on OSN platforms.

2.1.1 Multidisciplinary Concept of Deception

Let us start by looking at the dictionary definition of deception [56]. Deception is defined as:
“To cause to believe what is false.” However, the definition is too broad, and many deception
researchers raised doubts about the definition. In the literature, the concepts of deception
have been discussed from different perspectives under different disciplines. The following
sections briefly discuss how different discipline has studied deception.

Philosophy

In Philosophy, intentional and unintentional (by mistake) deception has been discussed, such
as ‘inadvertent or mistaken deceiving’ [30]. However, the common concept of deception was
mostly agreed with ‘misleading a belief’ either inadvertently or mistakenly [78, 225]. The
core aspects of deception in Philosophy lie in an intentional act to mislead an entity into
believing a false belief.
Behavioral Science

Behavior scientists\textsuperscript{1} investigated the concept of deception and its process in the behaviors of animals or humans. Two main concepts of deception are: (1) Functional deception for an individual’s behavior (i.e., a signal) to mislead the actions of others; and (2) Intentional deception referring to intentional states, such as beliefs and/or desires, guide an individual’s behavior, leading to the misrepresentation of belief states [107, 154, 243].

Psychology

Psychologists defined deception as a behavior providing information to mislead subjects in some direction [2] or explicit misrepresentation of a fact aiming to mislead subjects [119, 193]. The major psychological deception study focused on identifying cues as committing a crime [83], psychological symptoms for self-deception [32, 110], individual differences and/or cues to deception [222], and verbal or non-verbal communication cues [324].

Sociology

Sociological deception research has mainly studied the effect of deception in various social contexts on both positive and negative aspects [180] or deception as a relational or marketing strategy [208].

Public Relations

In this domain, self-deception has been studied as a strategic solution to resolve internal or external crises [231]. The external role of self-deception is to avoid a disastrous impact on an organization [201] by attributing a problem (or guilty) to an individual or victim.

Communications or Linguistics

In this domain, deception research often aims to identify verbal or non-verbal indicators for deceptive communications. Interpersonal deception theory (IDT) views deception as an interactive process between senders and receivers, exchanging non-verbal and verbal behaviors and interpreting their communicative meanings. IDT further explains that deceivers strategically manage verbal communications to deceive receivers successfully [23, 24]. Experimental studies showed that deceivers produced more words, fewer self-oriented (e.g., I, me, my), and more sense-based words (e.g., seeing, touching) than truth-tellers [105].

\textsuperscript{1}We consider biologists, ecologists, neuroscientists, and medical scientists as ‘behavioral scientists’.
CHAPTER 2. LITERATURE REVIEW

Command and Control

In the military domain, deception refers to any planned maneuvers to reveal false information and hide the truth from an enemy to mislead the enemy and entice the enemy to undertake the wrong operations [49, 181, 287]. Military deception involves many individuals or organizations as both deceivers and victims and takes a long time period [49].

Computing and Engineering

Cyber attackers frequently employ deceptive techniques in various forms, including phishing, social engineering, fraudulent advertisements, and stealthy attacks, among others [108, 218]. Furthermore, with the rise in the number of phishing attacks, researchers have studied the susceptibility of online users to such attacks in terms of demographics [169, 200, 235] and personality traits [50, 66, 102, 103, 185, 206, 207]. We will discuss many detection mechanisms for OSD attacks that have been developed in the literature in Section 2.4.

For easy grasping of the key multidisciplinary concepts of deception, we summarized the key deception concepts under different disciplines in Figure 2.1.
2.1.2 Types of Deception

Although deception can be intentional or unintentional, we focus on intentional deception in this work, which is more related to an attacker’s intent. Intentional deception consists of deception with malicious intent and non-malicious intent for a deceiver’s interest [65].

The goals of malicious deception include:

- **Financial benefit**: Many deceptive behaviors aim to obtain a monetary benefit. Financial benefit is a common reason for an individual’s online deceptive behavior. For example, a spammer can be paid by clicking advertisements by attracting online traffic to specific sites [192]. Malicious users spread phishing links to collect credentials from victims [266].

- **Manipulation of public opinions**: In social media, social and political bots play a role in influencing public opinions [73]. Malicious bots spread spam and phishing links. Politicians and governments worldwide have been using such bots to manipulate public opinions.

- **Cooperative deception**: Cooperation is a strategy of balancing costs and benefits and maintaining stakeholder relationships in the deception or cooperation interactions with opponents [254], often used in public relations.

- **Parasitism** [231]: This refers to ‘the false framing of responsibility’, which can be easily used to solve complicated issues without introducing long-term investigations that may cause structural changes.

The goals of non-malicious deception are commonly discussed as follows:

- **Privacy protection**: Deception can be used as a defense for privacy protection at the organization or individual level. It is also called defensive deception. There are a few methods for individual-level privacy protection in cyberspace. Some privacy techniques add noise to a user’s data to protect against attackers [209] because the data can be modified before publishing.

- **Self-presentation**: People use a fake presentation to present themselves in certain roles or intents [229]. Self-presentation is an activity to impress others for both liars and truth-tellers. Self-presentation is one way of understanding nonverbal communication [54]. Self-presentation can be one of the prediction cues for deception [54].

- **Self-deception**: This is to hide true information reflecting the conscious mind unconsciously [254], with the two main benefits of not being detected easily and reducing immediate cognitive costs.

In TABLE 2.1, we summarized what social deception is malicious or not and how it is associated with a breach of security goal.
Table 2.1: Goal, intent, and security breach according to a different type of social deception.

<table>
<thead>
<tr>
<th>The goal of social deception</th>
<th>Malicious vs. non-malicious intent</th>
<th>Breach of security goals</th>
</tr>
</thead>
<tbody>
<tr>
<td>Financial benefits</td>
<td>Malicious</td>
<td>Loss of confidentiality and integrity</td>
</tr>
<tr>
<td>Manipulation of public opinions</td>
<td>Malicious</td>
<td>Loss of integrity</td>
</tr>
<tr>
<td>Cooperative deception</td>
<td>Malicious</td>
<td>Loss of integrity</td>
</tr>
<tr>
<td>Parasitism</td>
<td>Malicious</td>
<td>Loss of integrity</td>
</tr>
<tr>
<td>Privacy protection</td>
<td>Non-malicious</td>
<td>Loss of confidentiality</td>
</tr>
<tr>
<td>Self-presentation</td>
<td>Non-malicious</td>
<td>Loss of integrity</td>
</tr>
<tr>
<td>Self-deception</td>
<td>Non-malicious</td>
<td>Loss of integrity</td>
</tr>
</tbody>
</table>

2.1.3 Taxonomies and Spectrum of Deception

This section discusses the related concepts and spectrum of deception. Deception can be defined and explained by a set of related terminologies in which those concepts should be defined and compared. Deception exists in our daily life in both verbal and nonverbal forms. Deception ranges a broad spectrum with varying intent and detectability (i.e., the extent of deception being detected).

Key Taxonomies of Deception

In this section, we discuss a set of related terminologies related to deception. Most common concepts are defined in the dictionary and discussed in the cybersecurity literature [32, 54, 56, 223, 231].

- **Deceivee** [223]: The victim of a deception.
- **Deceiver** [223]: The perpetrator of a deception.
- **Susceptibility** [56]: Likelihood to be deceived.
- **Exploitation** [56]: The use of resources and benefit from them (e.g., damage to systems) by attackers.
- **Self-deception** [32]: A conscious false belief held with a conflicting unconscious true belief.
- **Trust** [56]: Reliance on confidentiality and integrity from other sources and with confidence. Earning high trust from a deceivee can be easily exploited by a deceiver.
- **Lying** [54, 223]: Deliberate verbal deceptions. People often lie in pursuit of material gain, personal convenience, or escape from punishment.
- **White lying** [231]: Normal standards for the lighthearted type of deception.
• **Belief** [56]: A truth in somebody’s mind, truth basis.

• **Misbelief** [56]: A misplaced belief (i.e., mistakenly believing in false information)

• **Perception** [56]: The state of being aware of something through the senses.

**Spectrum of Deception**

In daily life and social networks, deception spans a spectrum of verbal and non-verbal behaviors. This section lists a few deceptions based on [64, 223, 238].

• **White lies** [223]: Harmless lies to avoid hurting others’ feelings and smooth relationships.

• **Humorous lies** [238]: Jokes that are obvious lies, such as practical jokes.

• **Altruistic lies** [223]: Good lies for protecting others, such as for preventing children from worrying.

• **Defensive lies** [223]: Lies to protect the deceiver, such as lies to eliminate repeated telemarketers.

• **Aggressive lies** [223]: Lies to deceive others for the benefit of the deceivers.

• **Pathological lies** [223]: Lies by a deceiver with a psychological disorder.

• **Nonverbal minimization** [64]: Understating an essential case in nonverbal camouflage.

• **Nonverbal exaggeration** [64]: Overstating a vital case to hide others.

• **Nonverbal neutralization** [64]: Intentionally hiding normal emotions when inquired about emotional things.

• **Nonverbal substitution** [223]: Intentionally changing a sensitive concept with a less sensitive one.

• **Self-deception** [223]: The goal is to push reality into the subconsciousness.

Figure 2.2 represents the spectrum of deception from the lowest detectability to the highest detectability, from the lowest bad intent (good intent) to no intent, and to the highest bad intent. In general, the deception with lower detectability is more with good intent, such as altruistic lies and white lies. Nonverbal deception is usually with bad intent and can be detected by professionals. Those behaviors can also be used as cues to detect lies. The deceptions with neutral intent can also be easily detected. These concepts can be applied to detect malicious behaviors in online social networks, as many offline human behaviors are also easily observed in online user behaviors.
2.1.4 Properties of Deception

Via the in-depth literature review, we observe the following unique key properties of deception:

- **Misleading one’s belief**: Regardless of intent, deception can mislead one’s belief which is actually false. Since deception, as an action, induces confusion or false information, false beliefs may be formed regardless of intent or outcome.

- **Impact by deception**: Confusion or misbelief introduced by deception brings a negative or positive outcome based on its original intent or proper execution. However, when deception with a certain intent is not adequately executed as planned or is used mistakenly, the outcome as its impact may not be predictable, resulting in high uncertainty (e.g., uncertain outcome). Hence, if deception is intended, it should be planned with multiple scenarios to lower the risk introduced by deception in terms of a deceiver’s perspective.

- **Success only by a deceivee’s cooperation**: For deception to be successful, a deceivee should be deceived by the deception. Even if deception is performed, but the deceiver detects the deception, no effect can be introduced.

- **Action as a strategy**: Deception can be used as a strategy to deal with situations with conflicts. Intentional deception aims to mislead a target entity’s belief and make the target choose a suboptimal (or poor) action that can benefit the deceiver.

- **Signals as deception cues**: When deception is used, even if it can be very subtle, some signals exist. Well-known deception strategies are to increase uncertainty (e.g., no signal increases uncertainty) or mislead one’s belief (e.g., a false signal leads to false beliefs). Although both deception techniques aim to make a deceiver choose a wrong
2.2 Types of Online Social Deception Attacks

Various types of OSD attacks have been discussed in the literature. In this section, we classify various types of OSD attacks into five classes based on the key intent of each attack class. In addition, since similar studies have used ‘online social network attacks’ and ‘cybercrime’ to discuss OSD, we discussed our view on how they are distinguished from and related to each other. All the OSD types are summarized in TABLE 2.2, and the corresponding work count for each OSD type is illustrated in Figure 2.3. Lastly, we discussed how OSD attacks breach security goals in the CIA triad and safety, intending to give an alert on how severe the OSD can be as a societal problem.
Table 2.2: Classification of Online Social Deception Attacks

<table>
<thead>
<tr>
<th>OSD Class</th>
<th>Type</th>
<th>Description</th>
<th>Intent &amp; Potential Damage</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>False Information</td>
<td>Fake news</td>
<td>News contradicts, fabricates, or conflates the ground truth and spreads in OSN.</td>
<td>Credibility loss, economical and political misleading, controlling public opinions</td>
<td>[127, 233, 273]</td>
</tr>
<tr>
<td></td>
<td>Rumors</td>
<td>An unverified assertion that starts from one or more sources and spreads over time from node to node in a network.</td>
<td>Misleading people’s decisions, panic in public, government credibility loss</td>
<td>[272]</td>
</tr>
<tr>
<td></td>
<td>Information</td>
<td>False information is deliberately and often covertly spread in order to influence public opinion or obscure the truth.</td>
<td>Advertising, campaigns</td>
<td>[53]</td>
</tr>
<tr>
<td></td>
<td>manipulation</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Fake reviews</td>
<td>Malicious users write fake reviews, opinions, or comments on social media to mislead others.</td>
<td>Influencing user’s options or decisions, advertising, reputation loss</td>
<td>[301]</td>
</tr>
<tr>
<td>Luring</td>
<td>Phishing</td>
<td>Attackers trick users into revealing sensitive information related to work, financial credentials, or even personal data to be used in fraudulent activities</td>
<td>Confidential personal data leakage, the launch of advertising campaigns, pornography</td>
<td>[1, 60, 266]</td>
</tr>
<tr>
<td></td>
<td>Spamming</td>
<td>Attackers send unsolicited messages (spam) in bulk to OSN users</td>
<td>Reputation loss, malicious advertising</td>
<td>[218]</td>
</tr>
<tr>
<td>Fake Identity</td>
<td>Fake Profile</td>
<td>Attackers create a considerable amount of fake identities for their own benefit.</td>
<td>Personal information leakage, stealing money</td>
<td>[100]</td>
</tr>
<tr>
<td></td>
<td>Compromised account</td>
<td>Attackers hack legitimate user accounts that are created and used by their fair owners and then for ill purposes.</td>
<td>Reputation loss, account loss, personal privacy leakage</td>
<td>[63, 137]</td>
</tr>
<tr>
<td></td>
<td>Profile cloning</td>
<td>Attacker clones a pre-existing user profile either in the same or a different OSN.</td>
<td>Reputation loss, sensitive information leakage, account loss</td>
<td>[218]</td>
</tr>
<tr>
<td>Crowdturfing</td>
<td>Crowdturfing</td>
<td>Attackers are gathered by crowdsourcing systems and speak fake and inaccurate information to mislead people</td>
<td>Spreading malicious URLs, forming astroturf campaigns, manipulating opinions</td>
<td>[159, 161, 275, 276, 291]</td>
</tr>
<tr>
<td>Human Targeted Attacks</td>
<td>Human trafficking</td>
<td>Traffickers use computers and networks to transport a great number of victims and advertise services across geographic boundaries for the labor trade or sex trade</td>
<td>Sexual exploitation, modern slavery, forced labor or services, removal of organs</td>
<td>[69, 86, 156]</td>
</tr>
<tr>
<td></td>
<td>Cyberbullying</td>
<td>Deliberate and repetitive online harassing or harming of someone.</td>
<td>Reputation loss, cyber harassment, teen depression</td>
<td>[218]</td>
</tr>
<tr>
<td></td>
<td>Cyber-grooming</td>
<td>An adult tries to establish an online, emotional connection with a child in order to sexually abuse the child.</td>
<td>Reputation loss, cyber harassment</td>
<td>[218]</td>
</tr>
<tr>
<td></td>
<td>Cyberstalking</td>
<td>Attackers exploit personal information, such as phone number, home address, and schedule, in SNS user’s profile</td>
<td>Reputation loss, personal data leakage, cyber harassment, safety loss</td>
<td>[218]</td>
</tr>
</tbody>
</table>

2.2.1 False Information

False information on the web and social media can be classified as misinformation and disinformation. Misinformation can be considered ‘deception without intent’, which mistakenly misleads people’s beliefs due to the false information propagated. Disinformation can be categorized as ‘deception with intent,’ aiming to mislead people’s beliefs. False information can also be categorized as opinion-based vs. fact-based. Opinion-based false information is propagated without ground truth. On the other hand, fact-based false information can mislead people’s beliefs due to fraud from ground truth, such as hoaxes and fake news in social media [127].

Although no formally accepted terminologies exist to distinguish different kinds of false
information, we follow Jiang and Wilson [127]’s two criteria, which are veracity and intentionality [236], to discuss false information as below:

- **Fake News**: Fake news caused by serious fabrications or large-scale hoaxes [224] has wildly spread via OSNs since the beginning of the 2016 US presidential election cycle. Flintham et al. [71] reported that two third of survey respondents accessed news via Facebook. Facebook and Twitter have banned thousands of pages, identifying them as the major culprit in generating and promoting misinformation [127]. Fact-checking news articles from different sources has become a common way to determine the veracity of social media posts. Vosoughi et al. [273] found that fake news spreads faster than truthful news. The time lag between fake news and fact-checking by fact-checking websites is 10-20 hours [233].

- **Rumors**: Vosoughi et al. [272] defined a rumor as an unverified assertion that starts from one or more sources and spreads over time from one user to another in a network. A rumor can be validated as true or false via real-time verification on Twitter or remain unresolved.

- **Information Manipulation**: One of the causes of information manipulation is opportunistic disinformation [53]. This term means false information is deliberately and often covertly spread (e.g., planting a rumor) to influence public opinions or obscure the truth. Malicious users propagate opportunistic disinformation mainly for financial interests or political purposes.

- **Deceptive Online Comments or Fake Reviews**: Malicious users write fake reviews, opinions, or comments on social media to mislead other users. Usually, fake reviews are classified as opinion-based false information [148]. Social bots often generate fake reviews automatically [301].

### 2.2.2 Luring

Luring has been commonly used as one of the popular deception strategies. The most common luring techniques in online worlds include:

- **Spamming**: Social media platform users can receive unsolicited messages (spam) that range from advertising to phishing messages [218]. Malicious users usually send spam messages in bulk to influence many legitimate users.

- **Phishing**: Online phishing attacks, such as phishing webpages or phishing emails, are one type of cybercrimes that can lure users into revealing sensitive or credential information and stealing private or financial information through social engineering attacks [60] or using other fraudulent, illegal activities [1]. These malicious activities
can cause severe economic losses and threaten the credibility and financial security of OSN users.

2.2.3 Fake Identity

Attacks using fake identities have their basis in social deception and include:

- **Fake Profile**: In OSNs, attackers create a huge amount of fake identities for their benefit, also called a Sybil attack. For example, on Facebook, attackers can leak other users’ personal information, such as e-mail and physical addresses, date of birth, and employment data. Identity theft can take financial interests and access photographs of the friends of the victims [100].

- **Profile Cloning**: Attackers can secretly create a duplicate of an existing user profile on the same or different social media platforms. Since the cloned profile resembles the current profile, attackers can utilize the friend relationship and deceive and send friend requests to the contacts of the cloned user. The attacker can steal sensitive data from the user’s friends by constructing a trust relationship with a potential victim user. Profile cloning has exposed severe societal threats because attackers can commit more serious cybercrimes, such as cyberbullying, cyberstalking, and blackmail, which can introduce physical threats to potential victims [218].

- **Compromised Accounts**: Attackers can hack and compromise legitimate user accounts [63]. Unlike Sybil accounts, compromised accounts are originally maintained by real users with normal social network usage history and have established social connections with other legitimate users.

2.2.4 Crowdturfing

Malicious, paid human workers can perform malicious behaviors to achieve their employer’s goal. This social deception type is called *crowdturfing*. For example, participants in an astroturfing campaign are organized by crowdsourcing systems [275]. Crowdturfing gathers crowdturfing workers and spreads fake information to mislead people’s beliefs and/or public opinions on social media. Crowdturfing activities in social media exploit social networking platforms (e.g., instant message groups, microblogs, blogs, or online forums) as the main information channel of the campaign [291]. Crowdturfing in social media usually spreads malicious URLs, forms astroturf campaigns, and manipulates public opinions. Usually, it is challenging to detect crowdturfing accounts because their social media accounts are mixed with regular posts as camouflage.

Chinese crowdsourcing sites [275] and Western sites [161] have been studied to analyze crowdturfing in campaigns. Three classes of crowdturbers (i.e., professional users, casual users, and
middlemen) are identified in Twitter networks. In addition, their profiles, activities, and linguistic characteristics have been analyzed to detect crowdturfing workers [159]. Machine learning (ML)-based crowdturfing detection mechanisms have been considered [276]. Two common types of adversarial attacks are evasion attacks (i.e., attacks changing behavioral features) and poisoning attacks (i.e., administrators polluting training data) [276].

2.2.5 Human-Targeted Attacks

Advanced online platforms have provided efficient tools for human-targeted criminals to achieve their goals. Cybercriminals often begin their cybercriminal activities by establishing trust relationships with potential victims. Since this typically involves social deception [111], we have included human-targeted attacks as one of the categories of OSD examined in this chapter.

The typical human-targeted OSD attacks include:

- **Human Trafficking**: Offline traditional human trafficking means traffickers kidnap the victims (mostly women and children) for trading with the purpose of labor exploitation or sex trafficking [69]. Cybertrafficking means that traffickers leverage cyber platforms to efficiently traffick a significant number of victims by using advertising services across geographic boundaries [86, 156].

- **Cyberbullying**: An individual repeatedly harasses someone online, particularly targeting adolescents [218]. Such attacks can cause significant harm to the victim through various means, including deception, public humiliation, and unwanted contact [61].

- **Cybergrooming**: In this attack, adult criminals attempt to establish trust relationships with potential victims, mostly female children, using online social media platforms. They intend to have improper sexual relationships with them or produce child pornography products [218, 306].

- **Cyberstalking**: Malicious users can exploit legitimate users’ online information and harass them by stalking [218]. Without proper security protection, an individual can expose their private information (e.g., phone number, home address, work location) on social media platforms without awareness.

2.2.6 Relationships between Online Social Deception, Social Network Attacks, and Cybercrimes

Social network attacks, including traditional threats, social threats, and multimedia content threats, are the general security threats concerned in the literature [218]. Those security and
18

CHAPTER 2. LITERATURE REVIEW

privacy threats include all detrimental activities with malicious intent. Social deception is part of social network attacks, as shown in Figure 2.4 because social deception attacks can only be successful when the victims are being deceived from the attacker’s perspective.

Four types of social network attacks are considered OSD attacks: Unsolicited fake information attacks, identity attacks, crowdtrufing, and human-targeted attacks. The specific types of attacks are described in Section 2.2. Some OSD attacks, such as personal and confidential information leakout or identity theft, have been treated as cybercrimes [198] since the early 2000s. The advanced features of social network service technologies further facilitated the significant increase in serious, sophisticated cybercrimes, such as human trafficking, online consumer fraud, identity cloning, hacking, child pornography, and/or online stalking [262].

Figure 2.4 illustrates the relationships between OSN attacks, OSD attacks, and cybercrimes. Although cybercrime is considered the most serious of cyberattacks, we can observe that many attacks overlap. OSD attacks overlap either OSN attacks or cybercrime or both. Cybercrimes, such as consumer fraud, cryptojacking, enterprise ransomware, supply chain attacks, and malicious email attacks [246], fall into a separate group because these attacks are spread on the Internet, which is much broader than OSN platforms. There are no explicit guidelines if certain OSN attacks or threats are illegal or if threats are illegal, but their impact may not be direct. For example, when a user’s data privacy (or integrity) is breached but no actual loss is found, it is hard to predict if there are future security concerns.

Although cybercriminals caused severe adverse effects on society and individuals, 44% of the victims reported to the police [82]. Victims’ reporting is a beneficial practice to increase the awareness of the communities to defend against potential cybercrimes. Victims may report to not only the police but also the corporation in an active dialogue environment.
2.2. TYPES OF ONLINE SOCIAL DECEPTION ATTACKS

Table 2.3: Impact of Online Social Deception Attacks on Loss of Security Goals and Safety

<table>
<thead>
<tr>
<th>Social Deception Attack</th>
<th>Security Breach</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fake News</td>
<td>Data Integrity</td>
</tr>
<tr>
<td>Rumors</td>
<td>Data Integrity</td>
</tr>
<tr>
<td>Information Manipulation</td>
<td>Data Integrity</td>
</tr>
<tr>
<td>Fake Reviews</td>
<td>Data Integrity</td>
</tr>
<tr>
<td>Spimming</td>
<td>Account Confidentiality</td>
</tr>
<tr>
<td>Phishing</td>
<td>Account Confidentiality</td>
</tr>
<tr>
<td>Fake Profile</td>
<td>Account Integrity</td>
</tr>
<tr>
<td>Profile Cloning Attack</td>
<td>Authentication</td>
</tr>
<tr>
<td>Compromised Account</td>
<td>Account Integrity, Account Availability</td>
</tr>
<tr>
<td>Crowdturfing</td>
<td>Data Integrity, Network Integrity</td>
</tr>
<tr>
<td>Human traffickering</td>
<td>Confidentiality, Safety</td>
</tr>
<tr>
<td>Cyberbullying</td>
<td>Confidentiality, Safety</td>
</tr>
<tr>
<td>Cyber-grooming</td>
<td>Confidentiality, Safety</td>
</tr>
<tr>
<td>Cyberstalking</td>
<td>Confidentiality, Safety</td>
</tr>
</tbody>
</table>

or share the victim stories with families and close friends [82]. Cybercriminal profiling is highly challenging compared to profiles of traditional criminals because cybercriminals can easily leave the platforms. However, it is very beneficial to identify common characteristics of cybercriminals [198, 278] and useful for their early detection. Profiling can follow the procedure in the Behavioral Evidence Analysis [259]. Since most cybercrime victims are corporations and/or their customers, corporations can predict potential insider criminals more intelligently with the help of cybercriminal profiling [198].

2.2.7 Effect of Online Social Deception Attacks on Security Goals and Safety

The CIA triad security goals play a significant role in the information security practice. With the growth of socio-technical security issues, the original CIA triad is expanded with more specialized aspects, such as authentication and non-repudiation [179]. However, they still have limitations in systems and data for the broader organizational and social aspects of security [227]. OSN security has three levels of security goals: network level, account level, and message level. Achieving the CIA security goals can contribute to all social network security levels. In addition to the three security goals, we added another goal: safety. A person and other non-information-based assets must also be protected in the cyber security practice [270]. For example, cyberbullying can cause direct physical harm to a victim even without losing information confidentiality, integrity, or availability [270]. Therefore, we included human safety as a non-information security goal. For readers’ convenience, we summarize how OSD attacks can breach security goals and safety in TABLE 2.3.
2.3 Prevention Mechanisms of Online Social Deception

In this section, as proactive defense mechanisms, we discuss two types of OSD prevention mechanisms: *Data-driven prevention mechanisms* and *social honeypots*.

2.3.1 Data-Driven Prevention Mechanisms

Prevention mechanisms against OSD attacks have been little explored. We discuss several types of data-driven prevention mechanisms that have been commonly used to deal with OSD attacks as follows:

- **Fake News Prevention**: Saad et al. [226] proposed a blockchain-based system to fight against fake news by recording a transaction in blockchain when posting a news article and applying authentication consensus of the record. The result was measured by an authentication indicator along with the post. In this design, when a user saw a post, the authentication indicator associated with the post was shown as the verification status: successful, failed, or pending. This mechanism addressed the following services for preventing fake news spread in the OSN: (i) Determine the authenticity of the news by users’ consensus to ensure the trustworthiness of posts; (ii) identify a malicious user from the transaction record; and (iii) delete false information posts with a penalty applied to the fake news attackers. In general, the malicious attackers are the normal users, but normal users do not have write access to the blockchain. Only the information source from a group of publishers or a group of a social network is allowed to commit transactions to the blockchain.

- **Phishing Prevention**: Florêncio and Herley [72] proposed a low-delay phishing prevention method where a client reports the reuse activities of a user password on unknown websites, and a server makes decisions and updates the blocked list. Gupta and Pieprzyk [99] proposed a defense model to classify web pages on a collaborative platform PhishTank, which integrated a plug-in into a browser to verify blacklisting and blocking lists.

- **Identity Theft Prevention**: Tsikerdekis [257] proposed a proactive approach to prevent identity theft by social network data to establish a behavioral profile of a community through their shared contributions. Malicious accounts can be barred before joining a community based on the deviation of user behaviors from the community’s profile.

- **Cyberbullying Prevention**: Dinakar et al. [57] proposed a dashboard reflective user interface in social network platforms for both cyberbullying attackers and victims. The reflective user interface integrated notifications, action delay, and interactive education. Their user study revealed that the in-context dynamic help in the user interface is effective for the end-users.
2.3. Prevention Mechanisms of Online Social Deception

Pros and Cons: Preventing OSD attacks requires assessing users or information to determine whether to allow the user or information to stay or be propagated in a given OSN. However, the so-called trust assessment is not clear. The key merit of the prevention mechanisms should be how quickly false information or malicious users are detected. Otherwise, it is not distinguishable from detection mechanisms. In addition, the effectiveness of the prevention mechanisms is still measured by detection accuracy. There should be more valuable metrics that can capture the nature of the proactiveness of the prevention mechanisms. In addition, no real-world implementation using the prevention mechanisms is considered, which limits the applicability of the prevention mechanisms.

2.3.2 Social Honeypots

The concept of good bots has recently appeared by creating social network avatars to identify malicious activities through highly intelligent, sophisticated attacks, such as advanced persistent attacks (APTs) [269]. Honeypot technology is not new and has been popularly used in communication networks as a defensive deception to proactively deal with attackers by luring them to honeypots to prevent them from accessing a target [44]. The existing approaches using social honeypots have mainly focused on detecting social spammers, social-bots [322], or malware [158, 160, 203, 204, 205, 244, 285] as a passive monitoring tool. These works use some profiles of attackers to detect them based on the features collected from the social honeypots placed as fake SNS accounts (e.g., Facebook or Twitter).

Although the original purpose of social honeypots was to proactively prevent attackers from accessing system/network resources, they have been used as a complement to detect various OSN attacks. However, the original purpose of social honeypots lies in a proactive intrusion prevention mechanism. In addition, although the social honeypots can be used as a detection tool for OSN or OSD attacks, their goal is early detection or mitigation based on proactive defense. Hence, we include social honeypots as prevention mechanisms for OSD attacks.

For the social honeypots to be used as detection mechanisms, they are defined as information resources that monitor a spammer’s behaviors and log their information (e.g., their profiles and contents in social networking communities) [158]. This early study detected deceptive spam profiles in MySpace and Twitter by social honeypot deployment. Based on the spammer they attracted, an SVM spam classifier was trained to identify spammers and legitimate users. An ML-based classifier was also developed to identify unknown spammers with high precision in two social network communities. Lee et al. [160] detected content polluters on Twitter by designing Twitter-based social honeypots. The 60 social honeypot accounts followed other social honeypot accounts and posted four types of tweets to each other. They investigated the harvested users in nine clusters via the Expectation-Maximization (EM) algorithm. They also used content polluters classification by Random Forest and improved the results by standard boosting and bagging, considering different feature group combinations. Haddadi and Hui [100] focused on privacy and fake profiles by characterizing fake profiles
and reducing identity theft threats. They set up social honeypots using the fake identities of celebrities and ordinary people and analyzed the different behaviors (e.g., the number of friends, friend requests, and public/private messages) between those fake accounts. Stringhini et al. [244] studied 900 honey-profiles to detect spammers in three social network communities (e.g., MySpace, Facebook, and Twitter) where their honey-profiles have geographic networks. They collected activity data for a long time (i.e., one year). In addition, this work identified both spam profiles and spam campaigns based on the shared URL.

Virvilis et al. [269] described the common characteristics of APT attackers and malicious insiders and discussed multiple deception techniques for the early detection of sophisticated attackers. They created social network avatars in the attack preparation phase (information gathering), along with fake DNS records and HTML comments. Zhu et al. [322] showed the analysis and simulation of infiltrating social honeybots defense into botnets of social networks. The framework SODEXO (SOcial network Deception and EXploitation) had three components: HD, HE, and PAS. The HD deployed a moderate number of honeybots in the social network; The HE modeled the dynamics and utility optimization of honeybots and botmaster by a Stackelberg game model. The results showed that a small number of honeybots could significantly decrease the infected population (i.e., a botnet) in a large social network.

Paradise et al. [203, 204] simulated defense strategies for account monitoring attacks in OSNs. The attackers sent friend requests to some community members chosen by different attacker strategies. In addition, the attackers may have full knowledge of the defense strategies. The defender chose a set of accounts to monitor based on various criteria. They analyzed the acceptance rate, hit rate, a number of friends before hit, and monitored cost between combinations of attackers and defenders. The result showed that under the sophisticated attackers with the full knowledge of defense strategies, defense using PageRank and most connected profiles had the best detection with minimum cost. Paradise et al. [205] targeted at detecting the attackers in the reconnaissance stage of APT. The social honeypot artificial profiles were assimilated into an organizational social network (Xing and LinkedIn) and received friend requests from organization employees. The authors analyzed the attacker profiles collected in the social honeypot.

Badri Satya et al. [16] collected the so-called ‘fake likers’ on Facebook, who are paid workers to propagate fake likes using linkage and honeypot pages. The authors extracted the four types of profiles and behavior features and trained classifiers to detect fake likers. The temporal features were cost-efficient compared to the previous research. They also evaluated the robustness of their work by modifying features using individual and coordinated attack models. De Cristofaro et al. [51] studied paying for ‘likes fraud’ on Facebook and linking the campaigns to honeypot pages to collect data. They analyzed the page advertising and promotion activities. Nisrine et al. [195] discovered malicious profiles by social honeypots and used both feature-based strategy and honeypot feature-based strategy to collect data. Combining honeypot features can increase ML accuracy and recall compared to the scheme with traditional features only.
Zhu [321] defined “active honeypots” as active Twitter accounts that capture more than ten new spammers daily, similar to the spammer network hubs. They extracted 1,814 accounts from the Twitter space and studied the properties and identification of active honeypots. Yang et al. [297] deployed passive social honeypots to capture spammers’ preferences by designing social honeypots with various behaviors. The design considered tweet behavior (i.e., tweet frequency, tweet keywords, and tweet topics), followed behaviors of famous people’s accounts, and application installation. They analyzed which type of social honeypots has the highest capture rate and designed advanced ones based on their results. They demonstrated that the advanced honeypot could capture spammers 26 times faster than the regular social honeypots.

Pros and Cons: Social honeypots would be highly effective, particularly when it is well deployed to attract targeted attackers. However, the existing studies discussed above did not consider key, unique characteristics of vulnerable victim profiles to develop social honeypots. The effectiveness of existing social honeypots is evaluated based on intrusion detection accuracy rather than the coverage of attack types or the main attack types attracted to the social honeypots. Since an individual honeypot did not target a particular attack, it is unclear what types of attackers are more attractive to certain characteristics of the social honeypots from the existing approaches. In addition, developing social honeypots with fake accounts may introduce ethical issues because the social honeypots itself is also based on deceiving all other users.

2.4 Detection Mechanisms of Online Social Deception

Most existing defense mechanisms against OSD attacks focus on detecting those attacks. We discuss those detection mechanisms based on three types: user profile-based, message content-based, and network feature-based.

2.4.1 User Profile-based Deception Detection Mechanisms

Most profile cloning studies utilized user profiles [135, 143, 232]. Various methods based on user profile attributes were employed to identify cloned profiles by calculating profile similarities. In their work, Kontaxis et al. [143] proposed a three-component approach for detecting profile cloning, consisting of an information distiller, a profile hunter, and a profile verifier. The profile verifier component evaluated the similarity score between a user’s original profile and testing social profiles to determine if any profile cloning had occurred. Kamhoua et al. [135] detected user profiles across multiple OSNs in a supervised learning classifier. The method consists of three steps: the profile information collection from a friend request, the friend list identity verification, and the report of possible colluders. The binary classifier was based on similarities of both the profile attributes and the friend list. Shan et al. [232] sim-
ulated profile cloning attacks by snowball sampling and iteration attacks and then detected the attackers by a detector ‘ChoneSpotter’. The context-free detection algorithm includes profile information and friendship connections. The input features include recently used IPs, a friend list, and the profile and its similarity. A cloned profile was determined using the same IP prefix and the similarity over a certain threshold.

User profile and behavior/activity features were extracted to detect malicious accounts [16, 29, 47, 168, 205, 240, 280] in Sybil attacks, fake reviews, or spamming attacks. Badri Satya et al. [16] studied feature engineering from the account of ‘fake likers’. They considered profile features, such as the length of user introduction, the longevity of an account, and the number of friends. Social activities represent a unique attribute observed in OSNs and consist of the behavior features of an account, such as sending a friend request, posting, retweeting, liking/disliking, and social attention [16]. More specific features under each activity category can be further extracted, such as the acceptance of a friend request sent from [205] and the average time interval of posting from [240]. Wang et al. [280] investigated several behavioral signatures for the output of crowdturfing campaigns and tasks. Cao and Caverlee [29] studied the behavioral features to detect spam URLs in OSNs. They used fifteen click and posting-based features in Random Forest classifiers and evaluated the top six features.

Cresci et al. [47] proposed a novel DNA-inspired social fingerprinting approach of behavioral modeling to detect spambot accounts. Twitter account behaviors were encoded as a string of behavioral units (e.g., tweet, reply, and retweet). This new model can deal with the new type of spambots, which most traditional tools can easily miss. Social fingerprinting sequences are characterized by the LCS curve. Spambots are related to high LCS values by sharing suspicious long behavioral patterns. The LCS curve from the behavioral model is used to detect more sophisticated types of crowdsourcing spammers.

User profiles and activities are the key features to detect OSD attacks (e.g., advanced spammers or crowdturfing), along with other content-based and graph-based features [120, 158, 159, 160, 272, 276]. We will discuss those hybrid detection examples in Section 2.4.4.

Pros and Cons: User profile information provides specific activity features and behaviors about each user. However, some profile information is private; thus, collecting private information violates a user’s privacy right. In addition, even if the information itself is open to the public, their owners should agree on how to use the information. Since each user enters his/her profile information, if the user is malicious, it is easy to enter fake information to make self-presentation look attractive, which is one of self-deception. Besides, collecting profile and behavioral data incurs high costs and/or time under the privacy protection of social media platforms.
2.4.2 Message Content-based Deception Detection Mechanisms

The majority of social deception detection approaches have used content-based features because the text of user posts and reviews can be easily collected and analyzed by linguistic models. The proliferation of social media and/or network applications allowed numerous types of raw and advanced content features to be available. Topic modeling and sentiment-based features have been popularly utilized for the linguistic analysis of deceptive messages.

**Topic Modeling-based Detection**

Most of the work developed topic distributions using Latent Dirichlet Allocation (LDA) [157, 171, 240, 245, 290]. If each user’s posts are collected as a document, LDA generates the topic probability distribution of the user’s document. Liu et al. [171] extended the topic features to two new features. A GOSS indicates a user’s interest in specific topics compared to other users, while a LOSS indicates a user’s interest in various topics. By adding those two topic-based features to classifiers, the averaged F1 score shows better performance. Swe and Myo [245] built a keyword “blacklist” to detect fake accounts by extracting topics from LDA and keywords from TF-IDF (term frequency-inverse document frequency) algorithms. The blacklist contributed to 500 fake words. The number and ratio of fake words and a few other content-based features were extracted for their classifier. The result using a “blacklist” showed better accuracy than the traditional spam word list by reducing the false positive rate. Wu et al. [290] extracted the topic distribution of 18 topics for one message following the official Weibo topic categories. The probability of 18 topics was used as one feature vector for the SVM classifier.

The LDA algorithm has been enhanced to detect cybercriminal accounts and spam. Lau et al. [157] developed a weakly supervised cybercriminal network mining method supported by a probabilistic generative model and a novel context-sensitive Gibbs sampling algorithm (CSLDA). It can extract the semantically rich representations of latent concepts to predict transactional and collaborative relationships (e.g., cybercriminal indicators) in publicly accessible messages posted on social media. Song et al. [240] used Labeled LDA (L-LDA) to indicate the probability of co-occurrence. The latent topics were normalized to topic-based features, which have distinct properties with TF-IDF-generated word-based features.

Golbeck et al. [81] detected two types of false article stories: fake news and satires by themes and word vectors. Then they defined a theme by a new codebook with seven theme types, such as conspiracy theory and hyperbolic criticism. The proposed classifier worked better for articles under a certain type of theme.

**Pros and Cons:** The topic features can be easily obtained. However, there would be unique network features distinguishing attackers from normal users. That is, the content-only features may not be able to capture other features of dynamic interactions with other users, such as likes, friend acceptance, or frequency of leaving comments or sharing. In addition,
topic models are highly sensitive to datasets, and topic models may perform differently in detection accuracy depending on datasets.

**Feature-based Deception Detection**

The commonly used features include raw features, such as word vectors, word embedding, hashtags, links, and URLs [176]. Advanced features include deep content features, statistics, LIWC, and other metadata, such as location, source, or time [95, 261, 263]. Most ML-based models use supervised learning. Among the supervised models, random forest, SVM, Naïve Bayes, logistic regression, and \( k \)-nearest neighbors are the most favorable classifiers for detection. Neural network models, such as Recurrent Neural Networks [301] and Convolutional Neural Networks with Long Short-Term Memory (CNN-LSTM) [300], are used for textural features. Temporal models, such as DTW and HMM [68, 272], are discussed in rumor detection. The boosting-based ensemble models are implemented for spammer detection [120, 300]. A few studies used semi-supervised models [120, 230] when the labeled dataset was unavailable.

Everett et al. [68] studied the veracity of the automated online reviews provided by regular users. They used the text generated by the second-order Markov chain model. The key findings include: (i) The negative crowd’s opinion reviews are more believable to humans; (ii) light-hearted topics are easier to deceive than factual topics; and (iii) automated text on adult content is the most deceptive. Yao et al. [301] investigated attacks of fake Yelp restaurant reviews generated by an RNN model and LSTM model. The model considers the reviews themselves only, not including metadata as reviewers. Similarity features, structural features, semantic features, and LIWC features were used in SVM to compare the character-level distribution. They found that information loss was incurred in generating fake reviews from RNN models, and the generated reviews can be detected against real reviews. Song et al. [239] detected crowdturfing targets and retweets from crowdturfing websites and black-market sites.

**Pros and Cons:** Feature-based models generate high accuracy and low false positive rates. The raw content features are easily obtainable, although extracting sophisticated features incurs a high cost. However, the temporal pattern of messages influences the detection performance. The semantic analysis methods may ignore hidden messages and background knowledge and require tuning many input parameters, which leads to high complexity and labor-intensive.

**Sentiment-based Deception Detection**

The sentiment of social media messages serves as an extra feature of message content. Sentiment provides emotional involvement, such as like, agree, or negation, calculated by lexicon analysis [20, 57, 114, 127, 260, 271]. Jiang and Wilson [127] introduced a novel emotional and
topical lexicon called ComLex. The authors analyzed the linguistic signals in user comments regarding misinformation and fact-checking. Specifically, they discussed the signals from user comments to misinformation posts, the veracity of social media posts, or fact-checking effects. There are signals for positive fact-checking effect as well as signals (e.g., increased swear word usage) indicating potential “backfire” effects [197], where attempts to intervene against misinformation may only entrench the original false belief.

Sentiment features are often used along with TF-IDF word vectors to improve the prediction of the supervised classifiers. Bhatt et al. [20] detected fake news stances from neural embedding, n-gram TF vector, and sentiment difference between news headline-body TF vector pair. Dinakar et al. [57] proposed a sentiment analysis to predict bullying, aiming at discovering the goals and emotions behind the contents. Note that the Ortony lexicon [202] maintains a list of positive and negative words describing the effect. The lexicon of negative words was only added to the feature list to detect bully-related rude comments.

**Pros and Cons:** Sentiment analysis includes more emotional and background information and explicit content, which can increase prediction accuracy compared to semantic-only methods. However, sentiment analysis cannot fully leverage the linguistic information in the contents where the lexicon is domain-specific. In addition, more elaborated dimensions of emotions or sentiments should be considered to capture fake information and its intent.

### 2.4.3 Network Structure Feature-Based Detection

Several general network features were extracted in supervised learning methods, such as topology, node in-degree and out-degree, edge weight, and clustering coefficient [150, 219, 272]. Wu et al. [289] summarized false information spreader detection based on network structures. Ratkiewicz et al. [219] built a Truthy system to detect astroturfing on Twitter. The proposed Truthy system extracted a whole set of basic network features for each meme and sent those features with a meme mood by sentiment analysis to the supervised learning toolkit. Kumar et al. [150] developed four feature sets, including network features, to identify hoaxes in Wikipedia. The network features measure the relation between the references of the article in the Wikipedia hyperlink network. The performance of feature sets was evaluated in a random forest classifier.

In the following sections, we discuss algorithms and supervised learning methods specifically designed for the network structure, such as propagation-based models, graph optimization algorithms, and graph anomaly detection algorithms.

**Epidemic Models**

The epidemic model is a direct way to model and simulate disease diffusion [191]. Since the spread of disease in a certain population is similar to the propagation of false information
(a) SIR Model

- **Susceptible (S)**: Users who have not received information (e.g., rumor posts or fake news) yet but are susceptible to receiving and believe it.
- **Infectious (I)**: Users who received the information and can actively spread it.
- **Recovered (R)**: Users who received the information and refused to spread it.

Figure 2.5: Three types of agent-based epidemic models. The solid line arrows are transitions from one state to another with probabilities. The dotted line arrows are transactions that may not exist at all times. (a) SIR model: $\beta$ is the infection rate, $\gamma$ is the recovery rate, and $\xi$ is the rate of Recovered to Susceptible. (b) SIHR model: $\alpha$ is a stifling rate, $\beta$ is a refusing rate, $\gamma$ is a spreading rate, $\delta$ is a forgetting rate, $\eta$ is a wakened remembering rate, and $\xi$ is the spontaneous remembering rate. (c) SEIZ model: $\beta$ is an infection rate, $\epsilon$ is a self-adoption rate, $\phi$ is a contact rate, and $\xi$ is a skeptic rate. The details of $p$ and $l$ and the whole model were explained in [128].

in social media communities, epidemic models have often been modified to quantify the extent of false information propagation [128]. The epidemic models are agent-based, where an individual node is modeled as an agent. Different types of agents are characterized by distinct states and behaviors, such as the agents Susceptible (S), Infectious (I), and Recovered (R) in the traditional SIR (Susceptible, Infectious, and Recovered) model [187] in false information propagation. In OSNs, agents in the SIR model represent a group of users in each state as follows: (i) **Susceptible (S)**: Users who have not received information (e.g., rumor posts or fake news) yet but are susceptible to receiving and believe it; (ii) **Infectious (I)**: Users who received the information and can actively spread it; and (iii) **Recovered (R)**: Users who received the information and refused to spread it [318].

The state transitions are $S$ to $I$ by infection rate $\beta$, and $I$ to $R$ by recovery rate $\gamma$ depicted in Figure 2.5a. The current false information propagation research has two tracks employing the epidemic models: (i) Adding more links and parameters to the traditional SIR model; or (ii) Building the SEIZ model (Susceptible, Exposed, Infected, and Skeptic–Z; discussed below) to fit the OSN data.

**SIR Model with Variations.** The current false information propagation research proposes many variants of the basic SIR models. Zhao et al. [318] added forgetting mechanisms to the SIR model for rumor spreading so that the spreader ($I$) can be converted to stiflers ($R$). Stiflers are defined similarly to the Recovered state. They used the population size of $R$ to measure the impact of rumor. They found that a forgetting mechanism can help reduce rumor influence, and the rumor saturation threshold can be influenced by the average degree of nodes in the network. Another Hibernator state (i.e., users who refuse to spread rumors...
just because they forgot) was added to the SIHR (Susceptible, Infectious, Hibernator, and Recovered) model [315] to measure the forgetting rate $\alpha$ and a remembering mechanism $\eta$. The new remembering mechanism was proved to delay the rumor termination time and reduce the rumor’s maximum influence. The direct link from $S$ to $R$ was added by [315] and was extended by [316]. The update was that all users in state $S$ were finally converted to either $I$ or $R$ state if they had the chance to be exposed to spreaders ($I$). Figure 2.5a and Figure 2.5b describe the SIR and SIHR models, respectively.

Cho et al. [42] extended the basic SIR model by replacing the transition between states to a decision based on the agent’s belief in the extent of uncertainty in the agent’s opinion. The Subjective Logic opinion model is used to model an agent’s opinion composition and update based on the extent of uncertainty. The three states in the SIR are defined based on the degree of each dimension of an opinion which is defined by belief, disbelief, and uncertainty. The opinion update involved interaction similarity between two agents, a conflicting measure between belief and disbelief, and opinion decay upon no interactions between agents for opinion updates. Based on the degree of uncertainty in a given opinion, an agent’s opinion can move from any state to any other state. This work investigated the effect of misinformation and disinformation on how well false information can be effectively mitigated by propagating countering (true) information by selecting a good set of true informers.

The evolutionary SIR model simulation has been used to model decision strategies in fake news attacks [145]. The state transitions in the SIR model were replaced by the decision model Iterated Prisoner’s Dilemma (IPD). The deception strategies can modify the prior knowledge of the agents by either adding uncertainty or changing false perceptions. In their expensive simulation experiments, only a small population of fake news attackers can initiate the spread, but the fitness of attackers was sensitive to the cost of deception.

**SEIZ Model with Variations.** Jin et al. [128] captured the diffusion of false and true news by the SEIZ epidemic model. Instead of considering the Recovered state, they modeled a state of users being heard of the rumor but not spreading it (Skeptic, Z) and influenced users (E) to post the rumor with an exposure delay. The SEIZ model accurately captured the diffusion patterns in real news and rumors events and was evaluated to be better than the simple SIS (Susceptible, Infectious, and Susceptible) model. They also proposed a ratio $R_{SI}$, the transition rates entering $E$ from $S$ to the transition rates exiting $E$ to $I$, to differentiate rumor and real news events data. Isea and Lonngren [121] extended the SEIZ model by considering a forgetting rate of rumor posts. The forgetting rate is the probability that a user forgets the rumors across all the states. Figure 2.5c shows the key components of the SEIZ model and its process with the states and rates given from one state to another.

**Pros and Cons:** Epidemic models provide a straightforward mathematical model for the diffusion dynamics of false information. The agent density plot with time is a good way of observing the differences between the simulation and real values. However, simulation tests face a common issue as the population size is unknown and stable, and initial variable values are unknown. If the population size is as large as the real social media network,
the computational cost cannot be ignored. In addition, in the SIR model, the state change is controlled by probability; but this autonomous behavior ignores a user’s intention and belief. To complement this, there have been some efforts [42, 145] focusing on modeling and evaluating the effect of subjective, uncertain opinions and trust of agents and the role of more agents in terms of false information diffusion.

Credibility-based Models

In OSNs, one of the detection mechanisms for false information attackers, Sybil accounts, or spammers is modeling the credibility score in the network [129, 130, 305]. Existing works used various ways to represent credibilities, such as reputation, trust, and belief scores. Credibility in OSNs can be modeled by classification-based and credibility propagation methods. A classification-based approach uses supervised learning algorithms [189]. On the other hand, the credibility propagation approach constructs a network to propagate credibility scores among users, tweet contents, events, and activities [129, 174]. Based on the credibility scores, ranking algorithms of users and posts can be conducted, such as PageRank [5, 38, 79, 305].

Negm et al. [189] used 5Ws (i.e., who, what, when, where, and why) credibility to distinguish credible news and RSS (Rich Site Summary) files from news agencies to extract publication dates, headlines, contents, and locations to feed into different algorithms to calculate the credibility of a news agency. The compared algorithms include TF-IDF, TF-IDF with location, Latent Semantic Index (LSI), and TF with LSI and log entropy. They concluded that TF-IDF and TF-IDF with location performed the best in calculating credibility. More recently, Norambuena et al. [196] leveraged the 5W1H extraction and news summarization techniques to propose the Inverted Pyramid Score (IPS) to distinguish structural differences between breaking and non-breaking news, with the long-term goal of contrasting reporting styles of mainstream and non-mainstream fake outlets.

Jin et al. [129] have introduced a credibility propagation network for news content composed of three layers: message, sub-event, and event. The event layer talks about the main event the news covers, the sub-event layer relates events to the main event, and the message layer holds the content of the news article. A graph optimization problem is formulated to calculate the credibility of this hierarchical network. All the layers are content-based and have direct relations with the credibility of the news. Jin et al. [130] further proposed a verification method for credibility in a propagation model using a topic modeling technique. Mitra and Gilbert [182] created the CREDBANK corpus by monitoring tweets, topics, events, and associated in-situ human credibility judgments associated with them, to systematically study the credibility of those events in real time. Subsequently, they utilized this corpus to develop language and temporal models for assessing credibility [183, 184]. By identifying linguistically-grounded dimensions, the authors presented a parsimonious model that links language cues to credibility levels. For instance, hedge words and positive emotional language were linked to lower credibility. Additionally, by examining the temporal dynamics of the event reportages, they found that the amount of continued collective attention given to an
Akoglu et al. [4] proposed the OddBall algorithm to detect anomaly behavior like malicious posts and fake donations. They studied a sub-graph (ego-nets) of a target node with its neighbors. They analyzed various scoring and ranking methods using feature patterns in density, weights, principle eigenvalues, and ranks and compared their performance in different network topologies.

Kumar et al. [152] detected fake reviewers in user-to-item rating networks. They developed a new trust system to rank users, products and ratings by fairness, goodness, and reliability. The intrinsic scores are calculated by combining network and behavior properties. Users rated with low reliability are more likely to be fake reviewers [152]. Akoglu et al. [5] developed the FraudEagle algorithm to spot fraudsters and fake reviews on online review platforms. There are two steps in the FraudEagle algorithm in scoring users and reviews and grouping the analyzed results. For each review, the sentiment from true and false is only analyzed to assign the belief score. The grouping step reviews top-ranked users in a subgraph by clustering and merging more evidence to reveal fraudsters.

Ghosh et al. [79] developed the CollusionRank algorithm for detecting link farming-type spammer attacks. The influence scores were given to the users and web pages. By decreasing the influence scores of the users connected to spammers, the follow-back behavior of social capitalists was discouraged. Yu et al. [305] developed the SybilLimit ranking algorithm for detecting Sybil attacks. A Sybil node was identified by calculating the node’s trust score. Chirita et al. [38] developed the MailRank algorithm for detecting Sybil attacks in the email network. A sender is assessed by a global and personalized reputation score.

**Pros and Cons:** Credibility models can be applied in different stages and levels based on content, user behaviors, and posts/comments in highly heterogeneous networks. In addition, a credibility model based on network features is agnostic to platforms and languages because the model only needs network features. However, accurately evaluating initial credibility values is not a trivial problem. Considering credibility at multiple levels makes the computation more complex and expensive, so it may not be preferred. Further, credibility may be subjective and cannot be ported across platforms or networks. Lastly, a credibility model may not be able to detect sudden changes caused by instances that are not easily observable, thus impacting the accuracy of the credibility score assessment.

**Cascades Features-based Models**

Information network propagation patterns can be represented by a cascading structure depicting the OSD information flow that users time-traveled through, posted, tweeted, and retweeted. The cascading structure has two forms: hop- and time-based cascades [320]. The cascade features can be grouped into two approaches: (i) Calculating the similarity of cascades between true and false information, and (ii) representing cascades using informative
representation and features in a supervised learning model.

**Cascades Similarity.** Cascades similarity is computed between fake news and true news. A graph kernel [320] was used as a common strategy for its computation. Wu et al. [290] proposed a fake news detection method using a hybrid kernel function. This graph kernel function calculates the similarity between different propagation trees. It also discussed the Radial Basis Function (RBF) kernel, which calculates the distance between two vectors of traditional and semantic features. The sentiment and doubt scores for user posts must be verified for fake news. Ma et al. [172] proposed a top-down tree structure using RNNs for false information detection. The RNN learns the representation from tweet content, such as embedding various indicative signals hidden in the structure to improve rumors identification.

**Cascades Representation.** Cascades representation pursues informative representation as features to distinguish fake news from true news. For example, the number of nodes is a feature in a non-automated way. Alternatively, cascades representation can fit deep learning models [292]. Wu and Liu [292] used LSTM-RNN to model the propagation cascades of a message. This work combines the propagation pathways with user embedding, which forms a heterogeneous network. A sequence of its spreaders represents a message. A modularity maximization algorithm is used to cluster nodes with embedding vectors. Ma et al. [173] proposed propagation trees using Propagation Tree Kernel (PTK) for rumor detection. It can explore the suggested feature space when calculating the similarity between two objects.

**Pros and Cons:** Similarity-based approaches consider the roles of users in propagating false information. Computing similarities between two cascades may require high computational complexity [320]. Representation-based methods automatically represent news to be verified; however, the depth of cascades may challenge such methods as it is equal to the depth of the neural network. All the approaches only provided experimental data to show their effectiveness. However, it may not correctly reflect real-world settings. Training data is a time-consuming process and is often computationally expensive.

**Game-Theoretic Models**

Game models explore the deception and defense by reward and penalty model in OSD attacks. In game theory, the actions and decisions of the players are mainly based on the reward and penalty of their previous activities and the other players’ actions [248].

Kopp et al. [145] discussed a game theoretic false information propagation model as a deception model that simulates the propagation of fake news in the OSNs. They used three types of game theories: Greenberg’s deception model [84], Li and Cruz’s deception model [163], and hypergame theory [18]. Greenberg’s deception model investigated the effect of deception on players’ payoffs [84]. Kopp et al. [145] mapped false information to Greenberg’s false signal model. Li and Cruz [163] used passive and active deception strategies by introducing noise and randomization to increase uncertainty. Kopp et al. [145] used the deception
game in [163] to consistently monitor constraints and conditions affecting game strategies. Bennett and Dando [18] used hypergame theory to model a deception game where players had subjective perceptions and understandings of a complicated game. Kopp et al. [145] also used [18] to consider players' subjective beliefs, which may also introduce uncertainty. Kopp et al. [145] proposed the information-theoretic model that attackers' deceptive behavior can be significantly mitigated when deception costs are relatively expensive.

**Pros and Cons:** Game theoretic approaches to model OSD attacks add extra features over other conventional network structure-based approaches above by considering the cost and benefit of performing a deceptive behavior by users in OSNs. Game theoretic deception detection is a promising approach that reflects human behaviors aiming to take an optimal action based on the expected outcome. However, game theoretic approaches have been rarely adopted in modeling and analyzing online social deceptive behaviors compared to data-driven deception detection approaches. Due to this reason, the effectiveness of game theoretic deception detection approaches has not been thoroughly investigated in the literature. In addition, aligned with a conventional drawback in using game theory, a large number of deceptive actions may introduce a high solution complexity. Uncertain, subjective beliefs of users should be carefully considered in terms of modeling incomplete information and/or imperfect information in game theory.

**Blockchain-based Models**

Huckle and White [118] developed a tool called Proventor to prove the origin of the media. The Proventor is based on Blockchain storing provenance metadata for users to trust the authenticity of the metadata. Proventor can be used to validate news for news outlets like CNN and BBC, where information and news are sometimes gathered from independent sources. However, since Proventor uses Blockchain and cryptography, a slight difference, such as a one-pixel difference between two images, can make the result vastly different, leading to generating numerous false alarms and human interventions for validation, which is labor-intensive. McEvily et al. [178] proposed a social media platform called Steem (i.e., a database) based on Blockchain technology for building a community reward system. The reward system relies on users for consensus voting, reading content, and commenting.

**Pros and Cons:** The original design of Blockchain has security benefits in terms of provenance, integrity, and immutability. The Blockchain system is a heterogeneous network incorporating other stakeholders to detect and control OSD activities. In addition, it is resilient against OSD attacks. Managing the large ledger size in Blockchain is an issue as shared information in social media and news outlets grows exponentially. Since both flagging accuracy and consensus verification rely on the contribution of crowd signals, it may break when too many users are malicious. For example, if a large volume of attackers contribute to the crowd activities and even control the system, a user cannot access write transactions. In addition, the authorized party may be compromised by advanced attackers.
Other Network Optimization Models

Several graph optimization algorithms were proposed for anomaly detection and community detection problems. Hu et al. [113] developed a matrix factorization-based algorithm to detect social spammers on Twitter. Their framework utilized both content information and network information of an adjacency matrix and solved a non-smooth convex optimization problem. Several approaches have been taken to detect link farming attacks via network structure-based algorithms. Araujo et al. [13] detected temporal communities in cell and computer-traffic networks based on the Tensor analysis. Jiang et al. [126] detected behavior patterns in OSNs where the spectral subspaces had different patterns and lockstep behaviors. In addition, Jiang et al. [125] identified synchronized behaviors from spammers. Kumar et al. [149] considered trolling as a social deception activity. They proposed a decluttering algorithm to break a network into smaller networks where the detection algorithm could be run. Kumar et al. [151] considered sockpuppets an OSD attack where users created multiple identities to manipulate a discussion. They found that sockpuppets could be distinguished from normal users by having more clustered egonets.

Pros and Cons: Graph-based features are more available than user profiles or interaction features without violating privacy issues. In addition, graph-based algorithms can be agnostic to any datasets with high applicability in diverse platforms. However, collecting graph-based features, such as centrality measures, and solving graph optimization often incurs high computational overhead, which hinders its applicability to platforms requiring real-time or lightweight streaming data detection.

2.4.4 Hybrid Detection

Since ML/DL-based models can take abundant features, one can train a hybrid feature set combining the user profile, message content, and network features to detect OSD attacks. Unlike several existing survey papers which discussed only individual feature categories [148, 291], our discussion will focus on dealing with OSD attacks using hybrid features [120, 158, 159, 160, 272, 276].

Lee et al. [159] detected crowdturfers from Twitter users. A total of 92 features were divided into four groups: User demographics, user friendship networks, user activity (behavior-based features), and user content similarity, including linguistic features from the LIWC dictionary. Vosoughi et al. [272] developed a tool called Rumor Gauge for automatically verifying rumors and predicting their veracity before they are verified by trusted channels. Since rumors are temporal, time-series features are extracted as the rumor spreads. A total of 17 features (e.g., linguistics, user involved, and propagation dynamics) were studied. They found that the fraction of low-to-high diffusion in the diffusion graph is the most predictive feature to represent the veracity of rumors. The time-series features are processed in DTW and HMM models, but DTW assumes all the time series are independent and assigns equal weight to all 17 features. The experiment evaluated the performance of the Rumor Gauge in terms of
2.5 Response Mechanisms to Online Social Deception

In this section, we survey existing mitigation or recovery mechanisms after OSD attacks are detected, along with early detection mechanisms of OSD attacks [57, 72, 289]. Florêncio and Herley [72] developed a mitigation strategy to deal with compromised accounts by detecting password reuse events and timely reporting them to financial institutions. The aftermath actions were to take down identified phishing sites, restore the compromised accounts, and rescue users from bad decisions.

Dinakar et al. [57] took a mitigation action to counter cyberbullying with two steps: (i) early detection; and (ii) reflective user interfaces that popped up notices and suggestions on user behaviors. Most efforts to mitigate OSD attacks in OSNs mainly focused on reducing the effect of false information propagation. Wu et al. [289] summarized two misinformation intervention methods: (i) detecting and preventing misinformation from spreading in an early stage; and (ii) developing a competing campaign to fight against misinformation. To limit the spread of fake news, a sample of fake news with maximal utility was identified in [256]. Within a certain constraint, this sample of fake news kept the largest number of users away from fake news posts. Their algorithm was robust against a high amount of spammers. Huckle and White [118] also made an effort to mitigate fake news spread based on the validity proof of digital media data, such as a picture in the fake news. Blockchain technology was used to prove the origins of digital media data; however, this method cannot prove the authenticity of the whole news article. Kumar and Shah [148] summarized misinformation mitigation by modeling true and false information. From the existing four different approaches, the authors concluded that these algorithms effectively detect the spread of rumor, and their simulations could suggest rumor mitigation strategies. Okada et al. [199] studied rumor diffusion using a SIR-extended information diffusion model and developed a mitigation mechanism to ask highly influential users to spread correction diffusion. The authors examined how false rumors diffused and converged when help or correct information was given and how fast the convergence appeared.

Pros and Cons: Mitigation and recovery mechanisms relied heavily on early detection. The simulation model of spreading true information can mitigate the negative influence. However, most studies are based on simulation models, limited in using real-world datasets,
or have not been validated based on the implementation in real-world platforms. Although it is highly challenging for the developed model to be deployed in real platforms, there should be more efforts to use empirical, real datasets to validate the recovery models. Recovery in OSNs is more complicated than in offline social networks because the relationships can be easily dropped. Only one research [72] designed a system for account restoration. More research efforts should be made to effectively mitigate the aftermath actions upon early detection.

Table 2.4: Defense Mechanisms Surveyed To Deal With Online Social Deception Attacks

<table>
<thead>
<tr>
<th>Technique</th>
<th>OSD Attacks Considered</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Attack Prevention</strong></td>
<td></td>
</tr>
<tr>
<td>Data-Driven</td>
<td>Fake news, phishing, fake profile, cyberbullying</td>
</tr>
<tr>
<td>Social honeypots</td>
<td>Spamming, fake profile, socialbot</td>
</tr>
<tr>
<td><strong>Attack Detection</strong></td>
<td></td>
</tr>
<tr>
<td>User profile</td>
<td>Rumor, fake review, spam, fake profile</td>
</tr>
<tr>
<td>Message content</td>
<td>Fake news, rumor, fake review, phishing, spam, fake account, compromised account, crowdturfing, cyberbullying</td>
</tr>
<tr>
<td>Network structure</td>
<td>Fake news, rumor, fake review, false information, spam sybil attacks, crowdturfing</td>
</tr>
<tr>
<td><strong>Attack Response</strong></td>
<td></td>
</tr>
<tr>
<td>Early detection</td>
<td>Compromised accounts, cyberbullying, false information propagation</td>
</tr>
<tr>
<td>Information propagation</td>
<td>False information (or fake news), spamming</td>
</tr>
<tr>
<td>mitigation</td>
<td></td>
</tr>
<tr>
<td>Blockchain-based authenticity</td>
<td>Fake news, rumor</td>
</tr>
</tbody>
</table>

TABLE 2.4 summarizes the classification of OSD defense mechanisms, including prevention, detection, and mitigation/response discussed in Sections 2.3–2.5. Existing works primarily focused on detecting OSD attacks we classified in Section 2.2. Less attention has been paid to prevention and mitigation, where the main focuses include false information, luring, and identity theft. There are still open questions about building trustworthy cyberspace against human-targeted attacks, especially for protecting children.
Chapter 3

SAFER: Social Capital-based Friend Recommendation to Defend Against Phishing Attacks

This chapter provides the SAFER framework to quantitatively model three dimensions of social capital and investigate the effects of resistance to phishing attackers by social capital on a friend recommendation system. The work in this chapter partly addresses the Combating Task based on our published paper [91].

3.1 Motivation & Research Goal

Cyber attackers have heavily utilized highly advanced social media technologies as convenient tools for deceiving people in online worlds [89]. In particular, the significant growth of phishing campaigns, including broad spear-phishing campaigns and targeted attacks, has introduced serious cybersecurity challenges [262]. Phishing campaigns had a growth rate of 15% from 2019 to 2020 among all emails, and a significant number of them are pandemic-related [284]. Phishing attackers often distribute fake URLs to steal sensitive information or financial credentials. In addition, they compromise real accounts and use them to launch attacks using the real users’ accounts. Cybercriminals exploit advanced phishing strategies to cause the loss of confidentiality, privacy, credibility, and financial loss of victims, including organizations.

The concept of social capital (SC) has been extensively studied in the social sciences to understand how and why entities (e.g., either individuals or organizations/communities) participate in social networks. Social capital is a powerful mechanism by which an entity utilizes its interpersonal relationships in social networks to achieve its goals [215]. The concept of social capital has also been leveraged in the computing and engineering domains [7, 21, 134, 211, 264]. However, the concept of social capital has not been applied to combat phishing attacks in online social networks (OSNs).

People often unconsciously use social capital as promising signals to make good friends on OSNs, who may help in various ways, such as being connected with valuable resources (e.g., for marketing or job hunting) or having critical help (e.g., quickly finding missing
people or collecting donations to help someone). However, suppose a bad friend is in one’s social network. In that case, it may perform various types of online social attacks, such as sending phishing or scam messages aiming to obtain private information or monetary benefits. Hence, when people decide whether to accept a new friend invite, they often check a number of (mutual) friends, posts, and likes from the inviter’s friends [294], which partially explains social capital as an indicator of maintaining trustworthy social connections.

The major vulnerabilities to OSN attacks are closely related to how users make friending decisions, e.g., whether to accept or decline a friend invite or send a friend invite (i.e., accept/decline or send a friend invite). A user’s friending decision process determines the characteristics and quality of that user’s social network in the presence of various types of OSN attacks [89]. The majority of friend recommendation systems (FRSs) [88, 116, 194, 283] have been proposed to recommend friends to maximize users’ satisfaction in making friends in OSNs. However, no FRSs have used the multidimensional concept of social capital, in terms of relational, structural, and cognitive capital, to combat phishing attacks in OSNs.

In particular, we are motivated to use a user’s features representing social capital to make good friends who can help users defend against phishing attacks. First of all, we will use social capital as a signal to select good friends. Second, even if an attacker becomes a friend of legitimate users by performing social capital manipulation attacks (e.g., an attacker can increase its social capital through connections with many other users and active interactions with other users), the attacker can be caught by other legitimate users who are friends of the attacked users. The reported attacker accounts by legitimate users or legitimate friend users can be suspended by the OSN system. We name this approach Social cApital-based FriEnd Recommendation, called SAFER.

Our proposed SAFER can be applied in a real OSN platform like Twitter or Facebook. When user $i$ receives a friend request from user $j$, SAFER can provide $j$’s social capital in terms of three dimensions, including structural, cognitive, and relational social capital [188]. We provide the details of SAFER in Section 3.6.

### 3.2 Research Questions

This study aims to answer the following research questions:

1. How can a user’s social capital be quantified in three key dimensions (i.e., relational, cognitive, and structural capital) based on the network and behavioral characteristics?

2. What dimension of social capital defends against phishing attacks more (or less)?

3. How does an attacker’s type, such as attacks by humans or bots, affect the FRS’s capability to defend against phishing attacks differently?
3.3 Key Contributions

In this work, we make the following key contributions:

- To the best of our knowledge, our work is the first to leverage the concept of multidimensional social capital to model its defense capability against phishing attacks and evaluate its effectiveness.

- We quantify multiple dimensions of a user’s social capital (i.e., structural, cognitive, and relational capital) based on behavioral characteristics in OSN contexts derived from two real Twitter datasets where attackers are bots [46] or humans [295, 296].

- We develop a social capital-based friend recommendation system called SAFER and compare its performance with that of existing counterparts, such as social attribute-based [88], topic-based [283], and trust-based [40]. In addition, we conduct simulation-based extensive experiments to identify the key dimension of social capital contributing significantly to combating phishing attacks.

3.4 Background & Related Work

Concepts and Applications of Social Capital. The common role of social capital has been agreed as ‘a vehicle to facilitate achieving individual or collective goals through personal relationships in social networks’. Although there have been many classifications of social capital [215], its most common concept is discussed in terms of bonding and bridging [25]. Bonding refers to being connected with people one can trust, while bridging is connecting with more people [25]. However, there have been more discussions on the concept of social capital in a broader sense. Nahapiet and Ghoshal [188] introduced three key dimensions of social capital: structural, cognitive, and relational. Structural capital (STC) refers to the capital derived from social structure (e.g., network ties and configuration, roles, rules, precedents, and procedures). Cognitive capital (CC) indicates the benefit of shared understandings (e.g., shared language, codes, narratives, shared values, attitudes, and beliefs). Relational capital (RC) is obtained from nature and indicates the quality of relationships (e.g., trust and trustworthiness, norms and sanctions, obligations and expectations, or identity and identification).

The benefits of social capital have been discussed in individual capital (e.g., personal or micro) or collective capital (e.g., society or macro) [215, 299]. The concept has been used for increasing virtual team productivity [21], establishing ‘bridging social capital’ in OSNs [264], investigating mathematical models of online social activities [7], or examining its effect on mental health [211]. However, to the best of our knowledge, social capital has not been used to combat online deception, such as phishing attacks. In addition, no prior work has
Table 3.1: Key three dimensions of social capital and their key characteristics

<table>
<thead>
<tr>
<th>Description</th>
<th>Structural capital</th>
<th>Cognitive capital</th>
<th>Relational capital</th>
</tr>
</thead>
<tbody>
<tr>
<td>Social structure (bridging)</td>
<td>Shared understandings</td>
<td>Quality of relationships (bonding)</td>
<td></td>
</tr>
<tr>
<td>Collective (Macro)</td>
<td>Collective &amp; Individual (Mezo)</td>
<td>Individual (Micro)</td>
<td></td>
</tr>
<tr>
<td>Node/graph centrality metrics</td>
<td>Positive shared experiences</td>
<td>Trust, reputation, and homophily</td>
<td></td>
</tr>
</tbody>
</table>

quantified three-dimensional social capital as a quantitative metric, although social science literature has extensively discussed it conceptually or measured it qualitatively to some extent [188].

Social capital has been measured quantitatively in social sciences by conducting empirical studies with human subjects [141, 175] or in laboratory environments [136]. Network scientists also measured structural social capital based on network topologies [10, 255]. Further, social capital in online platforms has been measured in Facebook and Twitter in the computer science domains [40, 210, 302]. However, none measured three-dimensional SC nor employed it as a defense mechanism to handle phishing attacks. The key features that can be considered for three-dimensional social capital are summarized in Table 3.1. Those features are also used in other works, showing the validity of those features in measuring social capital.

Friend Recommendation Systems (FRSs). An FRS has been extensively studied for matching similar friends. A personality-trait-based FRS, called PersoNet, was proposed [194], and a similarity metric was considered to weigh the contribution from both Big Five personality traits similarity and harmony rating similarity. Their online experiment tested the friend recommendation accuracy of a small set of selected users for one month. A two-stage FRS was proposed in [117]. The first stage generated a friend list by the network alignment algorithm of a contact network and tag-similarity network. The second stage used topic features from Flickr image information in a probabilistic topic model.

Wang et al. [283] discussed life style extraction as topics and word analysis by latent Dirichlet allocation (LDA) topic model and clustered by K-means. The similarity metric considered both the topic vectors and the dominant topics based on an edge weight in a friend-matching graph. Zheng et al. [319] analyzed a user’s temporal behaviors by a topic model to predict potential friends. A user’s temporal similarity was calculated at different time intervals by the topics at that time, where the friend recommendation value was estimated based on the aggregation of the temporal similarity with a time decay function, addressing more weight on the most recent similarity.

Cheng et al. [37] and Guo et al. [88] focused on the privacy-preserving property of friend recommendation in the social network in terms of encryption and decryption directed by a center authority. The similarity between users was based on tag matching [37], while the
cosine similarity was estimated based on social attributes and trust level in a multi-hop friend recommendation chain [88]. Cho et al. [40] used social capital to investigate a tradeoff between social capital and privacy preserving. However, to the best of our knowledge, no prior work has investigated users’ online social capital to combat phishing attacks.

Machine learning-based FRSs are also studied in the literature. Chen et al. [33] used the preferences and behaviors of informative friends as key features and ranked them based on gradient descent for those features to be matched to the preference of a target user. Ding et al. [58] also proposed a model called BayDNN based on structural features and used a Bayesian ranking to recommend new friends. Chen et al. [34] used a convolutional neural network (CNN) for user embedding in a graph convolutional network to process users’ and their neighbors’ features and recommended new friends based on the Bayesian ranking. Our approach differs from ML-based approaches because its goal is not to predict an attacker accurately but to guide users to form their own social capital features and build a safe and trustworthy online social network against phishing attacks.

3.5 Preliminaries

We consider an online social network (OSN) defined by a graph $G = (V, E, T)$ where $V$ refers to a set of vertices, $v_i$’s, representing users $i$’s and $E$ is a set of edges $e_{ij}$ for users $i$ and $j$, indicating that users $i$ and $j$ are friends. $T$ is a set of edge weights $T_{ij}$, referring to user $i$’s trust in user $j$ (detailed in Section 3.6.1). How users are connected is reflected in $E$ and $T$.

3.5.1 User Behavior Modeling

In this section, we discuss how each user’s behavior in a given OSN is modeled based on the features obtained from the two real datasets (i.e., Cresci15 [46] and 1KS-10KN [295, 296]). Note that a user’s friend list is unavailable due to privacy issues. In addition, this work aims to investigate how social capital-based friending decisions can ultimately help defend against phishing attacks. Therefore, instead of deriving user behaviors and their friending decisions in the real datasets, which is not feasible per se, we model a user’s behaviors based on the behavior trends observed from the datasets, as commonly used in the simulation and modeling research [41, 77]. A user in a given OSN is modeled by the following characteristics used to interact dynamically with other users in our simulation model.

**Feeding Information.** A user can feed information to other users by: (i) *posting his/her own information* (e.g., opinions/thoughts, knowledge in professional domains, preferences, activities); (ii) *sharing a third-party’s information* (e.g., news articles, posts by others, information from other third-party sources, such as blogs, websites); or (iii) *adding his/her own opinion* to a third-party’s information. The information can be texts, images/photos,
CHAPTER 3. SAFER: SOCIAL CAPITAL-BASED FRIEND RECOMMENDATION TO DEFEND AGAINST PHISHING ATTACKS

and/or videos. This work derives a user’s feeding behavior as behavioral seed probabilities from real social media datasets [46, 295, 296]. For example, based on the frequency of posting or sharing information, the probability of a user feeding information is calculated and used to exhibit his/her feeding behavior in our simulation model (e.g., tweets on Twitter).

Providing Feedback. A user can show his/her preferences or express his/her opinions towards the posts by his/her friends (e.g., ‘Likes’ on Facebook or ‘Favorities’ on Twitter). In addition, the user can leave comments on his/her friends’ posts or shared information. The frequency of leaving comments, ‘Likes’ or ‘Favorities’ can measure how often the user provides feedback to other users.

Inviting Friends. The frequency of inviting friends can differ depending on a user’s propensity to make friends. Based on real datasets that provide the number of friends per user, we derive the probability that a user invites a friend in terms of the number of friends the user has over the total number of users. However, it is difficult to know what types of friends a user prefers to be a friend because users’ privacy settings restrict the availability of datasets on friend-friend relationships. Hence, we evaluated a set of FRSs, including our proposed SAFER in Section 3.7, which governs friend relationships.

Capability to Detect Phishing Attacks. We model the user i’s probability of detecting attacker j based on the following factors: (i) an individual user’s competence (ci) to detect source credibility based on the number of followers. Based on Westerman et al. [286], a user can best detect source credibility when the number of followers is not too high or not too small. To reflect this, we modeled the degree of a user’s capability to judge source credibility by: $c_i = -\lambda(f_i - f_{\text{median}})^2 + 3$, which is ranged in $[0, 3]$ for $c_i \geq 0$ where $\lambda$ is a constant to adjust the range of observed number of followers in all users of a given OSN; (ii) depending on the quality of deception skills a phishing attacker j uses, it may be easier or harder for user i to detect attacker j. We denote $d_j$ as the degree of deception quality used in a given phishing message by attacker j (see the detail on how $d_j$ is modeled in Section 3.5.2); and (iii) how many times user i has experienced phishing attacks, denoted by $g_i$, also affects user i’s ability to detect attackers. Considering these three factors, we model the user i’s probability to detect attacker j by:

$$P_{ij}^{\text{crd}} = e^{-\frac{d_j}{(c_i + g_i)}}.$$ (3.1)

Behavioral Features Related to Social Capital. In this work, we estimate a user’s social capital (SC) in terms of structural capital (STC), cognitive capital (CC), and relational capital (RC). According to [188] (1998), we collected features to measure each dimension of social capital from two real datasets, Cresci15 [46] and 1KS-10KN [295, 296], where the Cresci15 dataset has features of bots attackers, while the 1KS-10KN dataset has features of humans attackers along with the majority of legitimate users’ features. Features used to measure each dimension of social capital are described as follows:
Table 3.2: Features used to quantify dimensions of social capital, trust, and friend network in the Cresci15 [46] and 1KS-10KN [295, 296] datasets.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>$h_i$ for STC</td>
<td>Account longevity, # of self-introduction words</td>
</tr>
<tr>
<td>$cc_i$ for CC</td>
<td>Average # of retweets per post, average # of hashtags per post, average # of mentions per post, average # of URLs per post, # of followers</td>
</tr>
<tr>
<td>$rc_i$ for RC</td>
<td>Average # of tweets per day (post_freq), average # of replies per day (com_freq), total # of favorite tweets (favorite frequency)</td>
</tr>
<tr>
<td>Friend network</td>
<td>Friend count (friend_num), a user’s friends, followers count</td>
</tr>
<tr>
<td>$T_{ij}$</td>
<td>Average # of tweets per day (post_freq), average # of replies per day (com_freq)</td>
</tr>
</tbody>
</table>

- **Structural capital (STC):** User $i$’s structural capital is measured based on the extent of individual bridging social capital of all the friends of user $i$. The individual bridging social capital is called *human capital*, representing a graph node. Accordingly, the user $i$’s STC is all of the friends $j$’s human capital $h_j$ from the social structure or friend network. Hence, a user’s STC is mainly affected by (i) how many friends the user has; and (ii) the extent of human capital the user’s friends have, where the human capital is measured by the friend’s individual capability. Due to the limited user features identifiable in various social media platforms (e.g., Facebook or Twitter), a user’s human capital is measured based on the longevity of the user’s account (i.e., the user’s stable activity and/or relations with other users) and the length of self-introduction (i.e., how well the user self-introduces himself/herself) where these two features are identified as key features representing trustworthy users in the literature [16, 120].

- **Cognitive capital (CC):** User $i$’s CC is measured based on how much his/her friends support or like his/her posts or activities. A user’s CC is mainly measured based on (i) the amount of feedback received (e.g., the average number of shares or retweets per post, likes per post, or replies per post); (ii) the broadness of interest in posts (e.g., hashtags/URLs/mentions per post), and (iii) the number of followers.

- **Relational capital (RC):** User $i$’s RC is measured based on how actively user $i$ interacts with other users. The main activities to represent a user’s RC are: (i) The total amount of posts; (ii) the frequency of posting, post_freq (e.g., the number of uploading posts/tweets per day); or (iii) the frequency of providing feedback, com_freq (e.g., the number of providing comments, replies, likes, or favorites per day or month).

Table 3.2 summarizes the features used to estimate the three social capitals from the real datasets [46, 295, 296]. Since no existing dataset is perfect, we use the most relevant user profile or user behavioral features to approximate the three social capital dimensions based...
on the intuitions of description in Table 3.1. In Table 3.2, STC features are from user profile attributes; CC features are from the response or interactions with other users; and RC features are from the user’s social posting or liking behaviors.

3.5.2 Adversary Model

A phishing attacker may trick users into revealing sensitive, private, or confidential information related to work, financial credentials, or even personal data to be used in fraudulent activities, leading to the loss of confidentiality and/or privacy. We modeled random phishing attacks by an attacker simply selecting a set of users among its friends (i.e., its friend network) at random and disseminating phishing/scam messages. We assume the attacker may send various types of phishing messages applying different levels of deception quality. A legitimate user may be unable to easily detect a phishing message with high-quality deception and vice versa. To model this, attacker $j$ will apply a different level of deception quality, ranging in $[0, 3]$ as a real number, respectively, denoted by $d_j$. We model each attacker’s deception quality following Gaussian distribution with a given mean and standard deviation. When a legitimate user receives the phishing message with deception level $d_j$ from attacker $j$, the user’s ability to detect this attack is estimated based on Eq. (3.1), implying that higher $d_j$ decreases the user’s attack detection ability. The attacker may send the same type of phishing message to other users.

The attacker randomly selects new friends and invites them to accept their friend invites. When the attacker receives any friend invite, it will always accept to maximize its influence with more friends in OSNs, increasing its social capital. In addition, the attacker can invite many friends to increase its own social capital. This may lead other legitimate users to invite the attacker more frequently. Further, the attacker can actively interact with its friends by sharing, commenting, or posting information more frequently. These kinds of attack behaviors are to manipulate attackers’ social capital. Further, the attacker may compromise legitimate users’ accounts and let the compromised accounts perform phishing attacks, which can ruin the legitimate user’s reputation. This may lower the user’s social capital because some friends may terminate their relationships when detecting phishing attacks.

A legitimate user can make friends through FRSs. We discuss our proposed social capital-based FRS, namely SAFER, as detailed below.

3.6 Proposed Approach: SAFER

This section details the proposed SAFER (Social Capital-based Friend Recommendation) scheme. We first describe our proposed multidimensional social capital metric. In addition, we provide the details of how social adversaries are detected in the SAFER and how a
3.6. Proposed Approach: SAFER

user’s status is updated based on the proposed SER-SEIR (Susceptible, Exposed, Recovered-Susceptible, Exposed, Infected, and Recovered) model.

3.6.1 Quantification of Social Capital

We already described what features to measure the three dimensions of social capital in ‘Behavioral Features Related to Social Capital’ of Section 3.5.1. This section focuses on mathematically formulating each dimension of social capital as a metric. A user’s structural capital, \( STC_i \), measures how well user \( i \) is connected to other users with high human capital. We denote the node \( i \)’s human capital by \( h_i \). A user’s cognitive capital, \( CC_i \), measures how much the user \( i \)’s interactions with other users are supported (preferred) by his/her social network community. A user’s relational capital, \( RC_i \), is measured based on how actively user \( i \) interacts with other users (i.e., supports or activities provided to other users).

We measure the user \( i \)’s social capital, \( SC_i \), by:

\[
SC_i = w_{STC} \cdot STC_i + w_{CC} \cdot CC_i + w_{RC} \cdot RC_i,
\]

where \( CC_i \) is the user \( i \)’s cognitive capital, which is measured based on the extent of the support the friends of user \( i \) have received from their friends. \( w_X \) is a weight to consider each social capital where \( X = STC \) (structural capital), \( CC \) (cognitive capital), or \( RC \) (relational capital). Given \( cc_j \) representing the extent of the support friend user \( j \) received from its friend network, user \( i \)’s cognitive capital, \( CC_i \), is computed by:

\[
CC_i = \frac{1}{|F_i|} \sum_{j \in F_i} T_{ij} \cdot cc_j,
\]

where \( T_{ij} \) refers to the user \( i \)’s trust in user \( j \). We discuss \( T_{ij} \) later in this section with Eqs. (3.6) and (3.7). \( F_i \) is the set of user \( i \)’s friends. \( CC_i \) is scaled as a real number in \([0, 3]\) because \( cc_j \) considers three features, each normalized as a real number in \([0, 1]\) based on Table 3.2. \( STC_i \) is estimated from the human capital of \( i \)’s friends by:

\[
STC_i = \frac{1}{|F_i|} \sum_{j \in F_i} T_{ij} \cdot h_j,
\]

where \( h_j \) refers to the user \( j \)’s human capital described in Section 3.5.1 and is derived based on three features, each normalized as a real number in \([0, 3]\). Given \( rc_j \) indicating how actively friend user \( j \) feeds information and/or provides feedback in its social network (see Table 3.2), user \( i \)’s relational capital, \( RC_i \), is estimated by:

\[
RC_i = \frac{1}{|F_i|} \sum_{j \in F_i} T_{ij} \cdot rc_j,
\]
where $RC_i$ is scaled as a real number in $[0, 3]$ as we consider three features in $rc_j$. Hence, $SC_i$ is ranged in $[0, 3]$ as a real number based on these three components of social capital described in Table 3.2.

$T_{ij}$ is user $i$’s trust in user $j$ [40]. A user’s trust is generally estimated by sharing or feeding information and providing feedback, such as liking or commenting, as described in Section 3.5.1. $T_{ij}$ is estimated by:

$$T_{ij} = \sum_{x \in X} t_x \cdot T^x_{ij},$$

(3.6)

where $X$ is the trust dimension indicating feeding and feedback behaviors in this work. $t_x$ is the weight for trust in $x$ where $\sum_{x \in X} t_x = 1$. $T^x_{ij}$ is calculated based on the number of positive interactions $I^x_{ij}$ between user $i$ and user $j$. The equation of $T^x_{ij}$ is:

$$T^x_{ij} = \frac{I^x_{ij}}{\max(I^x_{ik} \text{ for } k \in F_i)},$$

(3.7)

where $F_i$ is a set of user $i$’s friends. Positive interactions are defined as feeding and feedback activities between legitimate users $i$ and $j$. However, when user $i$ is an attacker sending a phishing message to the user $j$, if user $j$ cannot detect it, we treat it as a positive experience. However, if user $j$ detects the attack, it is treated as a negative experience for attacker $i$. For the details of what features are used to measure $hi$, $cc_i$, $rc_i$, $T_{ij}$, and a friend network ($F_i$), please refer to Table 3.2.

### 3.6.2 Phishing Attack Detection

When attacker $j$ sends out a fraud message to its friend, which is legitimate user $i$, legitimate user $i$ can detect attacker $j$ based on $P_{ij}^{crd}$, which is discussed in Eq. (3.1). Legitimate users in one’s OSN can help their friends detect phishing attacks. If the user $i$’s friend, $k$, can detect a phishing message by attacker $j$ using $P_{kj}^{crd}$, then user $i$ is immune to the attack or can be recovered from the attack with the help of user $k$. Then, user $k$’s can increase their $g_k$’s (i.e., the number of phishing attack experiences) like user $i$ and use the indirect experience to detect similar phishing attacks. This can naturally increase the probability that the user $i$’s friends can detect similar future attacks.

If legitimate user $i$ receives a phishing message from someone $j$ in his/her friend network but is unsure of whether the received message is from a phishing attacker based on his judgment ability in information credibility, $P_{ij}^{crd}$ (see Eq. (3.1)), user $i$ will share his/her information based on his/her posting frequency probability, $P_{i}^{post} = \text{post}_i$. If the user $i$ decides not to share it, he/she will not have a chance to get helped by his/her friends. When friend $k$ sees the posting of user $i$ on the phishing attack, he/she can help by leaving a comment based on $P_{k}^{com} = \text{com}_i$. If friend $k$ can correctly detect the phishing message posted
3.6. Proposed Approach: SAFER

Figure 3.1: A user’s status update in the SER-SEIR model.

by user $i$, the attack from attacker $j$ can be successfully detected with the probability of $P_{\text{post}}^i \cdot P_{\text{com}}^k \cdot P_{\text{crd}}^{kj}$. Otherwise, user $i$ fails to detect the phishing attack from attacker $j$. This implies that as long as user $i$ is active enough to reach out to other friends who have a sufficient level of willingness and competence to help, user $i$’s friends can be good assets to protect the user from phishing attacks, which aligned with the core concept of social capital in terms of accessibility to resources [212].

If more than three users detect an attacker and report it to the OSN provider, the attacker’s account will be suspended, and all its social connections will be removed from this network. If a legitimate user’s account is compromised by an attacker (with the probability $P_{\text{cp}}$), the account can be used to propagate phishing messages to its friends. If the compromised account is detected, the OSN provider will require the original owner to reset his/her password. After the password is reset, this user can be back in $R$, and the original user of the compromised account may lose some extent of social capital because some friends detect the user as an attacker and terminate their friend relationships with the compromised account.

3.6.3 Updating a User’s Status

The behaviors and activities of attacks and defenses are measured by the state update using our proposed SER-SEIR (Susceptible, Exposed, Recovered-Susceptible, Exposed, Infected, and Recovered) model, which is extended from [167], as depicted in Figure 3.1. The proposed SER-SEIR model allows users to move from $S$ to $E$ when the user receives phishing attacks. Then, the user can self-defend or defend against it with the help of friends, which allows the user to move from $E$ to $R$ directly without going through $I$. However, if the user cannot defend against the attack, its account is compromised by the attacker, moving from $E$ to $I$. If the user in $I$ changes the password of his/her account, it can move from $I$ to $R$. A user can only interact with his/her friends. Each state is described as:
A legitimate user, who has not experienced a given attack, is in $S$ (Susceptible).

A legitimate user in $E$ (Exposed) if the user receives a phishing attack and the user fails to detect an attack when: (i) The user received an attack but could not detect it without asking other friends for help; (ii) The user received an attack but could not detect it correctly even if the user asked other friends for help.

A legitimate user is in $I$ (Infected) if the user account that did not detect the attack is compromised and manipulated by the attacker. A legitimate user in $I$ can be infiltrated by an attacker as a compromised account with the probability $P_{cp}$.

A legitimate user is in $R$ (Recovered) if the user successfully detects an attack by: (i) The user detects the attacker correctly with his/her own detection ability, $P_{self-cr}$ (see Eq. (3.1)); (ii) The user detects the attack if his/her friend(s) can help and correctly detect the attack when the user shares the incident with his/her friends as he/she cannot send-defend, with the probability $(1 - P_{self-cr})P_{post} P_{com} P_{friend-cr}$; or (iii) The compromised account is reported by at least three friends and the OSN provider asks the legitimate user to change his/her password.

We summarize the overview of the proposed SAFER scheme in Figure 3.2.
3.7 Experimental Setup

3.7.1 Datasets

We use two real datasets from Twitter: Cresci15 [46] with bot attackers and 1KS-10KN [295, 296] with human attackers. Each dataset is detailed as follows:

- *Cresci15 [46]*: This dataset contains 1,481 normal Twitter accounts from a sociological study, 469 certified human accounts, and 845 fake bot accounts bought from online markets. After combining the three datasets and filtering out the accounts that posted no tweets, the network size is 2,664. There are 1,946 legitimate users with 398 friends and 718 attackers with 554 friends on average.

- *1KS-10KN [295, 296]*: This dataset is comprised of 10,000 legitimate users, and 1,000 human attacker accounts crawled from Twitter. Spammers are accounts that are identified as posting malicious URLs. For meaningful result analysis, we filtered out the accounts that posted zero tweets and obtained the network size with 10,766 users, comprising 9,766 legitimate users with 7,744 friends and 1,000 spammers with 2,520 friends on average.

3.7.2 Ethics

We use publicly available datasets collected from Twitter API in the existing research [46, 295, 296] to evaluate our proposed approach. The datasets are all anonymized by hiding identity information.

3.7.3 Default Parameterization

The default attacker ratio is set to 20% with $P_a = 0.2$. We considered 1,946 legitimate users and 389 attackers to consider the same network size for both datasets. When the attackers perform attacks, the fraction of targeted victims from legitimate users is set to $P_{AS} = 0.1$ by default. The experimental results are based on the average data points from 100 simulation runs ($N_{sim}$). We used the ratios of susceptible, infected, and recovered users over the total number of legitimate users (see Sections 3.6.3) as the key metrics to evaluate the friending decision schemes. The default values of key parameters used in our experiment are summarized in Table 3.3.
3.7.4 Experiment Procedures

We take the following steps:

- Each user is assigned a set of features based on the normalized nine features.

- An attacker will perform attacks where its deception quality is randomly selected as a real number ranging in \([0, 3]\) based on Gaussian distribution with 1.5 for mean and 0.3 for a standard deviation (i.e., \(N(1.5, 0.3)\)).

- Given a set of behavioral seed probabilities for each user, which include behaviors (i.e., feeding information, providing feedback, inviting friends, and judging credible information) modeled in Section 3.5.1, each user starts making friends and interacts with other users. We allow 100 times of interaction steps (i.e., chances to interact with other users, such as whether to invite a new friend or post/share information) where each user will exhibit his/her behavior based on the derived behavioral seed probabilities.

- Each user starts with friends selected based on individual features of its social capital. Recall that a user’s social capital is estimated based on his/her friends’ social capital. That is, when all users are not connected, an individual user’s social network starts with himself/herself alone. However, since the features representing a different type of social capital are different, we can allow the estimation of a user’s social capital without any friends based on \(cc_i, h_j, \) and \(rc_j\) in Eqs. (3.3), (3.4), and (3.5).

- For other non-social capital-based FRSs, we will initialize a user’s features and select friends based on the given features. For example, for Trust-based FRS (TR), each user is initialized based on the average sum of posting and commenting behaviors (i.e., \(T_i = w_{post} \cdot P_{post}^i + w_{com} \cdot P_{com}^i\) where \(w_{post} + w_{com} = 1\)). We weigh each trust component equally in our case study. We let a user select his/her first friend based on social attributes and topics of interest, respectively, for Social Attributes-based FRS (SA) and Topic model-based FRS (TM).

- Upon every interaction chance, calculate each user \(i\)’s \(h_i, rc_i, cc_i, \) and \(T_i\), respectively.
3.7. Experimental Setup

- Make friending decisions based on FRSs (see ‘Comparing Schemes’) where each scheme needs the current interaction states in estimating the criteria features, such as social capital, trust, topic features, or social attributes.

- From the 101-st to the 105-th interaction step (i.e., five interaction steps), perform a one-time attack per step where each potential victim can respond based on his/her behavioral characteristics associated with social capital, as described in Section 3.6.2. Update each node’s status based on the proposed SER-SEIR model in Figure 3.1. A one-time phishing attack is explained in Section 3.5.2. We allow each attacker to apply its corresponding attack to a normal user friend currently in the state of S or E.

- Repeat the attacks from each attacker for the 101-st to the 105-th interaction step.

- Calculate $S$, $E$, $I$, and $R$, as described in the ‘Metrics’ section below with seven FRSs under phishing attacks performed by human or bot attackers based on the two datasets [46, 295, 296].

3.7.5 Comparing Schemes

We will compare the following FRSs that determine how each user is connected to other users, where each user is only given the number of friends (i.e., friend_num) based on the two datasets used. Note that we allow each user to select five friends at the very beginning of the network deployment and use the corresponding FRS to create a user’s social network.

- **RC-based FRS (RC):** A user selects new friends based on the top friend_num number of users using $RC_i$ in Eq. (3.5).

- **STC-based FRS (STC):** A user selects new friends based on the top friend_num of users using $STC_i$ in Eq. (3.4).

- **CC-based FRS (CC):** A user selects new friends based on the top friend_num number of users using $CC_i$ in Eq. (3.3).

- **Multidimensional SC-based FRS (MSC):** A user selects new friends based on the top friend_num number of users using $SC_i$ in Eq. (3.2).

- **Social Attributes-based FRS (SA):** A user selects new friends based on similar shared attributes [88]. The shared attributes are defined by a vector of $h_i$, $cc_i$, and $stc_i$. The top friend_num number of users with the highest cosine similarity score of social attributes will be selected as new friends.

- **Topic model-based FRS (TM):** A user selects new friends based on the top friend_num number of users with the highest topic similarity score [283]. Under Cresci15 and 1KS-10KN, each user has a document of all his posts. The LDA algorithm processes
those documents to generate the top 20 topics and the probability for each topic. The similarity between the two users is a cosine similarity of 20 topics probability scores.

- **Trust-based FRS (TR):** A user selects new friends based on the highest trust [40]. A user’s trust is estimated based on how much other friends trust a given user by \( T_i = \frac{1}{|F_i|} \sum_{j \in F_i} T_{ji} \). A user selects new friends based on the top friend_num number of users with the highest trust.

Upon receiving a friend invite, a user will accept it as long as its key qualification in each FRS is no less than his/her own. For example, if the user uses MSC, it will accept the invite if the inviter has no less than MSC the user has.

### 3.7.6 Metrics

We used the following metrics to evaluate the effectiveness of friending decision schemes in terms of their resilience against phishing attacks:

- **The ratio of susceptible users** \((S)\) measures the ratio of susceptible users over the total number of legitimate users. A susceptible user has not received any attack before and has a chance to be infected or directly recovered if the user has a self-defense capability to detect the phishing attack.

- **The ratio of exposed users** \((E)\) measures the exposed users over the total legitimate users. An exposed user receives a phishing attack but fails to detect an attack.

- **The ratio of infected user** \((I)\) measures the ratio of infected users over the total number of legitimate users. A legitimate user received a phishing attack but could not detect it by itself or with the help of other friends.

- **The ratio of recovered users** \((R)\) measures the ratio of recovered users over the total number of legitimate users. A legitimate user received a phishing attack but self-detected the attack or detected the attack with the help of his/her friends. A legitimate user who did not receive a phishing attack can also be recovered if his/her friend self-detected the attack and posted a security alert.
3.8. Simulation Results & Analysis

3.8.1 Probability Distributions of Relational, Cognitive, and Human Capital

In this section, we discuss how social capital in three dimensions (i.e., relational, cognitive, and structural social capital) can be differently identified for attackers and legitimate users. Figure 3.3 shows the histograms of social capital dimensions, including relational capital, cognitive capital, and human capital, for attackers and legitimate users, where the distributions are identified based on the best-fit probability density functions. Each social capital corresponds to $rc_i$, $cc_i$, or $h_i$. Here we note that because structural social capital is not known (it is to be derived after a user determines his/her social network structure based on his/her friend network features), we capture it by an individual user’s human capital, $h_i$, which is the key input to estimate the structural capital as in Eq. (3.4).

In Figure 3.3, the fitting process applies 86 probability density functions using `scipy.stats` packages and picks the one that gives the minimum errors. From the two datasets (i.e.,
CHAPTER 3. SAFER: SOCIAL CAPITAL-BASED FRIEND RECOMMENDATION TO DEFEND AGAINST PHISHING ATTACKS

3.8.2 Ratio of Reported Attackers

Figure 3.4 shows the ratio of attackers who are detected by at least three users and the OSN provider suspends their accounts and cuts the friend connections. The reported attackers cannot deliver phishing attacks at later attack times. The four SC-based FRSs have more reported attackers than the non-SC-based schemes SA and TM for both Cresci15 and 1KS-10KN. However, Figure 3.4a and 3.4b indicate that the growth rate of reporting attackers with more attack times in Cresci15 is much higher than in 1KS-10KN for four SC-based FRS. Reported attackers of TP are lower than the four SC-based FRS in Cresci15, but those of TP have the same scale as SC-based FRS in 1KS10KN. In Figure 3.4a, SC-based FRSs can...
report 50% of attackers during the first attack, and they can almost report all the attackers from three attack times. The three baseline approaches report around 10-20% of attackers in the first attack, growing gradually during the five attack times. In Figure 3.4b, all seven schemes report less than 30% of attackers from the first attack and then grow slowly with increasing attack times. These findings suggest that SC-based FRSs effectively detect bot-related attackers, even at the beginning of phishing attacks. Under both networks, SC-based FRSs effectively defend against phishing attacks better than the other three baseline FRSs.

Figures 3.4c and 3.4d show the effect of varying the fraction of attackers when the number of attacks is set to 1 in terms of the ratio of reported attackers. We observe that the detection ratio decreases particularly under non-SC-based FRSs due to less power to deal with attacks upon the increase of the attackers. This implies that SC-related FRSs outperform non-SC-based counterparts in defending against phishing attacks. In Figure 3.4d, under human attackers, the trend of baseline methods, SA and TM, is similar to that under the Cresci15 dataset. However, the four SC-related FRSs have lower report ratios than in Cresci5 and trust-based FRS performances similar to the four SC-related FRSs. This pattern reveals that the defense capability against phishing attackers by SC-related and trust-based FRSs is insensitive to more numbers of attackers in the social network, in contrast to the trends observed under SA and TM. Among the SC-related FRSs, CC and RC show the best performance in defending against human attackers.

### 3.8.3 Comparative Performance Analysis Under Varying the Frequency of Attacks

Figure 3.5 demonstrates the performance of the seven FRSs regarding the effect on $S$, $E$, $I$, and $R$ when the attack strength varies from a one-time attack to five-time attacks by each attacker. We set the percentage of attackers ($P_a$) to 20% among all users in the network. By increasing the attack frequency, the two networks under Cresci15 and 1KS-10KN have more recovered users ($R$) and less susceptible users ($S$). However, during the five-times phishing attacks, $E$ and $I$ reached a maximum point for some FRSs and then started decreasing, as shown for all four SC-based FRSs and TR in Figures 3.5c, 3.5f, and 3.5g.

In Figures 3.5a and 3.5d, under Cresci15 with bot attackers, we can clearly observe lower $S$ and higher $R$ in SC-based approaches (i.e., RC, CC, STC, and MSC) than three baseline methods (i.e., TR, SA and TM). Under 1KS-10KN with human attackers in Figures 3.5e and 3.5h, the performance gaps between SC-based and non-SC-based schemes are less clear than under Cresci15. However, TR has comparable performance with SC-based FRSs for $R$ and SC-based schemes still outperform non-SC-based counterparts, SA and TM.

Under 1KS-10KN with human attackers, unlike what we observed in Cresci15, which has bot attackers, human attackers’ social capital degrees are less distinct than legitimate users. Due to this, it seems more human attackers can penetrate the friend networks of legitimate
users (being friends of more legitimate users). In addition, the network in 1KS-10KN, with a mean degree of about 73, is much denser than the network in Cresci15, with a mean degree of about 36. Higher network density is more likely to make each user’s social capital less distinct due to more users with many friends, leading to less distinctive characteristics of users’ friend networks. Therefore, compared to the results under Cresci15, the results under 1KS-10KN are less sensitive to varying the attack strength. However, we can still observe that SC-based and TR-based (highly similar to RC-based in dense networks) FRSs perform better than SA and TM due to high attack detection by friend users of a user with high social capital, as shown in Figures 3.4c and 3.4d.

3.8.4 Comparative Performance Analysis Under Varying the Percentage of Attackers

Figure 3.6 shows the performance of the seven FRSs under varying the percentage of attackers ($P_a$) on $S$, $E$, $I$, and $R$, where attackers perform one-time phishing attacks. As overall trends, more attackers in a network introduce more exposed users $E$ and accordingly increases more recovered users $R$, resulting in reduced $S$.

Under Cresci15 with bot attackers, the performance of SC-based FRSs clearly outperforms non-SC-based counterparts, TR, SA, and TM, showing the highest $R$. Interestingly, all SC-based schemes perform very similarly without many differences. During the first attack time, if more attackers exist in the network from 5% to 30%, the attacked users increase from 6% to 15%, while the recovered users increase more from 9% to 30% in SC-related FRSs. However, the outperformance of SC-based FRSs over non-SC-based counterparts (i.e., TR, SA, and TM) is still clear. Both are demonstrated in Figures 3.6b and 3.6d. Under 1KS-10KN, the gap between SC-related FRSs and non-SC-related FRSs is closer in $E$, $I$, and $R$ from Figures 3.6f, 3.6g, and 3.6h, especially $R$ in TR is close to SC-based FRSs.

Under both Cresci15 and 1KS-10KN, more users from $S$ move to $R$, $E$, and $I$ in the network built by SC-based FRSs than by three baseline FRSs. The increased $R$ implies that more phishing targets can verify the phishing attacks. This is the benefit of leveraging friend resources: The target can verify the attacker by his/her own judgment, and the target has a chance to help the friends to increase experience $g$, or the target cannot verify the targets, but his/her friends have the chance to detect the phishing attacks and reply to the target.

3.9 Key Findings

We proposed the SAFER (Social cApital-based FriEnd Recommendation) scheme to combat phishing attacks in OSNs. We quantitatively measured social capital by three dimensions in terms of structural, cognitive, and relational capital based on nine behavioral features.
3.9. Key Findings

collected from the two real datasets, Cresci15 [46] and 1KS-10KN[295, 296]. We analyzed and demonstrated how social capital in three dimensions is different between legitimate users and attackers.

We developed a set of FRSs based on four different types of social capital (i.e., relational, cognitive, structural, and multidimensional social capital) and compared their performance with three different non-social capital friending decision schemes, which are trust-based [40], social attributes-based [88] and topic-based [283]. We investigated the performance of these seven FRSs in terms of the extent of resistance against phishing attacks using two real datasets with phishing attacks by bots and humans, respectively. The users were modeled as legitimate users, while spammers were modeled as phishing attackers. Based on the proposed SER-SEIR model, a user’s resistance against phishing attacks was evaluated in three states, susceptible, infected, and recovered. We examined the performance of the seven FRSs under varying attack severity and portion in a given OSN.

We identified the following key findings from our study:

- Users with high social capital (SC) friends can self-defend against phishing attacks better than those with friends with the same topic interests or social attributes.
- SC-based FRSs can enable users to combat phishing attacks better than non-SC-based counterparts because the friends of the users with high social capital can help them defend against phishing attacks.
- Bot-based phishing attacks can be more easily detected and defended than human-based ones under all FRSs because bot attackers showed more distinctive characteristics than human attackers in social capital.
- Although SC-based FRSs can allow more attackers to infect (engage) users in the attacks due to users with high social capital attracting more attacks, even the infected users can be easily recovered with the help of their friends in their social networks.
- All SC-based FRSs performed comparably in detecting phishing attacks, while the cognitive SC has shown the best performance among all FRSs with a slightly better performance. This suggested that if a weighted linear model was to be used, cognitive SC could be assigned a higher weight than relational SC and structural SC.

In future work, we plan to: (1) collect real datasets with more user behavioral features and personality traits based on text information generated by users in OSNs; (2) conduct case studies in multiple OSN platforms for applicability; (3) study the influences of personality traits and demographic features on social capital quantification and the behaviors of legitimate users and human attackers types; (4) explore bot detections based on our SC quantification and findings; and (5) address the effectiveness of using social capital in terms of recommending ‘useful friends’, emphasizing resourcefulness, which is well-aligned with the core concept of social capital.
Figure 3.5: $S$, $I$, and $R$ comparisons of seven FRSs varying one to five phishing attacks by 20\% attackers from the Cresci15 [46] and the 1KS-10KN [295, 296], respectively.
3.9. Key Findings

Figure 3.6: The comparisons of seven FRSs varying the fraction of attackers ($P_a$) from Cresci15 [46] and 1KS-10KN [295, 296], respectively.
Chapter 4

Mitigating Influence of Disinformation Propagation Using Uncertainty-Based Opinion Interactions

This chapter provides a game-theoretic opinion model to simulate the disinformation propagation dynamics in an OSN. It shows opinion polarization and network communities by five opinion models based on our published papers [90, 92, 93]. The work in this chapter partly addresses the Combating Task.

4.1 Motivation & Research Goal

Social media platforms facilitate everyone to express their opinions and comments to the whole online world. Many people, especially young people, rely heavily on social media to acquire daily news. They learn about new events from diverse sources. Accordingly, disinformation propagated by malicious users can significantly mislead users, altering their opinions. However, regardless of whether the information is true or not, since information diffuses fast in online social networks (OSNs), it is implausible for OSN users to verify all the information they encounter. Disinformation causes severe privacy violations, ruins reputations, or produces financial losses. Recently, these serious issues caused by false information, fake news, or rumors have demonstrated detrimental effects on society, such as influencing decision-making processes in elections, pandemics, health, or education.

This work investigates how rational users update their opinions in the presence of disinformation propagated in an OSN. We define rational users as those who seek to maximize their utilities by updating their opinions based on their preferences. In particular, we consider the rational users who have the ability to reason and filter disinformation in terms of the credibility of information based on a source’s expertise, the veracity of information, their current opinions, and the opinions of their friends. Therefore, this study will examine how these users’ rational behaviors in updating their opinions can affect the mitigation or amplification of disinformation propagation, where disinformation propagation can also introduce network and/or opinion polarization.
OSN users’ behaviors towards online information are reflections of their innate propensities or personalities. Some people make decisions based on the competence (or expertise) or certainty of a source. Other people accept other people’s opinions based on like-mindedness (i.e., homophily), agreeableness (i.e., conformity), or stubbornness (i.e., not accepting different opinions) \[39, 41, 190\]. This can be explained based on confirmation bias \[67, 85\], a cognitive bias representing a tendency a person shows to understand, interpret, prefer, or remember information that confirms one’s previous values or beliefs. Another well-known cognitive bias, which explains a subjective belief that can mislead a decision-maker, is the Dunning-Kruger effect \[62, 147\]. This effect also well explains how a subjective belief affected by irrational cognitive bias can lead to a poor decision. The false information diffusion among users in an OSN has been modeled with specific user behaviors by \[313, 314, 317\]. The theoretical models supported by game-theoretic approaches consider user behaviors and human rationality by cognitive limitations \[14, 162, 293, 298, 310\]. However, to the best of our knowledge, the existing game theory models rarely help users make rational decisions in updating their opinions.

However, mitigating disinformation propagation in the existing game models by network users as decision-makers meet several main challenges: (1) **Users’ decisions to update uncertain opinions.** More real user behaviors, such as updating opinions from social interactions, can complement spreading decisions. However, little work has leveraged game theory to justify the significant tendency of users’ information processing. (2) **Defense from both individual and network levels.** Individual users’ types of subjective opinion updates can mitigate the effect of disinformation in the OSN through the dynamic opinion-based epidemic model. (3) **Network polarization.** It is critical to address the divergent influences of users’ opinion updates on network communities, as disinformation is often related to the polarization of users \[52\].

In this work, we propose a three-player game framework that models online users’ behaviors in updating their opinions based on interactions with other users or attackers. In the framework, we also model attackers’ deception tactics to propagate disinformation and the defender (OSN platform system administrator)’s policy to ensure a safe and trustworthy OSN environment. We aim to demonstrate how OSN users’ rational information processing behaviors based on opinion update models using various criteria can influence the mitigation of disinformation propagation, which impacts network dynamics and opinion polarization. Via this experiment, we aim to suggest what OSN user behaviors and the OSN platforms can ensure safe and trustworthy cyberspace from disinformation propagation.

### 4.2 Research Questions

In this work, we aim to answer the following research questions:

1. How can game theory help users formulate and update their opinions to maximize
their utilities that properly reflect their preferences?

2. What type of opinion model can help users combat false information propagation?

3. How does a user update his/her opinions via interactions with other users, and how does it affect the dynamics of the network in distributing power or influence in a network?

We will answer the above research questions in Section 4.9.

4.3 **Key Contributions**

This work has the following **key contributions**:

1. We design and test a generic game-theoretic opinion framework in an OSN environment against the spread of disinformation. We demonstrate the flexibility of this framework to accommodate various user interactions and opinion models (OMs), including uncertainty, homophily, encounter, herding, and assertion-based updates.

2. We clarify the defense against disinformation propagation based on the three network views in Figure 4.1, including a number of decision-makers’ social interactions and opinion-updating decisions, each decision-maker’s opinion scale in the opinion polarization network, and the overall epidemic status in the SIR network. This enables users’ interactions and opinion models to affect disinformation propagation, which also influences the segregation of two parties in believing true or false information and opinion polarization in a given OSN.

3. Utilizing the rich features in two real-world datasets collected from the Twitter platform, we capture and model real OSN users’ social activities and propensities, such as information sharing or exchanging.

4. We leverage a belief model called *Subjective Logic* (SL), which represents users’ subjective, uncertain opinions in real-world, uncertain situations to decide whether to accept other users’ opinions. Further, we expand the SL-based opinion model to consider users’ five opinion models and realize attackers’ deception strategies [145].

5. We construct a set of uncertainty-aware payoff equations based on the SL opinion in a game of three agents, i.e., attackers, a defender, and users. Each player can make choices based on uncertainty, observations, and inherent preference to maximize utility when interacting with or accepting other users’ opinions. Furthermore, we solve each player’s preferred strategies by Nash Equilibria (NE), making decisions based on correct beliefs towards the opponents’ moves. Since perfect NE strategies may not be
4.3. Key Contributions

Figure 4.1: The overview of the proposed opinion update game in an OSN. The top figure illustrates the three subgame examples in one interaction, compared to their related NEs. The bottom-left figure represents the opinion polarization in the last interaction. The bottom-right figure depicts the network status of the bottom-left figure by the epidemic model.

realistic in complex real-world scenarios, we compare the performance gap between each player’s best strategies chosen under uncertainty and its NE choices.

6. We optimize epidemic model parameters in an effective gradient descent algorithm. Since the opinion dynamics of all the network agents can reflect the transition of states in the Susceptible-Infected-Recovered (SIR) model, we optimize their infection (i.e., believing in disinformation) and recovery (i.e., disbelieving in disinformation) rates.

7. We also examine network community structure and opinion polarization after OSN users interact with each other upon disinformation propagation, where network community structure is investigated by three graph partitioning algorithms [43, 45, 139] by calculating network polarization scores using four different methods [43, 74, 76, 87]. From the polarization of opinions under the five opinion models, we investigate how disinformation can influence network topology changes and network power (i.e., social
8. We investigate how different ways of updating opinions can introduce different opinion dynamics and change the social capital of OSN users. We measure a user’s bridging social capital by betweenness and bonding social capital by a trust because both metrics can represent a user’s influence or power in a network [274].

4.4 Background & Related Work

In this section, we provide a brief overview of state-of-the-art research about opinion models, game theoretic information diffusion, and the effect of disinformation on polarization.

4.4.1 Opinion Models

One of the pivot questions on opinion models is how to accept new evidence to update the current opinions from pairwise interactions. Opinion models show how OSN users update their opinions when encountering another OSN user with a different opinion. An assertion model provided two criteria for users to update an opinion [323]: the amount of knowledge and the degree of belief. The users can determine the level of knowledge exchange regarding their forgetfulness, learning capability, and trust in another user. A herding-based opinion model was also developed by considering the amount of pairwise social interactions with all other friends [241]. Uncertainty-based opinion model was also discussed, where a user could update an opinion when an encountered user’s opinion has higher certainty, such as high expertise [39]. Similarly, an uncertainty range interval-based opinion model was used for a user to update an opinion [307]. This model calculated the distance between two opinions based on the range of the uncertainty interval length if an agent has uncertain opinions. However, the works above [39, 241, 307, 323] did not consider a user’s rational behaviors in updating opinions. Unlike those works, we leverage game theory in diverse opinion models to investigate the impact of the user’s rational behavior on the spread of disinformation.

4.4.2 Game-Theoretic Information Diffusion

Yang [298] modeled two user strategies, ‘cooperative’ or ‘defective’ in the prisoner’s dilemma game and the public goods game for binary opinion diffusion when the equilibrium of opinion consensus was reached. Several evolutionary game theory (EGT) works of opinions models solved a stable evolutionary state to model the user’s strategy transition rate [14, 162]. The goal of the EGT model [14, 115, 162, 247, 304] is to consider several factors and population preferences to influence user decisions. However, EGTs mainly deal with only three behaviors, spreading rumors, not spreading rumors, or spreading anti-rumors, without considering the opinions of individual users. Szabó and Tőke [247] studied the likelihood of strategy
imitation in the Fermi updating rule, determined by the actual advantages of the fitness of the neighbor. Li et al. [162] simulated rumor diffusion in an OSN with various personal and social attributes, such as users’ tie relationship with friends, judgment ability of others, strategy imitation, and the cost of spreading the rumor. Askarizadeh et al. [14] discussed a user’s behavior of spreading rumors by attitude or awareness, community anxiety, and the intensity of rumor and anti-rumor cascades in OSN. Huang et al. [115] developed users’ cost-effective defense strategy against rumors where the users’ opinions toward the rumors were updated by a differential game model. Yoshikawa et al. [304] studied a mode where users update their friends’ reliability and doubt (or distrust) and then exchange opinions by Bayesian estimation.

Recently, psychological factors have been commonly considered to model the defense strategies in rumor-spreading game models [293, 310]. Xiao et al. [293] modeled the spreading behaviors from their psychological factors and investigated the competition by messages supporting rumors or clarifying rumors. Zhang et al. [310] studied the resistance of malicious users based on the reputation dynamics of a user’s neighbors. The authors have not explored any game-theoretic opinion framework aiming to mitigate disinformation influence. Further, they did not investigate the OSN network and opinion dynamics when users interact for opinion exchanges and updates.

Unlike the above works [14, 115, 162, 247, 293, 298, 304, 310], our work uses game theory to model users’ opinion propagation behaviors and strategies to deal with disinformation. In addition, we investigate how the game-theoretic opinion models and rational user behaviors affect opinion dynamics and the patterns of disinformation spreading.

### 4.4.3 Impact of False Information on Network Polarization

Research has shown that false information spreading can polarize users [52], which can facilitate false information circulation [19]. Polarized users tend to gain access to similar content and may have a long response lag for their fake news posts [268]. A network of polarized users can be divided into several communities due to the echo chamber effect that users assigned to the same group are highly interconnected [6, 220]. Those previous studies modeled the spreading patterns of false information but not how the opinions can be updated. Hence, polarized users and communities were examined only based on users’ binary behaviors, such as whether to spread false information. A social network with a polarized topology can cause a non-trivial reduction of social capital access (e.g., cognitive and relational capital) [220]. There is a strong correlation between users’ opinion homophily and activities towards false information [19], so clusters can be predicted by the fact that like-minded users with similar polarization scores can gather quickly. In the meanwhile, homophile clusters can expedite the speed of false information spreading [19]. In addition, the authors have not considered any game-theoretic decision-making process. In particular, the studies in [19, 220] investigated the influence of polarized users on the rate of false
information spreading in a sense inverse to our research.

Several social science studies [15, 101, 258] have drawn a conclusion. However, their findings have not been reflected in the simulation models. Homophily increases with a larger community size [101]. In most communities, intra-community information diffuses more quickly and broadly than inter-community information because more people tend to be exposed to information. We usually observe this phenomenon in political campaigns where disinformation can increase conflicts and break strong social ties and social capital between inter-communities [15]. In addition, the long-term effects of political disinformation on social capital have not been well studied [258]. Thus, this research tried to show this effect on social capital in our model.

Unlike the above works [6, 15, 19, 52, 101, 220, 258, 268], our work pioneers in investigating the effects of disinformation propagation on network and opinion polarization and the distribution of network power using social capital where OSN users interact with other uses based on different types of exchanging their opinions.

### 4.5 Uncertain, Subjective Opinion Model

This work leverages a belief model, Subjective Logic [41, 131], in an OSN to quantify users’ binomial opinions. Through pairwise interactions, an initial uncertain opinion can be updated by a user. Table 4.1 succinctly summarizes all the notations used and their meanings.

#### 4.5.1 Opinion Formation

A binomial opinion, \( \omega = (b, d, u, a) \), is represented by belief \( b \), disbelief \( d \), uncertainty \( u \), and base rate \( a \) in SL [131]. Those four dimensions are formed as follows:

\[
\begin{align*}
    b, d, u, a & \in [0, 1], & b + d + u & = 1, \\
\end{align*}
\]

(4.1)

where belief \( b \) means the degree of pro, agree, or true information held by an agent to believe a proposition even if the real truth is unavailable. Disbelief \( d \) means the degree of con, disagree, or false information for an agent to oppose or disbelieve a proposition. Uncertainty \( u \) means the level of vacuity normally due to an insufficient amount of evidence. Base rate \( a \) is the prior belief, expertise, or bias [131] for an agent’s prior knowledge in a given domain. An agent updates its opinion in the four dimensions, each updated by considering an interacted agent’s opinion. Based on the obtained evidence, a user’s binomial opinion can be represented by the observed or available evidence by the following mapping rule:

\[
\begin{align*}
    b &= \frac{r}{r + s + W}, & d &= \frac{s}{r + s + W}, & u &= \frac{W}{r + s + W}, \\
\end{align*}
\]

(4.2)
4.5. Uncertain, Subjective Opinion Model

Table 4.1: Notations of Design Parameters and Their Meanings

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\omega, \omega_i)</td>
<td>User i’s SL-based binary opinion</td>
</tr>
<tr>
<td>((b, d, u, a))</td>
<td>Belief, disbelief, uncertainty, and base rate</td>
</tr>
<tr>
<td>(P(b_i))</td>
<td>User i’s projected belief from (\omega_i)</td>
</tr>
<tr>
<td>(P_{\text{true}}, P_{\text{false}}, P_{\text{uc}})</td>
<td>Proportion of true informers, false informers, and normal users</td>
</tr>
<tr>
<td>(\oplus)</td>
<td>Consensus operator from SL</td>
</tr>
<tr>
<td>(\otimes)</td>
<td>Trust operator from SL</td>
</tr>
<tr>
<td>(c^i_j, uc^i_j, hc^i_j)</td>
<td>Discounting factors from SL, uncertainty-based and homophily-based</td>
</tr>
<tr>
<td>(\omega_i)</td>
<td>Uncertainty maximized opinion of user i</td>
</tr>
<tr>
<td>(\xi)</td>
<td>Threshold of uncertainty maximization</td>
</tr>
<tr>
<td>(P_{f}^i, P_{p}^i)</td>
<td>User i’s feeding probability and posting probability</td>
</tr>
<tr>
<td>(PD_{ij})</td>
<td>Projected discrepancy between two opinions</td>
</tr>
<tr>
<td>(\phi^k_i)</td>
<td>Threshold to accept or request a friend</td>
</tr>
<tr>
<td>(a^k_i)</td>
<td>Attacker’s strategy k in {DG, C, DN, S}</td>
</tr>
<tr>
<td>(a^\ell_i)</td>
<td>Normal user’s strategy (\ell) in {SU, U, NU}</td>
</tr>
<tr>
<td>(a^m_i)</td>
<td>Defender’s strategy m in {T, M}</td>
</tr>
<tr>
<td>(\omega_{FP})</td>
<td>False opinion ((0, 1, 0, 0))</td>
</tr>
<tr>
<td>(\omega_{TP})</td>
<td>True opinion ((1, 0, 0, 1))</td>
</tr>
<tr>
<td>(EP^A_{kj})</td>
<td>Expected payoff of attacker’s strategy k</td>
</tr>
<tr>
<td>(u_{kj})</td>
<td>Utility of an element in (EP^A_{kj})</td>
</tr>
<tr>
<td>(EP^D_{\ell k})</td>
<td>Expected payoff of defender’s strategy (\ell)</td>
</tr>
<tr>
<td>(u_{\ell k})</td>
<td>Utility of an element in (EP^D_{\ell k})</td>
</tr>
<tr>
<td>(c_{\ell k})</td>
<td>Defender’s cost of strategy (\ell)</td>
</tr>
<tr>
<td>U, H, A, HE, E</td>
<td>Uncertainty, Homophily, Assertion, Herding, and Encounter-based user types</td>
</tr>
<tr>
<td>(EP^{U}_{m})</td>
<td>Expected payoff of user’s strategy m</td>
</tr>
<tr>
<td>(a^m_i)</td>
<td>Probability of user j as an attacker</td>
</tr>
<tr>
<td>(stc_i)</td>
<td>User i’s structural social capital</td>
</tr>
<tr>
<td>(T_i)</td>
<td>User i’s trust by other friends</td>
</tr>
<tr>
<td>(N_R)</td>
<td>Number of reports to alert a defender</td>
</tr>
<tr>
<td>(\rho)</td>
<td>Tolerance to report a malicious user</td>
</tr>
<tr>
<td>I</td>
<td>Number of interactions in simulation</td>
</tr>
<tr>
<td>N</td>
<td>Number of nodes in the OSN</td>
</tr>
</tbody>
</table>

where \(r\) is the amount of positive evidence and \(s\) is the amount of negative evidence for a certain proposition. \(W\) is the amount of uncertain evidence as inherent errors from a system or an environment, such as the unavailable source of evidence or the limited and partial ability of observation. In a binomial opinion, \(W\) is commonly set to 2, representing the number of beliefs (i.e., two belief masses, including belief and disbelief) [131].

For user i, the binomial opinion is \(\omega_i = (b_i, d_i, u_i, a_i)\). The expected belief \(P(b_i)\) and expected disbelief \(P(d_i)\) are the factors for user i to make a decision by:

\[
P(b_i) = b_i + a_i u_i, \quad P(d_i) = d_i + (1 - a_i) u_i,
\]

(4.3)

where \(P(b_i) + P(d_i) = 1\) as \(b_i + d_i + u_i = 1\). If user i needs a decision but \(b_i\) or \(d_i\) is very similar, \(a_i\) can be a critical decision factor because \(a_i\) is used to interpret uncertainty \(u_i\).

4.5.2 Initialization of Opinions

*True and false informers* can be zealots [267] to support extreme opinions in our network.
The zealots not only propagate their true or false opinions but also refuse to change their opinions. All other users with initial uncertain opinions are willing to learn new opinions via sharing or feeding behaviors toward other users. We set the fractions of users with
\[ P_{\text{true}} + P_{\text{false}} + P_{\text{uc}} = 1 \]
where \( P_{\text{true}} \), \( P_{\text{false}} \), and \( P_{\text{uc}} \) refer to the fractions of true informers, false informers, and normal users, respectively. When the false informers (i.e., attackers) take the action Subversion \( (S) \), they propagate the false opinion (see Section 4.6). We initialize a true opinion, false opinion (i.e., disinformation), and uncertain opinion, held by false informers (i.e., attackers), true informers, and legitimate users, respectively, by:

- **True opinion**, \( \omega_T \), is initialized with \( (b \to 1, d \to 0, u \to 0, a = 1) \), implying true opinion’s belief (i.e., believing true information) is close to 1 (highly true) while disbelief (i.e., disbelieving true information) is close to 0.

- **False opinion (i.e., disinformation)**, \( \omega_F \), is initialized with \( (b \to 0, d \to 1, u \to 0, a = 0) \). This means that belief in false opinion is close to 0 while disbelief is close to 1.

- **Uncertain opinion** is formulated to represent the opinion of the rest of the users, except true and false informers, initialized as \( (b, d, u, a) = (b \to 0, d \to 0, u \to 1, a = 0) \), without showing strong preference. Their opinions can be updated depending on an interacted user’s opinion and the way to update his/her opinion (i.e., opinion models).

### 4.5.3 Opinion Update

False and true informers are zealots and do not change their opinions while influencing other users’ opinions. Legitimate users who lack confidence will refresh their opinions with new information from pairwise interactions when interacting with their friends. Since uncertainty-based OM, homophily-based OM, and encounter-based OM are all grounded by SL’s consensus update mechanism, we describe how the SL framework updates user \( i \)’s opinion \( \omega_i \) as below.

In SL, the first step to update agent \( i \)’s opinion \( \omega_i \) is to consider how much agent \( j \)’s opinion can be accepted by agent \( i \). Agent \( i \) discounts agent \( j \)’s opinion by discounting operator \( c_i^j \), which implies agent \( i \)’s trust in agent \( j \). Hence, the agent \( j \)’s opinion is considered by agent \( i \) based on \( \omega_{i \otimes j} = (b_{i \otimes j}, d_{i \otimes j}, u_{i \otimes j}, a_{i \otimes j}) \). Each opinion element is given by:

\[
\begin{align*}
    b_{i \otimes j} &= c_i^j b_j, \\
    d_{i \otimes j} &= c_i^j d_j, \\
    u_{i \otimes j} &= 1 - c_i^j (1 - u_j), \\
    a_{i \otimes j} &= a_j,
\end{align*}
\]  

(4.4)

where the quantity of \( u_{i \otimes j} \) above is the same as \( u_{i \otimes j} = 1 - b_{i \otimes j} - d_{i \otimes j} \) since \( b_i + d_i + u_i = 1 \). These trust opinion calculations from Eq. (4.4) are all at time \( t \).

The second step to update SL opinion \( \omega_i \) is integrating the discounted opinion \( \omega_{i \otimes j} \) by the consensus operator [131]. Using a consensus operator, the agent \( i \)’s new opinion at time \( t + 1 \)
Each element is given by:

\[ \omega_i \oplus \omega_{i \oplus j} = (b_i \oplus b_{i \oplus j}, d_i \oplus d_{i \oplus j}, u_i \oplus u_{i \oplus j}, a_i \oplus a_{i \oplus j}). \]

Based on the above SL operations and opinion structure, we describe the five OMs for the time step where the uncertainty \( u_j \) after meeting agent 4.

### 4.5. Uncertain, Subjective Opinion Model

#### Uncertainty-based OM

Uncertainty-based OM

Uncertainty-based discounting operator, \( uc_i^j \), as a specific \( c_i^j \) in Eq. (4.5), is derived by judging two users’ uncertainties as:

\[ uc_i^j = (1 - u_i)(1 - u_j). \] (4.6)

Uncertainty (or lack of confidence) from one’s opinions has been investigated as a deciding factor to reflect the information from the updates of users’ opinions [39]. Although uncertainty comes from many sources [94], we refer to uncertainty for two reasons: insufficient evidence and conflicting evidence. To represent the uncertainty property from both vacuity and conflict, the uncertainty (or vacuity) maximization technique [131] is also applied to prevent the uncertainty from going down to zero. This is because if an agent collects enough evidence for both belief and disbelief, it will reach zero uncertainty and stop accepting new evidence. The uncertainty maximization technique [131] can prevent this from not being updated and transfer evidence supporting the belief and disbelief masses to the uncertainty mass. This means moving conflicting evidence to vacuity. We use a threshold, \( \xi \), to determine whether to use this uncertainty maximization, i.e., apply only when \( u_i < \xi \) (i.e., only when uncertainty is sufficiently low).

The vacuity-maximized opinion for user \( i \) is defined by \( \tilde{\omega}_i = (\tilde{b}_i, \tilde{d}_i, \tilde{u}_i, a_i) \) where \( \tilde{u}_i, \tilde{b}_i \) and \( \tilde{d}_i \) are computed by:

\[ \tilde{u}_i = \min \left[ \frac{P(b_i)}{a_i}, \frac{P(d_i)}{1 - a_i} \right], \quad \tilde{b}_i = P(b_i) - a_i \cdot \tilde{u}_i, \quad \tilde{d}_i = P(d_i) - (1 - a_i) \cdot \tilde{u}_i. \] (4.7)

where the projected belief and disbelief \( P(b_i) \) and \( P(d_i) \) are from Eq. (4.3). The uncertainty maximization also plays a critical role in \( uc_i^j \), such that if \( u_i < \xi \), the vacuity-maximized \( \tilde{u}_i \) would replace the \( u_i \) in Eq. (4.6).
• **Homophily-based OM**: Homophily (or like-mindedness) between two opinions is critical for opinion updates [166]. Like [41] and [166], we also use cosine similarity [251] within the range of [0,1], as the homophily-based discounting operator, $hc_i^t$. This cosine similarity can adequately capture the measure of the distance between beliefs, including belief and disbelief masses, where we formulate an opinion based on Subjective Logic. This $hc_i^t$ can replace the $c_i^t$ in Eq. (4.5) by the following definition [41]:

$$
hc_i^t = \frac{b_i b_j + d_i d_j}{\sqrt{b_i^2 + d_i^2} \sqrt{b_j^2 + d_j^2}}. 
$$

(4.8)

We ignore uncertainty for the two opinions’ dissimilarity above because of the assumption of $b + d + u = 1$, where belief and disbelief can indirectly reflect the uncertainty.

• **Encounter-based OM**: We use this model as a baseline model in which the opinion is simply updated by the existing consensus method (i.e., $w_i \oplus w_j$) in SL [131] without applying any filters such as uncertainty, homophily, or assertion. This OM is implemented as $c_i^t = 1$ in Eqs. (4.4) and (4.5).

The following two opinion models are proposed from other existing research [241, 323]. For the fair comparison of the five OMs, we extend the subjective opinion to fit those two models as baseline counterparts to the OMs supported by the consensus operator as follows:

• **Assertion-based OM**: We use this opinion model as an existing counterpart. This model uses the so-called assertion [323], by $A_i = \{k_i, spb_i\}$, formulated based on knowledge and a subjective prior belief. The original update rules for the two values are $k_{i\oplus j} = k_i + k_j(1 - k_i)$ and $spb_{i\oplus j} = spb_i + k_jspb_j(1 \pm spb_i)$, where ‘+’ is for negative $spb_i$ and ‘-’ is for positive $spb_i$. We convert this model to our SL’s opinion with $(b_i, d_i, u_i, a_i)$ where $k_i$ quantifies the evidence of $b_i$ and $d_i$ and $spb_i$ is the base rate $a_i$. By converting $spb_i$’s range in $[-1, 1]$ to $[0, 1]$ for $a_i$ and maintaining $k_i$’s range in $[0, 1]$ for $b_i$ and $d_i$, the opinion update rule for this opinion model is formulated by:

$$
b_{i\oplus j} = b_i + b_j(1 - b_i), \quad d_{i\oplus j} = d_i + d_j(1 - d_i), \quad u_{i\oplus j} = 1 - b_{i\oplus j} - d_{i\oplus j}, \quad a_{i\oplus j} = a_i + b_ja_j(1 - a_i). \quad (4.9)
$$

• **Herding-based OM**: We adopt this model to update an opinion considering the bias towards a user’s neighbor (i.e., leaning more towards his/her neighbors’ opinions) [241] to emphasize the convincing power of the neighbors’ opinions. This model mainly relies on the neighbors’ opinions, so we consider this herding-based opinion update. The term ‘herding’ has been previously used in the network science domain when herding behavior indicates one’s behavior following his/her friends or neighbors [155]. Since we use an SL-based opinion format, for a fair comparison, we consider the following opinion update operator when each user updates his/her opinion upon the interaction:

$$
x_i = \min[1, x_i + \frac{u_i}{|F_i|} \sum_{j \in F_i} (1 - u_j)(x_j - x_i)], \quad x \in \{b, d, a\}, u_i = 1 - (b_i + d_i). \quad (4.10)
$$
4.5. UNCERTAIN, SUBJECTIVE OPINION MODEL

Eq. (4.10) above implies that user $i$ will consider his/her neighbor $j$’s opinions when being unsure of the opinion with high uncertainty ($u_i$). Neighbor $j$’s opinion with higher certainty (i.e., $(1 - u_j)$) will be more considered when updating user $i$’s opinion. This implies that the neighbor $j$’s opinion with lower uncertainty ($u_j$) has a more convincing power to user $i$. In SL, an opinion’s uncertainty represents how much confidence the owner has in the opinion. The rationale of the uncertainty-based opinion model is well aligned with the prior research that expert sources can influence persuasion because they can motivate recipients to more seriously consider the information provided by them compared to the information provided by non-experts [109, 253].

4.5.4 Interaction Model for Opinion Update

Users are assumed to share opinions with friends and update opinions during interactions. A user’s high posting frequencies tend to attract more interactions with other users. Hence, if a user tends to be more exposed to the information, the user will have higher chances to interact with the users posting more. In addition, a user can interact with another user based on the probability that the two users interact directly. In this chapter, we consider the following user activities:

- **Sharing**: This behavior is the precondition of an opinion update. A user can share his/her opinion by:
  - *Pairwise interaction*: Pairwise sharing includes receiving tweets or leaving comments or feedback, such as likes or other sentiments. We use $P_{fi}^j$ as $i$’s feeding rate to model the feeding behavior between two users.
  - *Posting*: The posting behavior is sharing posts or messages with all the friends. We use $P_{pi}^j$ as $i$’s posting probability to share with all the friends.

A user updates his/her opinion by interacting with one of the neighbor $j$’s. Each neighbor user $j$ is characterized by $P_{ji}^j$ and $P_{ji}^p$, the probabilities of leaving feedback (e.g., sentiments such as likes or comments) and posting, respectively. Each user $i$ judges the relative level of the neighbor’s sharing behavior to find user $j$ to interact. We assume that users like interacting with more active than less active users in a given OSN. We quantify user $i$’s likelihood, $P_{ij}$, to select $j$ for possible interaction, assuming $F_i$ is $i$’s friends:

$$P_{ij} = \frac{P_{ji}^f + P_{ji}^p}{\sum_{k \in F_i} (P_{ki}^f + P_{ki}^p)}, \quad (4.11)$$

where $P_{ji}^f$ and $P_{ji}^p$ are initialized by the features in the datasets (see Section 4.7). Users will interact with other users and act accordingly, including updating and sharing (i.e., $a_1^A$). Hence, when they share their opinions, their feeding and sharing probabilities are
also dynamically updated accordingly. For example, if users $i$ and $j$ interact, it will increase the feeding probabilities, $P_{ij}^f$ and $P_{ij}^f$, for both. If user $i$ takes sharing strategy $SU$ (see Section 4.6.4), the user $i$’s posting probability (i.e., $P_{i}^p$) will increase.

- **Maintaining a friend network**: A user can add new friends or make unfriending decisions based on the corresponding opinion differences. The projected difference $PD_{ij}$ between two opinions held by users $i$ and $j$ is obtained by [39]:

$$PD_{ij} = \frac{|b_i - b_j| + |d_i - d_j|}{2}. \quad (4.12)$$

This PD is symmetric such that $PD_{ij} = PD_{ji}$. All of the elements are at time $t$. A user will make the friending or unfriending decisions based on the PD by:

- **Friending**: Users can invite a friend if they have a tendency to make friends. This tendency is the probability of inviting a friend derived from the current number of friends. The probability of a new edge connecting to any node with degree $k$ is from the Price Model [213] as $p_k(k+1)/(m+1)$, where $m$ is the mean out-degree and $p_k$ is the fraction of nodes with degree $k$. For any user $i$, $\phi_i^1$ is a threshold to accept a friend and is scaled in the range of $[0, 1]$ at random following the Gaussian distribution. In an uncertainty-based OM, user $j$ will accept a friending request only by $u_i < \phi_j^1$; user $j$ in other OM types will accept it when $PD_{ji} < \phi_j^1$. Otherwise, user $j$ will always ignore a new friend request.

- **Unfriending**: It is the opposite process to friending. A user can dismiss a current friend if the user finds an opinion discrepancy from the friend user. To ensure a sufficient amount of updating $j$’s opinion, we bound uncertainty to $u_j < \phi_i^2$. Then user $i$ using uncertainty-based OM will unfriend $j$ if $\phi_i^1 < u_j < \phi_i^2$; while $i$ using other OMs (i.e., by assertion, herding, homophily, and encounter) will unfriend $j$ when $PD_{ji} > \phi_i^1$.

These two operations will change a network topology and affect a user’s influence in a network (e.g., centrality or social capital) as discussed in Section 4.4.

### 4.6 Game Theoretic Agent Model

The social network in this work is denoted by an undirected graph structure, $G(V,E,\Omega)$ in Figure 4.1 that holds users as $V$ and holds all the friendship connections as $E$ ($e_{ij} = 1$ only when $i$ and $j$ are friends). $\Omega$ is the collective of the subjective opinion $\omega_i$ for each user $v_i$. We describe how each agent is characterized by a set of features. In this game model, the agents have three roles: attackers, users, and a defender (i.e., service providers). They all take calibrated actions in response to disinformation in this OSN.
Table 4.2: Input and Output of the Game Model with Three Players

<table>
<thead>
<tr>
<th>Input (Strategies)</th>
<th>Attackers</th>
<th>Users</th>
<th>Defender</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_1^A, a_2^A, a_3^A, a_4^A$</td>
<td>$a_1^U, a_2^U, a_3^U$</td>
<td>$a_1^D, a_2^D$</td>
<td></td>
</tr>
</tbody>
</table>

### Output (Expected Payoff)

<table>
<thead>
<tr>
<th>Attackers</th>
<th>Users</th>
<th>Defender</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E P^A_k(a^U_k, a^D_k) = \sum_{\ell \in D} \sum_{m \in U} p^D_{\ell_m} \cdot p^U_{m \cdot u^i_{k\ell m}}$</td>
<td>$E P^U_m(a^U) = p^U_m \cdot u^A_{A^j} + (1 - p^U_m) \cdot u^A_{U^j}$</td>
<td>$E P^D_k(a^A_k) = \sum_{k \in A} p^A_k \cdot u^D_{k\ell k}$</td>
</tr>
</tbody>
</table>

Our proposed game-theoretic opinion models belong to the category of a networked game in a social network (i.e., game on networks), which has several properties:

1. A large number of participating players exist in a social network;
2. The given social network mediates the interactions between players and payoffs where the players only interact with their friends and each player’s payoff is estimated based on their opinion status updated based on their interaction with other friends and the way they interact (i.e., an opinion model);
3. It is not feasible to derive an exhaustive table to specify payoffs because there is randomness for online users (players) to interact; and
4. The proposed game considers dynamic, gradual interactions among players. Nash Equilibrium (NE) can predict the strategy distributions if all players know the moves of the other players. However, the actual choices of the human players (i.e., online users) may differ. According to behavioral game theory, this finding is likely due to the bounded rationality (e.g., limited memory or a lack of perfect observability) and simultaneous moves in each interaction; and
5. The collective user behaviors may vary over time due to different choices of actions taken in each single-shot interaction.

Each user’s opinion model reflects one’s preference in updating an opinion based on one’s behavioral propensity, so the formulation of the utility in a game follows this preference. Users aim to maximize their utility based on their preferences, which may look irrational to other users who use different opinion models. However, this also reflects real-world situations.

Now we explain the goals, strategies, and payoffs of the key three players in detail below. In Table 4.2, we present a summary of the input (i.e., strategies) and output (i.e., expected payoff) of each player’s payoff function.

### 4.6.1 Agents’ Features

We characterize each user $i$’s key attributes by:

$$F(v_i) = [\omega_i, P_i^f, P_i^p, \phi_i^1, \phi_i^2, \rho_i, bw_i]$$  \hspace{1cm} (4.13)

where $\omega_i$ is the subjective opinion, as shown in Section 4.5.1 by SL. $P_i^f$ and $P_i^p$ are the probabilities of $i$’s feeding and posting activities, $\phi_i^1$ and $\phi_i^2$ are thresholds for $i$ to identify the opinion difference between $i$ and $i$’s friend as described in Section 4.5.4, and $\rho_i$ is a tolerance threshold of $i$ to judge the current friend to be a suspect attacker upon exchanging
opinions (discussed in Section 4.7.3). The $bw_i$ refers to the user $i$'s betweenness, which is commonly used to capture the user $i$'s structural social capital [26] (i.e., higher betweenness refers to higher structural social capital). In this work, we use $bw_i$ to examine the impact of disinformation on betweenness, where individual users' structural social capital (or bridging) is analyzed using the betweenness. We chose ‘betweenness’ to measure an online user’s structural social capital (i.e., bridging). The reason is that the betweenness centrality has been commonly used to indicate a person’s bridging capability, which is the core aspect of social capital [153, 164, 265]. Based on our investigation in [274], we found that ‘betweenness’ represents one of the highly comparable centrality metrics (e.g., much more potent than $k$-shell [142], collective influence [186], or redundancy [26]).

4.6.2 Attacker Model

The malicious users, as attackers, deploy information deception tactics [145] to disseminate disinformation and fabricate or block true information to mislead the beliefs of legitimate users. SL opinion framework featuring uncertainty can materialize the theoretical deception strategies [145] where the whole set of strategies is denoted by $A = \{a^A_1, a^A_2, a^A_3, a^A_4\}$:

- **Degradation** ($DG; a^A_1$) is to confuse legitimate users by injecting noises into true information. In an SL-based opinion, $DG$ is modeled by sending out a highly uncertain opinion as $(b, d, u, a) = (0, 0, 1, 0.5)$.

- **Corruption** ($C; a^A_2$) generates false beliefs by injecting disinformation or replacing true information with false information. We model this by replacing a received opinion with a completely opposite opinion and then sharing it with a friend. For example, if an attacker receives an opinion $(b, d, u, a) = (0.7, 0.2, 0.1, 0.3)$ from a friend, the attacker forwards $(b, d, u, a) = (0.2, 0.7, 0.1, 0.3)$ to other friends.

- **Denial** ($DN; a^A_3$) prevents users from accessing true information by inhibiting true information flow. It can cause the vacuity of information sources which increases uncertainties and difficulties for users in judging the truthfulness of the information.

- **Subversion** ($S; a^A_4$) refers to an attacker’s deception by changing the user’s processing of perceived inputs. This attack aims to make targeted users trust credible information less while using more non-credible information. To launch an $S$ attack, the attacker will always forward false opinions, as discussed by $\omega_F$ in Section 4.5.2, to consistently increase the volume of disinformation.

As detailed above, an attacker’s strategy, $DG$, $C$, or $DN$, will modify the opinions received from its friends and share them for efficient propagation of the uncertain, noisy, conflicting opinions instead of pairwise interactions. Each attacker chooses a strategy with our game model’s highest expected payoff value. We calculate an attacker $i$'s expected payoff of a
strategy $k$ by considering the weighted sum of utility $u_{k\ell m}^{ij}$, caused by a specific condition of the defender’s strategy $\ell$ and user’s strategy $m$. The expected payoff of strategy $k$ is:

$$EP_A^k(a^D, a^U) = \sum_{\ell \in D} \sum_{m \in U} p^D_\ell \cdot p^U_m \cdot u_{k\ell m}^{ij}, \quad (4.14)$$

where $D$ and $U$ are the collections of all strategies for the defender and users. $p^D_\ell$ is a defender’s expectation of strategy $\ell$ (i.e., either $a^D_1$ or $a^D_2$ by terminating a suspect immediately or monitoring with caution). $p^U_m$ is a user’s expectation of strategy $m$ (i.e., one of $a^U_1$ − $a^U_3$). The attacker can obtain the probability distribution of taking each strategy by the defender and user based on the historical observations of $p^D_\ell$ and $p^U_m$.

The utility of a specific condition based on all assumptions, $u_{k\ell m}^{ij}$, is defined by benefit over an incurring loss as:

$$u_{k\ell m}^{ij} = ds(k, m, \omega_i, \omega_j) - g_\ell, \quad (4.15)$$

where the attacker’s benefit $ds(k, m, \omega_i, \omega_j)$ refers to how much attack strategy $k$ contributes to making user $j$’s opinion $\omega_j$ closer to false opinion $\omega_F$ in Section 4.5.2, by the cosine similarity of $\omega_j$ and $\omega_F$, when $k$ is taken and not taken ($\neg k$):

$$ds(k, m, \omega_i, \omega_j) = s(k, m, \omega_F, \omega_j) - s(\neg k, m, \omega_F, \omega_j), \quad (4.16)$$

where the first cosine similarity $s(k, m, \omega_F, \omega_j)$ from Eq. (4.8) considers the updated user’s $\omega_j$ when attacker $i$ has strategy $k$ and user $j$ has strategy $m$. The second cosine similarity $s(\neg k, m, \omega_F, \omega_j)$ refers to $j$’s opinion when attacker $i$ shares a legitimate opinion rather than a deceptive (or false) opinion with strategy $k$.

The cost $g_\ell$ measures attacker $i$’s loss when the defender takes action $\ell$ where $g_\ell$ is estimated by the mean similarity between true opinions $\omega_T$ and all existing user’s opinions, $\omega_j$’s.

### 4.6.3 Defender Model

As an OSN administrator, a defender aims to ensure a secure and safe OSN by not tolerating any presence of malicious users propagating disinformation. If the defender receives $N_R$ misconduct reports from legitimate users, the defender can take the following strategies whose set is $D = \{a^D_1, a^D_2\}$:

- **Terminating a malicious user** ($T; a^D_1$) is to remove the account and the corresponding connections with his/her all friends, aiming to ensure the safety and security of the given OSN. However, if the suspended user is actually legitimate (i.e., false-positive), the user’s reputation is ruined. If then, the user can lose his/her social capital due to removing all connections with others.
Monitoring a suspect user \((M; a_2^D)\) is to monitor a suspected user with no other actions. If this user is malicious, he/she may keep performing deception attack strategies which can endanger the security and safety of the given OSN. However, if the detected user is a legitimate user (i.e., false-positive), he/she can maintain current relationships with other users and social capital.

The defender selects the strategy with a higher expected payoff value. We quantify the defender’s expected payoff of strategy \(\ell\) by taking only an attacker’s strategies into consideration because the user’s activities are unrelated to the defender’s goal. The defender’s expected payoff is given by the weighted sum of utility \(u_{\ell k}^D\) as:

\[
EP_{\ell}^D(a^A) = \sum_{k \in A} p_k^A \cdot u_{\ell k}^D,
\]

where \(a^A\) is an action the attacker takes with a set of attack strategies \(A\) and \(p_k^A\) is the probability that the attacker chooses strategy \(k\). The defender can learn \(p_k^A\) based on historical reports of reported attackers. If a suspect user is reported by other legitimate users at least \(N_R\) times, the defender can make decisions toward this suspect based on the payoffs in Eq. (4.17). However, if the defender decides the reported suspect as an attacker, but it is a legitimate user indeed, it is a false positive, and the legitimate user is evicted. The \(u_{\ell k}^D\) is the utility of defense strategy \(\ell\) with the attacker’s deceptive strategy \(k\). In addition, the defender evaluates the utility of each strategy \(\ell\) by the overall impact on the OSN, as the gain over cost, as:

\[
u_{\ell k}^D = ds(\ell, k, \omega_T, \omega') - c_{\ell},
\]

where defender’s gain \(ds(\ell, k, \omega_T, \omega')\) is the protection of OSN by strategy \(\ell\), which refers to how closely the affected opinions \(\omega'\)'s of all the legitimate users are to true opinion \(\omega_T\). This gain calculates two cosine similarities in Eq. (4.8), for the cases of taking strategy \(\ell\) over not taking it (\(\neg \ell\)), as follows:

\[
ds(\ell, k, \omega_T, \omega') = s(\ell, k, \omega_T, \omega') - s(\neg \ell, k, \omega_T, \omega'),
\]

The defender’s loss term \(c_{\ell}\) quantifies the cost for the defender’s strategy \(\ell\) as two constants, i.e., \(c_T = 0.1\) for \(T(a_1^D)\) and \(c_M = 0\) for \(M(a_2^D)\).

4.6.4 User Model

OSN users consume useful information and interact with their friend users. We consider five user types corresponding to the opinion models described in Section 4.5.3, including Uncertainty-based (U), Homophily-based (H), Assertion-based (A), Herding-based (HE), and Encounter-based (E) user types. Accordingly, A user can be either of the five decision-makers (DMs) types: U-DM, H-DM, E-DM, A-DM, and HE-DM. The set of users’ strategies is denoted by \(U = \{a_1^U, a_2^U, a_3^U\}\) where each strategy is described as follows:
4.6. Game Theoretic Agent Model

- **Updating and sharing** \((SU; a_i^U)\) is to update the current opinion based on the received opinion and then share it with other friends.
- **Updating** \((U; a_i^U)\) is to update the current opinion based on the received opinion.
- **No updating** \((NU; a_i^U)\) is to ignore a received opinion and keep the current one.

In this game, user \(i\) interacts with a friend user \(j\), either a user or an attacker. We assume that user \(i\) can be aware of user \(j\)'s type based on the historical experience of the types of encountered users. That is, user \(i\) can estimate the probability of the interacted friend user \(j\) being an attacker by \(p_{U_i}^{A_j}\), and a legitimate user by \(p_{U_i}^{U_j} = 1 - p_{U_i}^{A_j}\) (i.e., not an attacker). Since a specific role of user \(j\) has a set of strategies when user \(j\) is an attacker or user, user \(i\) needs to choose a strategy \(m\) with the highest expected payoff by a weighted sum of the utilities when \(j\) is either an attacker or a user:

\[
EP_{m_i}^U(a_{U_j}) = p_{U_i}^{A_j} \cdot u_{m_i}^{U_iA_j} + (1 - p_{U_i}^{A_j}) \cdot u_{m_i}^{U_iU_j},
\]

where \(u_{m_i}^{U_iA_j}\) or \(u_{m_i}^{U_iU_j}\) is the utility based on the user \(j\)'s type. For an attacker, \(j\), the utility \(u_{m_i}^{U_iA_j}\) is defined by the weighted loss caused by each strategy of an attacker if user \(i\) accepts attacker \(j\)'s opinion. If \(i\) choose \(NU\), as \(m = a_i^U\), \(u_{m_i}^{U_iA_j}\) is 0, in:

\[
u_{m_i}^{U_iA_j} = \begin{cases} 
\sum_{k \in A} p_k^{A_j} \cdot s(m, \omega_F, \omega_i, \omega_j) & \text{if } m = a_i^U \text{ or } a_i^U, \\
0 & \text{if } m = a_i^U,
\end{cases}
\]

where \(p_k^{A_j}\) refers to user \(i\)'s belief that attacker \(j\) takes strategy \(k\) based on the user's historical observations. However, it may not be perfect; thus, we considered 90% accuracy of the belief accuracy. The cosine similarity \(s(m, \omega_F, \omega_i, \omega_j)\) by Eq. (4.8) is between \(\omega_F\) and the expected opinion of user \(i\) encountering attacker \(j\) taking deceptive strategy \(k\).

If \(j\) is a user, the utility \(u_{m_i}^{U_iU_j}\) is defined based on user \(j\)'s strategy \(m'\) in \(U_j = \{a_1^U, a_2^U, a_3^U\}\) and user \(j\)'s DM type. Hence, \(u_{m_i}^{U_iU_j}\) is given by:

\[
u_{m_i}^{U_iU_j} = \begin{cases} 
\sum_{m' \in U_j} p_{m_i}^{U_j} \cdot uc_{j_i}^{m'} & \text{if user } j \text{ is U-DM;} \\
\sum_{m' \in U_j} p_{m_i}^{U_j} \cdothc_{j_i}^{m'} & \text{otherwise},
\end{cases}
\]

where \(p_{m_i}^{U_j}\) is the belief of choosing strategy \(m'\) for user \(j\). To compute \(uc_{j_i}^{m'}\) and \(hc_{j_i}^{m'}\), we first update both users \(i\) and \(j\)'s opinions \(\omega_i'\) and \(\omega_j'\) by taking strategy \(m\) for user \(i\) and strategy \(m'\) for user \(j\) if an update is required for any of \(m\) or \(m'\). Afterward, the discounting operators \(uc_{j_i}^{m'}\) and \(hc_{j_i}^{m'}\) from Eqs. (4.6) and (4.8) are produced by \(\omega_i'\) and \(\omega_j'\), respectively.

The proposed three-player game is constructed by a series of repeated subgames, each of which is a game of incomplete and imperfect information. When each player plays, the opponent can be either one type or the other (e.g., a user or an attacker when a user plays).
CHAPTER 4. MITIGATING INFLUENCE OF DISINFORMATION PROPAGATION USING
UNCERTAINTY-BASED OPINION INTERACTIONS

Defender’s strategies = \{T, M\}

Attacker’s deception strategies = \{DG, C, DN, S\}

User’s opinion update strategies = \{S, SU, NU\}

Figure 4.2: The pairwise interactions of three agent roles, attackers, users, and a defender, in our networked game.

Since the opponent’s type is unknown in advance, each player will estimate its belief on the type of the opponent as described in this section. This is well-aligned with real-world scenarios because real situations are always filled with many aspects of uncertainties. To better reflect the imperfect observability of each player in its beliefs towards the opponents, we also model each player’s limited observability with 90% accuracy.

However, in game theory, Nash Equilibrium (NE) solutions are used to provide players’ best strategies, assuming that each player has a correct belief about its opponent’s move. We also formulate an incomplete information game with NE solutions that are compared against the strategy selections by the players under uncertainty, as described in this section. Appendix A.1 elaborates on the Nash game with three specific examples of how NE solutions can maximize the benefits of all players from the normal-form game trees and normal-form payoffs matrix, along with the corresponding detailed explanations of them. We elaborate on the interactions between the three key players in a single-shot interaction in Figure 4.2.

4.6.5 Opinion-based SIR Epidemic Model

The propagation of disinformation can alter the opinions of users. Each user can be assigned a status based on the projected belief \(P(b)\) and disbelief \(P(d)\) in the population view. By the SIR (Susceptible-Infected-Recovered) model, the Susceptible \(S\) users have \(P(b) \leq 0.5\) and
4.6. Game Theoretic Agent Model

\[ P(d) \leq 0.5. \] A user of \( P(d) > 0.5 \) belongs to the status Infected (I), and a user of \( P(b) > 0.5 \) stays in the Recovered state (R). The SIR model quantifies the transition dynamics from \( S \) to \( I \) and \( I \) to \( R \) by the infection rate \( \beta_t \) and recovery rate \( \gamma_t \). We use time-dependent \( \beta \) and \( \gamma \) because our opinion propagation is influenced by the decision-making process in the game model but not naturally transmitted by contact like the spread of disease. The ODEs solving this SIR model at any time \( t \) are:

\[
\begin{align*}
\frac{dS}{dt} &= -\beta_t S_I t, \\
\frac{dI}{dt} &= \beta_t S_I t - \gamma_t I_t, \\
\frac{dR}{dt} &= \gamma_t I_t,
\end{align*}
\] (4.23)

where \( S_t + I_t + R_t = N \) with \( N \) nodes in a given network. This SIR status and infection and recovery rates can represent the effect of disinformation propagation under the different opinion update models. However, the parameters \( \theta = \{\beta_1, \ldots, \beta_T, \gamma_1, \ldots, \gamma_T\} \) are only available from the game model simulation results. We investigate how each opinion model generates \( \beta_t \) and \( \gamma_t \), determining the extent of disinformation propagation in the network.

Parameter Optimization and Gradient Decent

Given the simulation of users’ opinions, we can calculate the \( S_t, I_t, \) and \( R_t \) for \( t \in [1, T] \). Then we need to fit those values to the ODEs in Eq. (4.23). We use gradient descent [22] to optimize the parameters of interest, i.e., \( \theta = \{\beta_1, \ldots, \beta_T, \gamma_1, \ldots, \gamma_T\} \). The objective function considering all \( T \) interactions is defined below:

\[
J(\theta) = \sum_{t=1}^{T} (\tilde{I}_{\theta,t} - I_t)^2,
\] (4.24)

where \( \tilde{I}_{\theta}(t) \) is the estimated number of infectious people at time \( t \) via the SIR model with parameters \( \theta \). However, the gradients are intractable since they are parameters of the SIR model, an ODE. We use the small difference (1%) between the objective function divided by the difference between the parameter to approximate a parameter’s gradient. For approximation, the \( \beta_t \)’s gradient is derived by:

\[
\nabla_{\beta_t} J(\theta) = \frac{J_t(\theta) - J_t(1.01 \times \theta)}{-0.01 \times \beta_t},
\] (4.25)

where \( J_t(\theta) = (\tilde{I}_{\theta,t} - I_t)^2 \). Note that the small difference (i.e., 1%) is applied by adding 1.01 in the numerator and -0.01 in the denominator in the above equation. After we get the gradients of parameters, i.e., \( \nabla_\beta(\theta_k) \) and \( \nabla_\gamma(\theta_k) \) for the \( k \)-th iteration, we use the gradient descent’s update rule to update the parameters to obtain \( \theta_{k+1} \) by:

\[
\beta_{k+1} = \beta_k - \eta \nabla_\beta(\theta_k), \quad \text{and} \quad \gamma_{k+1} = \gamma_k - \eta \nabla_\gamma(\theta_k),
\] (4.26)

where \( \eta \) is the learning rate.
SIR Status Prediction

When the lists of $\beta_t$ and $\gamma_t$ under each time point are available from the data-fitting step, we can estimate the future count of each $S_t$, $I_t$, and $R_t$ users from the ODE model in Eq. (4.23). Given the initial $S_0$, $I_0$, and $R_0$, we can predict the next-step values of each role, i.e., $S_1$, $I_1$, and $R_1$, from Eq. (4.23), by using $\beta_1$ and $\gamma_1$. This step can be iterated to predict the numbers of $S$, $I$, and $R$ at any future interaction time.

### 4.7 Experimental Setup

This section describes the datasets, metrics, and simulation experiment environmental setup.

#### 4.7.1 Datasets

From two real Twitter datasets, 1KS-10KN [295, 296] and Cresci15 [46], described in Chapter 3.7, we obtain active accounts with tweeting or retweeting behaviors. These datasets have a broad range of metadata, including user profiles, followers list, tweeting activities, and tweet contents. Thus, we extract complex social behavior features, such as favorite tweets, activity networks, and tweeting frequencies, to initialize individual sharing likelihood as $P^f_j$ and $P^p_j$.

#### 4.7.2 Metrics

We use the following metrics to evaluate the performance of the considered opinion models:

- **Opinions of agents ($\omega_i = (b_i, d_i, u_i, a_i)$):** SL opinions cover four dimensions, including belief ($b_i$), disbelief ($d_i$), uncertainty ($u_i$), and base rate ($a_i$) as introduced in Section 4.5.1. This metric can show the trend of diverging opinions or converging as more user interactions are performed. Hence, this metric will allow us to observe the extent of opinion polarization in an OSN.

- **Probability distribution of best-taken strategies:** This metric measures the frequency of a player’s chosen strategies with the highest payoffs during all repeated subgames. This metric enhances our understanding of each player’s preferences in both our proposed game under uncertainty and a Nash game.

- **Ratios of S, I, and R:** This counts the user status based on their SL-based opinions at each interaction time.

- **Infection rate $\beta$ and recovery rate $\gamma$:** These rates under each opinion model can affect the decision-makers differently.
4.7. Experimental Setup

- **Network communities**: Communities are generated by three popular graph partitioning algorithms, including Kernighan–Lin bipartition algorithm \([139]\), greedy modularity maximization algorithm \([43]\), and label propagation-based algorithm \([45]\).

- **Polarization scores**: Polarization refers to increasing differences in social, political, or attitudinal aspects between groups \([177]\). Accordingly, a polarization score indicates how different two groups are in those various characteristics of people in the groups. We consider methods measuring network polarization scores, including modularity \([43]\), community boundary connectivity \([87]\), random walk controversy \([76]\), and community performance \([74]\).

- **Structural social capital** (stc\(_i\)): We choose betweenness centrality \([75]\) to measure each user \(i\)’s structural social capital, denoted by bw\(_i\). Structure social capital (STC) measures how a person connects with others in a social network. STC is often captured as the degree of bridging in social capital, measured by betweenness \([26]\). We define a user’s STC based on the sum of the betweenness of the user’s friends, where we use a normalized STC, denoted by stc\(_i\) as a real number ranged \([0, 1]\) by:

\[
stc_i = \exp\left(\frac{-1}{\sum_{k \in F_i} bw_k}\right),
\]

(4.27)

where this metric reveals how individual users’ disinformation propagation and information processing methods can affect the degree of the STC. This can help identify key factors of creating or breaking the STC (i.e., bridging) when people exchange and update their opinions. We use Freeman’s betweenness centrality \([75]\) for bw\(_i\), which measures the degree of the shortest paths between two pairs of nodes going through given node \(i\), representing node \(i\)’s betweenness. If a tiny amount of users have high betweenness, it can represent the unbalanced distribution of network influence or power.

- **Trust** \((T_i)\): Trust measures the level of relational social capital, which is well-aligned with the quality of bonding among users. User \(i\)’s trust, \(T_i\), is quantified by how much other friend users, \(j\)’s, trust user \(i\) and given by \([40]\):

\[
T_i = \frac{1}{2|F_i|} \sum_{j \in F_i} (T^{f}_{ji} + T^{p}_{ji}),
\]

(4.28)

where \(F_i\) is a set of user \(i\)’s friends. \(T^{x}_{ji}\) is user \(j\)’s trust in user \(i\) in activity \(x\), including feeding \((f)\) and posting \((p)\) behaviors derived from social interactions, such as sharing information. The \(T^{x}_{ji}\) is calculated based on \(x\) activities, including the number of feeding \((f)\) or posting \((p)\) interactions, \(I^{x}_{ji}\), between user \(i\) and user \(j\) by:

\[
T^{x}_{ji} = \frac{I^{x}_{ji}}{\max(I^{x}_{jk} \text{ for } k \in F_j)}, \quad x \in \{f, p\}.
\]

(4.29)
Table 4.3: Key parameters, the explanations, and default values

<table>
<thead>
<tr>
<th>Param.</th>
<th>Explanations</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P_{true}$</td>
<td>Fraction of true informers</td>
<td>0.1</td>
</tr>
<tr>
<td>$P_{false}$</td>
<td>Fraction of false informers</td>
<td>0.1</td>
</tr>
<tr>
<td>$\xi$</td>
<td>Threshold for uncertainty maximization</td>
<td>0.05</td>
</tr>
<tr>
<td>$\phi^i_1$</td>
<td>Criterion to request or accept a friend</td>
<td>Normal(0.1, 0.1)</td>
</tr>
<tr>
<td>$\phi^i_2$</td>
<td>Upper bound to unfriend in uncertainty-based OM</td>
<td>0.5</td>
</tr>
<tr>
<td>$P^f_i$</td>
<td>Likelihood of agent $i$’s feeding behavior</td>
<td>0.142 (mean)</td>
</tr>
<tr>
<td>$P^p_i$</td>
<td>Likelihood of agent $i$’s posting behavior</td>
<td>0.186 (mean)</td>
</tr>
<tr>
<td>$\omega_T$</td>
<td>A true opinion with high belief and base rate</td>
<td>$(1, 0, 0, 1)$</td>
</tr>
<tr>
<td>$\omega_F$</td>
<td>A false opinion with high disbelief</td>
<td>$(0, 1, 0, 0)$</td>
</tr>
<tr>
<td>$c_T$</td>
<td>Cost of a defender’s taken strategies</td>
<td>T: 0.1, M: 0</td>
</tr>
<tr>
<td>$N$</td>
<td>Size of experiment sample</td>
<td>1,000</td>
</tr>
<tr>
<td>$I$</td>
<td>Number of interactions to choose a strategy</td>
<td>200</td>
</tr>
<tr>
<td>$\rho, \mu, \sigma$</td>
<td>Tolerance of a malicious user to generate a report</td>
<td>Normal(0.5, 0.05)</td>
</tr>
<tr>
<td>$N_R$</td>
<td>Number of malicious reports to alert the defender</td>
<td>3</td>
</tr>
<tr>
<td>$p_{A_j}^{U_i}$</td>
<td>Expectation of $j$ as an attacker by nature</td>
<td>0.1</td>
</tr>
</tbody>
</table>

4.7.3 Environment Setup

We consider an OSN with $N=1,000$ users randomly selected from datasets Cresci15 and 1KS-10KN for simulations. We assign the users with the top 20% degrees as true informers and false informers (or attackers), with $P_{false}=P_{true}=10$% users for each. The rest of the users (i.e., 80%) belong to the same type, i.e., one of the U, H, A, HE, or E-DMs described in Section 4.6.4. Each individual has a uniform distribution of possible actions for the three player types. This strategy distribution can be refreshed according to Dirichlet distribution [106] where the new outcome observed from each subgame contributes to a unit of supporting evidence. The top 20 topics of a user are generated by applying the latent Dirichlet allocation (LDA) algorithm to a collection of posted tweets and retweets. Based on lacking most followers from the real followers’ list, users connect to others with the highest topic similarity score [283] to form the initial friending network. We exclude an attacker-attacker relationship to maximize the coverage of disinformation propagation. The experiment is repeated by 100 runs, with each covering $I = 200$ interactions. Table 4.3 summarizes the key parameters and default values.

In the first interaction, each player starts a subgame by:

- **1-A**: An attacker takes Subversion ($S$) action by disseminating false opinions $\omega_F$ to one neighbor user.

- **1-B**: For all legitimate users, user $i$ first chooses friend user $j$ based on user $j$’s information sharing tendency $P_{ij}$ in Eq. (4.11). After then, user $i$ randomly selects a strategy, $SU$, $U$, or $NU$. If $SU$ or $U$ is chosen to update $i$’s opinions, user $i$ will use the OM in Section 4.5.3, corresponding to his type, and choose a strategy based on the payoff described in Section 4.6.4.

- **1-C**: If user $i$ accepts an attacker $j$’s opinion and accordingly updates $\omega_i$, attacker $j$ will modify $\omega_i$ with $j$’s chosen deception strategy to share the deceptive opinion with
another friend user in the next interaction, aiming to increase the uncertainty of other users’ opinions in the OSN.

- **1-D**: Each user decides to add a new friend or remove an interacting friend in this interaction based on Section 4.5.4.

This repeated game continues until the $I^{th}$ interaction:

- **2-A**: All legitimate users follow step 1-B except for deciding the best strategy based on the expected payoffs of available strategies in Eq. (4.20). User $i$ can report to the defender (i.e., an OSN service provider) that the interacting friend user $j$ is malicious if their opinion difference based on Eq. (4.12) is larger than the tolerance threshold $\rho$. Considering the varieties of individual users, $\rho$ follows a Gaussian curve with mean $\mu$ and standard deviation $\sigma$. Depending on the level of an individual user’s $\rho$, false positives can be generated.

- **2-B**: Each attacker selects a friend based on Eq. (4.11) and decides the best deceptive strategy based on the expected payoffs of the attack strategies in Eq. (4.14) to spread a deceptive opinion by taking the step 1-C.

- **2-C**: When the defender realizes that a suspect user receives at least $N_R$ misconduct reports, the defender can determine the best strategy based on the expected payoffs of the available strategies in Eq. (4.17).

- **2-D**: Each user repeats step 1-D to maintain the friend network.

Note that the roles of attackers and users do not change. In particular, regardless of whether the users hold true information or disinformation, they will propagate their opinions based on their sharing behaviors and update their opinions based on their preferences via various opinion models.

### 4.8 Simulation Results & Analysis

We present the results under $1KS$-$10KN$ [295, 296] and discuss their underlying trends in this section. By default, we presume 10% true informers, 10% false informers, and 80% of the agents (i.e., legitimate users) all used the same opinion model in Section 4.5.3. In Appendix A.2, we plot the distribution of strategies in our opinion game players when there are mixtures of H-DMs and other DMs. The sensitivity analysis of various false informers ratios $P_{false}$ is in Appendix A.3. Under Cresci15 [46], we observe that the overall trends are significantly similar and present those results in Appendix A.4.
Figure 4.3: The evolution of SL-based opinions of all legitimate users over 200 interactions in belief ($b$, blue), disbelief ($d$, red), uncertainty ($u$, green), and base rate ($a$, grey) under the dataset 1KS-10KN [295, 296].

4.8.1 Uncertain Opinions in Disinformation Propagation

Figure 4.3 illustrates the trends of opinion dynamics under the five OMs throughout the 200 interactions where all legitimate users are set to the same user type corresponding to the OM in each subfigure. The three opinion dimensions ($b$, $d$, $u$) belief, disbelief, and uncertainty, are plotted by contrasting colors. As expected, different opinion OMs introduce distinct impacts on the dynamics of opinions. From the view of belief $b$ supporting true information, opinions from U-DM and HE-DM have the top two highest $b$ in Figures 4.3a and 4.3d. U-DMs can form the highest uncertainty $u$ as well. In Figure 4.3a, the increase in accepting true information via more interactions with friends is mainly caused by uncertainty when two users update their opinions based on uncertainties from both opinions.

Moreover, U-DMs tend to neglect the attackers’ noisy, deceptive, and uncertain opinions to amplify false information or disturb true information. In Figure 4.3b, H-DMs have opinions with low uncertainties and less compromise with other opinions, indicating high opinion polarization. On the other hand, all opinions from both the A-DM and E-DMs can form a consensus with low beliefs and low uncertainties, while E-DM in Figure 4.3e has a similar
4.8. Simulation Results & Analysis

Figure 4.4: The histograms of three opinion masses (i.e., \((b, d, u)\)), base rate \(a\), and the projected belief \(P(b)\) for all normal users after 200 interactions. Each subfigure shows the histograms for the five OMs in a given element under the dataset 1KS-10KN [295, 296].

level of \(b\) and \(d\). A-DM in Figure 4.3c maintains equal \(b\), \(d\), and \(u\).

Figure 4.4 examines the histograms for five OMs in the same range by analyzing all users’ opinions after all interactions. Besides the four opinion components, we plot projected beliefs, \(P(b)\)’s, to know each user’s preference. Those results are correlated with the final states in Figure 4.3 and confirm the results of the U-DM: the highest \(b\) in Fig 4.4a and \(P(b)\) in Fig 4.4e, lowest \(d\) in Fig 4.4b, and high \(a\) in Fig 4.4d with the initial value of 0.5. Thus, disinformation is significantly mitigated by the U-DMs’ ability to report malicious users propagating highly uncertain opinions to the defender. H-DMs all have \(b\), \(d\), and \(a\) around 0.5 and reduce uncertainty in the starting interactions. The results from the H-DMs suggest that users relying on homophily may trust disinformation more because they cannot identify noisy and uncertain opinions if the observed opinion difference is less than \(\phi^1\).

4.8.2 Strategy Selection, Payoffs, and Nash Equilibrium

Figure 4.5 shows how the selected strategies in our game-theoretic framework differ from the NE solutions under the five OMs. Recall that NEs are derived based on the assumption of perfect observations so that the players can intelligently achieve mutual benefits based
CHAPTER 4. MITIGATING INFLUENCE OF DISINFORMATION PROPAGATION USING UNCERTAINTY-BASED OPINION INTERACTIONS

Figure 4.5: The probability distributions of the taken strategies by each player type based on NE solutions and the solutions by our proposed game under the dataset 1KS-10KN [295, 296].

(a) Attackers

(b) Users

(c) Defender

Figure 4.5: The probability distributions of the taken strategies by each player type based on NE solutions and the solutions by our proposed game under the dataset 1KS-10KN [295, 296].

on their accurate beliefs towards the opponents. In both attackers and users shown in Figures 4.5a and 4.5b, respectively, the probability distributions of taken strategies in NE and our game are inconsistent. This is because players can only partially observe the opponents’ previous actions in this repeated game.

Figure 4.5a depicts that attackers’ NE solutions for U, A, HE, and E-DMs are dominated by S, while those for H-DMs favor C 40% more. The homophily-based OM produces more similar probability distributions of the attackers’ strategies taken under both our and NE games than other OMs. Under the taken strategies from our game, the attackers in the H-DM network have an equal chance for each strategy; while the attackers in all other OM networks prefer S and C. Legitimate users’ NE solutions in Figure 4.5b for all OMs have a significant increase of U and SU to accept other opinions more, compared to the distributions of U and SU in our game. This comparison suggests that users in our designed model have more resistance to accepting other risky or uncertain opinions when handling disinformation.

Our game model has the opposite situation for the users, where all OMs choose NU at least at 50% probability in Figure 4.5b. Besides, U-DM and H-DMs are the least and most motivated to update opinions. The defender’s choices from NE and our game model fit well, except for the assertion-based OM in Figure 4.5c. Under both conditions, the defenders in the U-DM and HE-DM networks are more likely to remove malicious accounts, while T is
4.8. Simulation Results & Analysis

Figure 4.6: The comparison of the average payoffs of the strategies taken by each player type based on the NE and our proposed games in 1KS-10KN [295, 296].

rarely selected in H-DM and E-DM networks. However, the defender in the A-DM network has a relatively equal chance of taking the $T$ or $M$ strategy from NE solutions rather than having a solid bias to $T$ in our game.

The payoff values in Figure 4.6 can help understand the differences in best strategies between each user type and between NE solutions and the solutions taken by the players. The ranking of payoff values for strategies under each player type (i.e., an attacker, user, or defender) in Figure 4.6 correlates well with the best strategies chosen in Figure 4.5. This result confirms the preferences of the best strategies for a specific player type. In Figure 4.6a, the payoffs from NE solutions are all higher than the corresponding strategies taken. Also, the payoffs from the strategies chosen in our game are all negative, whereas the payoffs from NE-based strategies taken by A-DM and HE-DMs are positive. The higher NE payoffs are also found for the user strategies in Figure 4.6b and some of the defender’s strategies, such as the defender in U-DM, A-DM, and E-DMs in Figure 4.6c. In Figure 4.6b, the U-DMs have much lower payoffs compared to other DMs because, in Eq. (4.22), the utility of user $j$ depends on $j$’s type. If user $j$ is a U-DM, it will use $uc_{im}^j$, which is quite different from the utility $hc_{im}^j$ for all other DM $j$’s. In Figure 4.6c, the payoffs of strategy $M$ for all are zero due to how we estimate the payoffs in Eq. (4.18).
CHAPTER 4. MITIGATING INFLUENCE OF DISINFORMATION PROPAGATION USING
UNCERTAINTY-BASED OPINION INTERACTIONS

Figure 4.7: The SIR curves of five opinion models in our proposed game, including $R_m$ as the Removed nodes by the defender.

4.8.3 Analysis of Opinion Dynamics Using the SIR Model

Analyzing the opinion status by the SIR epidemic model, the dynamics of all false informers, true informers, and legitimate users in each of $S$, $I$, and $R$ are demonstrated in Figure 4.7. Another state Removed is included because the defender can terminate the malicious accounts if it chooses strategy $a_{1T} = T$. The curves of the SIR status show the different effects of disinformation propagation in the five DMs. The Infected users in all non-H-DMs increase and then decrease. The decrease of $R$ in U-DM and A-DM correlates with increased Removed nodes. However, no removed users or attackers exist in H-DM and HE-DM curves.

The gradient descent is applied in Section 4.6.5 to optimize the infecting rate $\beta$ and recovery rate $\gamma$ in the ODEs of users. The fitting curves of Figure 4.7 are presented in Figure 4.8 with the ranges of $\beta_i$ and $\gamma_i$. There is no Removed curve in Figure 4.8 because it is not modeled in the SIR model and this causes the higher numbers of $R$ in Figures 4.8a, 4.8c, and 4.8d. The curves of $S$, $I$, and $R$ in both figures are well-matched. Both A-DM and H-DM in Figures 4.8d and 4.8e have large infection and recovery rates because the SIR curves stay stable after the first few interactions. The H-DM in Figure 4.8b maintains a large number of $I$ throughout the interactions so that $\beta$ and $\gamma$ are close to 0 when the users are polarized into two extreme groups, and they hardly change their opinions. In a few beginning interactions, the H-DMs have a higher $\beta$ and lower $\gamma$ than the U-DMs in Figure 4.8a.
4.8. Simulation Results & Analysis

4.8.4 Effect of Disinformation Propagation on Polarization

Figure 4.9 compares network topology changes between the communities of the initial state and those after disinformation propagation under the five different opinion models (OMs). Figure 4.9 shows distinct communities formed depending on a different community detection algorithm with disparate clusters in the plots. The node colors reflect each node’s projected belief \( P(b_i) \) in a color bar, where belief in true information is in blue and belief in disinformation is in red. Green in subgraphs (a), (g), and (m) represents the degree of the initial projected belief, i.e., \( P(b_i) = 0.5 \). We used three community detection algorithms described in Section 4.7.2 to generate network communities. We also demonstrated the community topologies by separating individual communities and placing users within the same communities closer. Since each user has his/her opinion by updating their opinions through the interactions with other users after disinformation propagation, we plot the projected belief \( P(b_i) \) of the individual user using a color bar by showing complete belief supporting true information (\( P(b_i) = 1 \)) as blue and perfect disbelief supporting disinformation (\( P(b_i) = 0 \)) as red. We showed the initial uncertain opinions (\( P(b_i) = 0.5 \)) in green in Figures 4.9a, 4.9g, and 4.9m. Although various community detection algorithms generate different communities, they all have the same trends to reveal users’ opinion polarization caused by disinformation propagation. Most U-type users form their opinions as true information.
CHAPTER 4. MITIGATING INFLUENCE OF DISINFORMATION PROPAGATION USING UNCERTAINTY-BASED OPINION INTERACTIONS

Figure 4.9: The community plots of network topology based on three community detection algorithms, bipartitions, modularity, and label propagation, to show networks before and after disinformation propagation under the five OMs under the dataset 1KS-10KN [295, 296].
On the contrary, H-DMs in two communities believe either highly true or highly false information from all three community detection algorithms. This implies that disinformation can increase the polarization of the network if users update opinions based on like-minded attitudes, such as homophily. Herding-based DM also forms communities that have high beliefs similar to U-DM. However, Figures 4.9e, 4.9k, and 4.9q’s lighter blue color means that the projected belief is lower than U-DM communities. Both A-DM and E-DM networks generate polarized communities, although the polarization is not as strong as the H-DM network because the node color shows higher beliefs (more blue-wish) in the A-DM and E-DM networks than in the H-DM network.

Figure 4.10 demonstrates the polarization measures under three community algorithms. Polarization scores from the network influenced by U-DM are all lowered except for the community boundary scores within label propagation communities. The lower polarization denotes that in processing disinformation, U-DMs reduce more gaps between communities. Polarization scores from the H-DM network in Figure 4.10c increase under all four methods compared to the initial polarization scores. In the other two communities, the polarization scores increase under two methods while decreasing under the other two methods in Figures 4.10a and 4.10b. However, the H-DMs have the highest polarization scores compared with other OMs. Higher polarization means a more polarized network when diffusing disinformation in the network of U-DMs than in the network of other DMs. All results from network polarization analyses strongly support that the uncertainty-based opinion model can help users share uniform opinions and become united. On the contrary, the homophily-based opinion model divides users into more polarized opinion groups.

4.8.5 Effect of Disinformation Propagation on Social Capital

Figure 4.11 plots the distribution of all users’ bridging social capital in the five OMs before and after 200 user interactions upon disinformation. Since most betweenness values are below 0.001, we only plot the ones large than 0.001 for better visualization. In all the five user types, we can observe the remarkable decrease of betweenness after 200 interactions in Figure 4.11f, showing the STC of all the nodes in the initial and five DMs. The lower betweenness indicates weaker structural holes because the network is connected less than the original network after disinformation propagation. For H-DMs in Figure 4.11b, the users have the largest betweenness compared to other types. This implies that H-type users have more partial network power and influence by having more structural holes with higher STC.

Figure 4.12 plots the distribution of all users’ trust, as relational social capital (RC), in the five OMs after 200 user interactions upon disinformation propagation. All five OMs show a lower RC after the interactions in Figure 4.12f. Like the STC, trust values in H-DM networks are the highest among all five OM networks. This implies that in the H-type network, users with high homophily gather more tightly. Hence, it may be hard for the H-DMs to be with other users with a lack of similarity, leading to high opinion and network polarization.
4.9 Key Findings

Here we conclude our work as below:

- Uncertainty-based OM can assist users in excluding false, contradicting, and uncertain information and believing and accepting true information. However, other OMs based on homophily, assertion, and encounter can easily mislead users into believing false information from Section 4.8.1.

- Users using the uncertainty-based OM can also contribute to mitigating disinformation by effectively reporting suspicious friends to defenders, i.e., an OSN administrator. The misconduct reports can help defenders terminate malicious users to block the negative influence of disinformation from Section 4.8.2.

- The inconsistency of the actions taken in our game and NE game, shown in Figures 4.5 and 4.6, Section 4.8.2, was because our game model considered more realistic action...
4.9. KEY FINDINGS

(a) Uncertainty-based OM (b) Homophily-based OM (c) Assertion-based OM
(d) Herding-based OM (e) Encounter-based OM (f) Sum betweenness

Figure 4.11: The betweenness scores before and after 200 user interactions and opinions updates with respect to the five opinion models (OMs) under the dataset 1KS-10KN [295, 296]. In (f), we also indicated the mean betweenness of an individual user in the label on the top of each bar.

(a) Uncertainty-based OM (b) Homophily-based OM (c) Assertion-based OM
(d) Herding-based OM (e) Encounter-based OM (f) Sum trust

Figure 4.12: The trust scores before and after 200 user interactions under the five opinion models (OMs) under the dataset 1KS-10KN [295, 296]. In (f), we also indicated the mean trust of an individual user in the label on each bar.

scenarios where players might not perfectly keep track of correct observations towards the opponents’ previous actions.

- All network polarization analysis results in Sections 4.8.3 and 4.8.4 strongly supported that the uncertainty-based opinion model can help users share uniform opinions and
become united. On the contrary, the homophily-based opinion model divided users into more polarized opinion groups.

- Disinformation propagation could cause a decrease in social capital in terms of both structural social capital and relational social capital. Homophily-type users had more partial network power and gathered more tightly than other types from Section 4.8.5.

- A user’s game-theoretic process of disinformation propagated in a network can significantly affect the dynamics of the network in terms of network topology and network influence, represented by communities, polarization, and social capital. Compared to all other OMs, homophily-based OM caused the highest polarization in the network, while uncertainty-based OM helped users access true information best.
Chapter 5

mudRIA: Multidimensional Uncertainty-Aware Deep Reinforcement Learning-based Intent Analysis of Fake News Spreaders

This chapter addresses the Understanding Task. It develops a framework mudRIA for intent classification of the online fake news spreaders by multi-dimensional uncertainty-aware deep reinforcement learning (DRL). This work investigates a series of policy gradient-based DRL models based on our accepted paper [96] and papers in review [97, 98].

5.1 Motivation & Research Goal

We easily assume fake news is propagated by people only with malicious intent, such as intentionally ruining someone’s reputation or taking selfish financial benefits by doing it. However, recent social science studies [11, 12, 122, 144, 234, 250] show that people with no malicious intent often share fake news on online social networks (OSNs) because they are unaware of the lack of credibility or truthfulness in the fake news. Even they may have different, benign reasons to share the fake news. For example, OSN users spread information only for social purposes, such as entertaining others or sharing useful information with good intent (e.g., helping others by fundraising or encouraging social good). When online users receive information, they may fact-check to determine sharing or share without fact-checking. This sharing behavior in OSNs is influenced by the OSN users’ personality, propensity, or judgment capability.

Understanding and analyzing the intent of spreading fake news is critical for the following reasons. Firstly, based on text mining of fake news, we can analyze the information sharers’ intent and develop effective and efficient defenses (or interventions) against them. For example, a fake news sharer with no bad intent can be warned to stop propagating the fake news with some explanations. In addition, we can provide cybersecurity education/training programs to combat fake news for misinformers who propagate fake news mistakenly, as they do not have sufficient judgment capability to detect fake news. However, a fake news sharer
with malicious intent should be penalized more severely, such as suspending or terminating the account. Furthermore, detecting the intent of fake news spreaders is critical in measuring the negative influence of fake news over OSNs. Depending on the degrees of the negative influence, we should take more robust and proactive actions to combat them.

As there is a lot of fake and true news, we can predict their spreaders’ intent of propagating that fake or true news by intent mining from the available texts. Our intent prediction belongs to the group of text classification tasks. However, some basic models may not have good accuracy in text classification, especially for predicting multiple classes. In general, there are two ways to improve the accuracy: extracting useful features or optimizing the encoding structure. We want to analyze some intent-related semantic and lexical features by the existing text analysis tools to help our intent classifiers. Also, we can optimize the deep learning features of embedding and structure representation [303] by DRL. Deep reinforcement learning has improved the prediction of classes by optimizing sentence structure representation. Existing text classification DRL models can remove the noises for a particular sentiment class, such as the irrelevant words features and lexical features [35, 277, 281, 308].

With the promising DRL-based text classification approaches, we can analyze the fake news spreaders’ intent by their fake news text contents and improve the accuracy by learning better sentence structure representations. Also, removing intent-irrelevant words from fake news texts can help the models predict multiple classes more accurately. However, the existing DRL-based text classification approaches have limitations. In an episode of the Markov decision process, all steps receive a future return as the same delayed reward to calculate the gradient and update the policy parameters. Because the intent prediction score of fake news can only be accessed after the text encoding of the whole fake news, a delayed reward is precisely the final text encoding step’s immediate reward, leaving no immediate reward for any previous step.

In this work, we aim to provide solutions by two tracks to solve the issues of equal reward for each step and insufficient local responses caused by this unique delayed reward. Therefore, we propose an intent classification framework that can analyze the intents of fake news spreaders by DRL models. We develop a novel DRL model with a multi-dimensional uncertainty-aware reward function. In particular, we consider local critic-derived multi-dimensional uncertainties to allow local steps to trust a higher level of the delayed reward when the local critic shows a high certainty level. In the meantime, we reduce the gradient variances by an episode-level advantage value with the help of a pre-trained critic network. Our proposed framework, mudRIA, represents multidimensional uncertainty-Aware deep Reinforcement learning-based Intent Analysis framework of fake news spreaders.

5.2 Research Questions

In this work, we aim to answer the following research questions:
5.3. Key Contributions

1. What are the major intents of fake news spreaders?

2. What key factors can improve the accuracy of intent classification of fake news spreaders in our proposed intent classification framework, mudRIA?

3. What key benefits are introduced by the proposed mudRIA compared to other counterparts?

5.3 Key Contributions

We summarize the key contributions of our proposed mudRIA as follows:

1. Achieving both effectiveness and efficiency of intent classification models: We apply several DRL policy gradient (PG) models to predict fake news spreaders’ intent from texts of fake news in the mudRIA framework. Our DRL intent classifiers feature both effectiveness and efficiency from a delayed reward. A delayed reward combines two objectives of intent classification by maximizing the accuracy of its classification and minimizing the number of optimized words in fake news. No prior work has combined these two objectives in DRL for intent mining.

2. Enhancing the limitations of a delayed reward in text classification DRL: Some natural language processing (NLP) problems, such as text classification problems, have a unique aspect in that the delayed reward is the same as the final step’s immediate reward. Since a standard immediate reward is unavailable, each step obtains the same delayed reward to calculate gradients. To resolve this issue, we allow a step to trust a delayed reward with a higher weight if a local critic network-derived Subjective Logic (SL) opinion achieves a high certainty level. Accordingly, we calculate the policy gradients by a new multi-dimensional uncertainty-aware reward function (mUR) and demonstrate an increased accuracy for intent prediction.

3. Reducing gradient variances in text classification DRL: Policy gradient models usually suffer large gradient variances, but the regular step-level advantage estimation is unavailable because of missing immediate rewards. Leveraging the text classification model properties, we apply Advantage Actor-Critic (A2C) with an episode-level advantage from a pre-trained critic network. Since an episode-level advantage is identical for each step, we have combined the strengths of gradient reduction and a local certainty level to accomplish our intent prediction goals.

4. Validating with reliable datasets: To validate and test our intent classifiers with reliable datasets, we use the publicly available fake news dataset and annotate each piece of fake or true news with a label of spreaders’ intent. We assign a dominant intent class to each news data from three human annotators.
5.4 Background & Related Work

This section provides an overview of the related literature regarding intent mining, sentiment analysis of fake news, and DRL-based text mining.

5.4.1 Intent Mining

Purohit and Pandey [214] classified intent types for people to use social media and share information. They classified 13 specific intents based on five groups, including the intent of social good (e.g., helping others), social bad (e.g., spam or harassment), social ugly (e.g., propagating fake news with malicious intent), a mixture of good and bad, and a mixture of bad and ugly. The social good includes offering help, supporting emotionally, and sharing expertise. The social bad involves propagandizing, deceiving, and rumorizing. The social ugly includes harassing, manipulating, and bullying. The mixture of social good and bad includes joking and marketing. The mixture of bad and ugly has the intent of accusing and sensationalizing. Intent mining can be completed by content-, user-, and network-based methods, and intent can be represented by different information modalities, such as texts, images, audio, or video. Content-based methods were used in task domains, such as search queries, question-answering, ratings, and reviews. They have two types: rule-based and machine learning-based. User profile-based methods use historical activity patterns or messages to learn a user’s profile.

Alsmadi et al. [8] discussed the reasons for propagating fake news. First, people spread fake news motivated by political ideology, economic interests, eagerness for publicity, or a mixture of them. Those reasons can influence cognitive decisions. Second, people spread fake information because they had the fatigue to do the fact-checking; thus, they posted unverified information. Third, some people share information to help others without fact-checking. People share information intentionally or unintentionally. The reasons for the unintentional sharing of false information were routine social activity, publicity and attention, lack of resources, or emotional vent (e.g., anger, joy, depression). When people share information intentionally, the intent is related to causing political or religious polarization or neutralizing the effect of fake news by sharing counteracting information.

Hamroun and Gouider [104] discussed the structure of intention components as a verb, action, target, intention holder, and time of intention posted. They summarized the intent analysis work by four tasks. The intent classification task had the most significant number of research. The other three are spam detection, building resources, and transfer learning. Celliers and Hattingh [31] summarized the root causes of fake news spreading from existing literature. The root causes were similar to the motivations of the intentional fake news spreading. They generalized five major groups of factors, including social, cognitive, political, financial, and malicious factors, where each group has several specific intents. Social factors influenced users to seek similar opinions. Cognitive factors represented individual knowledge.
or intellectual characteristics. Political, financial, and malicious factors were derived from bad intent, such as hate propaganda.

Liu [170] discussed intention mining in a different way. There should be explicit and implicit intentions from the two definitions of intention. The explicit intention is “a course of action that a person or a group of persons intends to follow.” The implicit intention is “the goal or purpose behind a specific action or set of actions.” Since implicit intentions are hidden and subjective, this work only extracts explicit intentions, which implies future actions. The authors also analyzed emotional intentions based on sentiments and rational intentions that did not have sentiments. They defined an intention as action, target, intensity, holder, and time, similar to the intent structure in [104]. Non-trivial social science research has been conducted to learn the motivations of users sharing information (not fake news or false information) and their intent behind them. However, little work has studied the reasons for spreading fake news or rumors. Most of the literature has demonstrated that the main reason for spreading fake news or misinformation (i.e., spreading false information unintentionally or with good/neutral intents) was because they were unaware of whether given information is false (i.e., ignorance). Apuke and Omar [11, 12] studied the motivations of fake news-sharing behaviors in Africa. They designed predictors such as altruism, entertainment, socialization, passing the time, information seeking or sharing, self-promotion, and instant news sharing. The authors collected data in a survey and analyzed data by users and gratification framework to test the hypotheses. They found that altruism was the most significant motivation for sharing fake news, while entertainment showed no relation with it.

Koohikamali and Sidorova [144] surveyed the intent of information re-sharing behaviors in social media. They set up a model to show the influence of information quality (measured by information relevance, information reliability, and information enjoyment) on people’s attitudes towards using Social Network Services (SNS) and the effects of intention to information re-sharing. Re-sharing was defined as receiving some information first and sharing it with other users, which was defined as the sharing behavior in other work. They also found that re-sharing behavior was positively related to users’ risk-taking propensity.

Islam et al. [122] used cognitive load theory to study how motivations, including self-promotion and entertainment, and personal characteristics (e.g., exploration, religiosity, and deficient self-regulation) can influence social media fatigue and the sharing of unverified information. They showed that deficient self-regulation, exploration, and self-promotion were the most important factors, while all the factors were related to spreading unverified information. Shen et al. [234] studied psychological motivation for rumor spreading by both information characteristics and psychological attributes. They hypothesized that the information factors were more related to sense-making, fun, dreadfulness, and personal relevance, while the psychological factors were more influenced by fact-finding, self-enhancement, and relationship enhancement. This study concluded that all eight factors could promote online rumor-forwarding behavior. Talwar et al. [250] studied the bad intentions (dark side) in spreading fake news. They tested the intentional factors in terms of online trust, self-disclosure, fear of missing out, and social media fatigue positively related to sharing fake
news. However, social comparison (i.e., finding a similar person for comparison to evaluate one’s own ability) was negatively related to sharing fake news. Their experiment also showed that those factors could influence fact-checking behaviors before sharing, and online trust negatively influenced fact-checking.

Recently, there have been several natural language processing (NLP) methods to do intent mining in different task domains, such as product consumption [59], web queries [70], user reviews [140], music players [308], and email intent [237]. Ding et al. [59] used a convolutional neural network (CNN) to discover the product consumption intention words from social media posts. They used an intention word extraction algorithm and recommended a list of products to the intention word. Figueroa [70] classified three intentions as navigational, resource/transactional, and informational in the AOL web query collection. They detected the intents using SVM, maximum entropy, and naive Bayes. Zhang et al. [308] studied the intents from customer reservation system actions in the capsule neural network as a group of neurons. Each capsule was for different types of tokens, such as intent, slot, or word capsules. The intent capsule can capture higher-level features and then guide a dynamic routing among slot and word capsules to achieve synergistic effects.

However, bad or ugly intent detection or analysis has rarely been studied. Only two works discussed intention detection in an underground crime forum [27] and hate intention detection [112]. Caines et al. [27] used a crime blog corpus, including six forums, to analyze the cybercrime intents from the posts. They aimed to identify a post type, author intent, and target user. They labeled the author’s intents as positive, neutral, negative (from sentiment), arbitrate, vouch, gratitude, private message, and aggression. They can annotate the data with multiple labels, and the intents of vouch and gratitude belong to the positive intent type. They tested rule-based logical models and statistical models of Support Vector Machine (SVM), XGBoost, and linear regression using TF-IDF (Term Frequency-Inverse Document Frequency) features. Holgate et al. [112] analyzed the intentions of vulgar words by six categories in the hate speech detection dataset. In their annotation, the six intents were expressing aggression or emotion, emphasizing, auxiliary, signal group identity, and non-vulgar. In the logistic regression model, the authors added intention distribution features. The results showed that adding six vulgar group features improved the F1 score.

### 5.4.2 Sentiment Analysis of Fake News

Sentiments in fake news have been actively analyzed in recent studies. Ajao et al. [3] studied the relationship between fake news and sentiments by the emotional ratio score of negative words over positive words from the psychological and linguistic features in the LIWC (Linguistic Inquiry and Word Count). Then, they compared the emotions of rumor and non-rumor tweets in the PHEME dataset and used several ML models to make predictions. The combined features in the SVM model reached the highest accuracy.
Zhang et al. [312] discovered dual emotions for fake news detection, including a publisher’s emotion and followers’ high-arousal social emotions. The dual emotion features mainly used several DL models to predict fake news. The authors also used features from emoticons, punctuation, sentimental words, and personal pronouns. They evaluated the proposed algorithms under three datasets from social media posts to examine if the dual emotion features can improve fake news detection. This work also compared dual emotions to two other works using emotion features, Emoratio and EmoCred, to show the outperformance of the dual emotion-based approach.

Rangel et al. [217] described the overview of the PAN 2020 task, which is to profile fake news spreaders on Twitter. This project used only the linguistic features, including n-grams, stylistics, personality, sentiment, emotions, and embeddings, to predict fake news spreaders. Giachanou et al. [80] also examined if a user is a potential fake news spreader or a fact-checker based on personality features and linguistic patterns. They evaluated their algorithm in CNN and other ML models and found that CNN outperformed Long Short-Term Memory (LSTM).

The most recent work combined embedding features with other emotion features to make good predictions of fake news or spreaders. The limitation is that they did not filter noisy features from embedding the whole sentence. Hence, in this work, we will consider DRL to extract a better structure representation to make an accurate intent classification.

5.4.3 Deep Reinforcement Learning-based Text Mining

Deep Reinforcement Learning has been applied in the text mining applications, such as text generation, chatbots [279], and text-based games. Li et al. [165] combined the seq-to-seq LSTM-RNN model with a reinforcement learning (RL) framework to maximize the rewards of the seq-to-seq model by policy gradient. They provided the reward in three aspects: ease of answering, information flow, and semantic coherence. They used the original tuned LSTM-RNN model parameters as the initial parameters for the policy gradient of a DRL model. A delayed reward from the following sentence can feed back to the previous step to guide policy gradient training. The metrics were dialogue length, diversity of unigrams, and human evaluation. They found that the DRL model can produce more diverse and longer dialogues than the original LSTM-RNN model.

DRL-based sentiment analysis uses word-level sentiment to learn clause-level and sentence-level sentiments. Since each text level has its own utility, a hierarchical RL was used to train different goals step by step [242, 277], where RL was used to model these series of words. The DRL-based sentiment analysis can be focused on optimizing the accuracy of predicting sentiments or using sentiment to solve other problems. In the text-based games, Deshpande and Fleisig [55] discussed the problem of the long-term reward in sentiment analysis. They used LSTM to generate a current state and added an immediate sentiment reward to optimize a DQN model. Their sentiment system was trained in the BERT model
with a movie review dataset. They observed that the score of a game episode was improved by adding a sentiment-based immediate reward. In another Mario game, Krening et al. [146] used explanations of human advice to learn advice and warnings about the objects in the game. The agent needs to understand human advice by encouraging or avoiding an action by sentiment analysis. They evaluated the sentiment of the whole sentence as well as the clauses. RL maximizes the reward from object-focused actions and mitigates adverse actions. They used object-focused Q-learning to train the RL model so that the agent could earn more rewards from game episodes in their model.

The followings are some works to predict sentiment by the DRL model. Chen et al. [35] studied the sentiment of video clips by multimodal (e.g., text, audio, and video) sentiment analysis of each word and the fusion of different data sources. They used a Multimodal Corpus of Sentiment Intensity and Subjectivity Analysis (CMU-MOSI) dataset for sentiment classification, regression with texts, and visual and audio data representing sentiments. They had a controller of a gated multimodal embedding to control whether the input from each time point was useful or not. The state was the current embedding from multimodal contents, and the action kept this time input for sentiment prediction. The policy gradient algorithm REINFORCE was used to add a delayed reward from the prediction result of the LSTM model with a temporal attention layer. They compared the results to several baseline models and human evaluation by accuracy metrics. The results showed that adding an attention layer and multimodal fusion can improve sentiment prediction accuracy.

Zhang et al. [309] studied multimodal (i.e., texts and audio) sentiment analysis by analyzing the clause-level structure of an utterance. DRL was used in the clause structure prediction to divide a sentence into several clauses. The states were embeddings from texts and audio features, and the actions were whether the current time was the end of a clause. The sentence-level delayed reward had two components. One was the accuracy of predicting the true sentiment label with the current policy parameters at the end of a sentence. The other reward was how many clauses can be divided into a sentence by counting the number of ‘ending’ actions. The policy gradient algorithm maximized the rewards. This study used multimodal datasets CMU-MOSI and CMU-MOSEI, to train their LSTM modeling the sequence of text embedding, audio embedding, and the memory of text-audio input interaction. The authors compared their proposed model with several baseline models and demonstrated the outperformance of their model over other spoken language sentiment classification models.

Chen et al. [36] used three user reviews datasets to train a word-level sentiment analysis model in LSTM. They used the sentence-level sentiment as a reward to optimize the actor-critic parameters in a basic DRL model. The reward included the entropy loss and the ratio of matched sentiment words according to the sentence sentiment and the length of the sentence. However, the performance of the model in accuracy had room to be improved as they used the basic DRL model. Joseph [133] studied sentiment analysis for Amazon reviews using LSTM and DRL. The DL model used Q values and the proximal policy gradient model. The state was the current embedding of inputs, and the actions were ‘remove’ or ‘keep’ the current word embedding for the sentiment classifier. Following the action sequence of the
DRL model, the kept words can be sent to a Transformer-related sentiment classifier, such as pre-trained BERT, which can also provide a delayed reward of the policy network. When the parameters were optimized, they used the sentiment classifier to make new predictions. Hierarchical RL (HRL) has been applied to problems and sub-problems. Wang et al. [277] studied a document-level aspect sentiment classification problem that extracts several aspects of a user’s review ratings, such as the five-star ratings for a hotel’s location, value, room, and service. They had two policies in the LSTM model to reduce noises: aspect-related clause selection and sentiment-related word selection. A sentiment predictor can predict the document sentiment by concatenating the last hidden states of high and low-level LSTMs and providing a reward score from the known sentiment labels. The four aspect ratings were the reward signals from high- and low-level policies to guide the LSTM models. The word-level reward for one selected word trajectory had a delayed reward and a cost. The delayed reward was sentiment accuracy from the final sentiment predictor when all clauses and words were selected. The cost was the ratio of selected words to guide the DRL policy to reduce noisy words. They compared this proposed model with several baselines and individual selection RL models, finding that the HRL models performed best.

Unlike the works above having sentiment labels for supervised learning, we aim to identify the intents of news articles with no ground truth labels available. Hence, we develop a method with a small portion of labeled data to conduct our study in a large-scale dataset. In addition, we aim to identify an optimal action where an action goes through an uncertainty-aware decision process at each step. To the best of our knowledge, this work is the first that considers multidimensional uncertainty in the process of taking actions where the multi-dimensional uncertainty is estimated based on a belief model called Subjective Logic.

5.5 Proposed Approach: mudRIA

This section discusses our proposed intent analysis framework mudRIA with the following details. Firstly, we describe the processes of identifying fake news spreaders’ intent classes and intent-related text analysis features in this work. Secondly, we pre-train an LSTM intent classifier and fine-tune a DRL agent by REINFORCE to improve the intent prediction. Thirdly, we describe how to reduce the gradient variances in the policy gradient models by REINFORCE with baseline and policy-based A2C. Finally, we improve the learning process of those PG models with a multi-dimensional uncertainty-aware reward function (mUR).

5.5.1 Intent-related Text Features for Fake News Spreaders

Let us distinguish the concept of intent from similar terminologies, such as “goal” or “motivation”. The dictionary definition of “goal” is “an end state one aims to reach”. Motivation indicates the underlying force or influence that drives one to a particular action. The goal is usually specific, while the motivation is often a hidden cognitive feeling [17]. An intent is
CHAPTER 5. mudRIA: MULTIDIMENSIONAL UNCERTAINTY-Aware DEEP REINFORCEMENT LEARNING-BASED INTENT ANALYSIS OF FAKE NEWS SPREADERS

Figure 5.1: Intent classification of spreading fake news.

a purpose for action and can be analyzed to infer one’s future actions [214]. This work uses intent rather than goal or motivation to mean a more direct, immediate purpose for taking a specific action.

Intent Classification

Based on the surveys of intent classes of news spreading from the existing studies [8, 11, 12, 31, 122, 144, 214, 234, 250], we discuss OSN users’ primary intents in spreading fake news. Although dividing the intents into good, bad, and ugly groups by existing work [214], we realize it is intricate to determine if the intent is good or bad, such as political campaign and socialization. In social science research, the intents of news spreaders, regardless of real or fake news, have been studied based on conventional survey methodologies [144, 234]. However, different types of intents of fake news spreaders have not been investigated using data-driven approaches. Based on the popularity of the intents found by manual annotations in this work, we adopt and discuss the top five intent classes in the following:

- **Information sharing**: Online users may spread true news to users needing such knowledge. This intent includes sense-making or expertise sharing to facilitate the truth of shared information [234] or provide valuable domain knowledge [214]. However, unlike the good intents, due to a lack of fact-checking or knowledge, online users often share fake news [27] unintentionally.

- **Political campaign**: This intent uses fake news to create a false perception of an opposing party’s political figure or group to mislead public opinions to win elections [214].

- **Socialization**: Online users share information to attract more friends and stay connected in the OSNs [11]. They can expand their social cycles by sharing news, often finding common and interesting topics. Online users also share information for self-promotion [122] or self-disclosure [250] to expand their social cycles. Self-promotion means the focus of one user is more focused on himself in the social network so that users share their sound characteristics and hide their weaknesses [122]. The intent of self-promotion can bring fake news because the news can be biased or modified for
5.5. **Proposed Approach: mudRIA**

promotion. Self-disclosure is similar to self-promotion in sharing personal information but focuses more on building mutual understanding and increasing social bonds [250].

- **Emotion venting**: Online users can propagate fake news when they feel emotional happiness or disturbances from reading posts or experiencing good or bad events [8]. Venting positive or negative emotions, such as happiness, joy, anger, depression, and sadness, introduces negative reactions that can spread fake news. Emotion venting in social media often generates other fake news associated with the posting because one’s posting to vent subjective emotions is hard to be fact-checked.

- **Rumor propagation**: This intent can mislead users by sharing a rumor or unverified information [214]. Rumors can alter users’ emotions and attitudes toward certain events and increase uncertainty. The uncertainty often makes users judge without fact-checking and facilitates the massive propagation of fake news related to the rumors.

**Intent Class Annotation**

To identify fine-grained intents of spreading fake news, we first obtain the datasets of fake and true news and some datasets with news spreaders. Because all the datasets have the labels of fake, true, or partly true, as described in Section 5.6.1, only true and fake news are kept for intent mining. By analyzing the contents of posts or articles, we can classify the underlying intents of true or fake news spreaders.

In the current dataset, we call one piece of news data “a news piece” for consistency. First, we manually label each fake or true news piece with an intent class. Three annotators manually annotate each news piece to give one of the five intent classes discussed. Then, 33% of all news is annotated as a dataset $D$ of 1,500 tweets, with 835 fake and 665 true news. Finally, we assign the dominant intent class from the three sources to each news piece. Accordingly, this label is a gold intent class $y$ for a news piece $x$.

**Text Analysis Features**

We analyze the intents of true and fake news spreaders from lexical and semantic features. Figure 5.2 illustrates all the steps of intent mining. There are two popular feature sets provided by LIWC (Linguistic Inquiry and Word Count) [252] and SEANCE (Sentiment Analysis and Social Cognition Engine) [48], where they can capture various types of psychological, cognitive, and emotional features. The NLP tool LIWC can analyze the underlying features of given texts. We analyze the possible features for the five intents by the LIWC psychological processes, such as affective processes, cognitive processes, perceptual processes, and drives. SEANCE is a similar tool to LIWC, but this service is publicly available with no charges. SEANCE combines the word categories from multiple lexical sources, such as Harvard IV-4 in General Inquirer (GI), Lasswell, and Geneva affect label coder (GALC). Both tools generate component or category scores.
To extract more meaningful features related to the intent classes, rather than intuition and human knowledge, we run the correlation analysis for each intent class by one-against-all other classes. Then we pick the highly correlated features for each intent class defined in Section 5.5.1 and summarize those features in Table 5.1. Those lexical features for intent serve as intent lexical feature vectors in the intent classification models. We normalize the scores of each category to the range \([0, 1]\) using the min-max normalization technique.

### 5.5.2 REINFORCE-based Intent Classifier

DRL has been explored in NLP classification tasks to select key tokens of words and maintain their sequential relationship [311]. In sentiment classification, several studies [124, 277, 281, 309] have used DRL to reduce noisy tokens from the whole sentence to improve the accuracy of a prediction model. DRL enables learning a better structure representation of the entire fake news piece to enhance our intent classification task.

Our model learns a basic recurrent language model Long Short-Term Memory (LSTM), by pre-training. Then the fine-tuning step learns a DRL model, which appends a DRL agent
5.5. Proposed Approach: mudRIA

Figure 5.3: LSTM pre-training model. Rounded rectangular boxes represent the NN layers. An orange circle highlights the golden intent class, for a news piece example $x$.

to each LSTM cell to reduce the number of recurrent units from $k$. This section details the pre-training and fine-tuning steps of intent classifiers.

LSTM-based Pre-training

This language model has a recurrent structure of $k$ embedding layers ($\theta_{em}$) and $k$ LSTM cells ($\theta_{lstm}$) to process a sequence of $k$ input words in a news piece $x = \{x_1, \ldots, x_k\}$. The word embedding layer ($\theta_{em}$) determines how to convert the sequence of words into representations vectors. The embedding layer takes one token $x_t$ at time $t$ and generates a word vector $w_t$. The LSTM cell ($\theta_{lstm}$) is a recurrent structure that iterates $k$ times. At each time $t$, the LSTM cell takes a cell vector $c_{t-1}$, a hidden state vector $h_{t-1}$, and an input word vector $w_t$. The outputs for the next LSTM cell are $c_t$ and $h_t$. The final output of the LSTM cells is a state vector $h_k$, which is connected to the critic network ($\theta_{critic}$). This dense linear layer with ReLU and softmax activation functions provides an accuracy score to validate the intent prediction from the LSTM final state $h_k$, comparable to the critic from the RL domain. In the next sections, we borrow the term ‘critic’ to highlight and connect the role to DRL. Lastly, the critic network with parameters $\theta_{critic}$ generates a probability distribution of $P$ intent classes by a softmax layer.

Loss Function The pre-training updates a parameter set $\theta_{pre}$ of the three components in the intent classifier, where the pre-training aims to minimize the cross-entropy loss with the known gold intent labels. Considering the over-fitting prevention, the L2 regularization term $||\theta_{pre}||^2$ is added to this loss by:

$$L_{pre} = -\sum_{y=1}^{P} \hat{p}(y, x) \log p(y|x, \theta_{pre}) + \alpha||\theta_{pre}||^2,$$  \hspace{1cm} (5.1)
Fine-tuning by Policy Gradient-based DRL

This section discusses the sentence structure optimization steps by our PG-based DRL in Figure 5.4. Keeping the temporal relationship of the intent-related words can improve the intent prediction from the pre-trained LSTM intent classifier so that this fine-tuning step can optimize the number of pre-trained LSTM units.

The DRL model adds a word selection step to the LSTM classifier to form a time series of actions $\tau = \{a_1, \ldots, a_k\}$ from each ‘keeping’ or ‘masking’ decision of the actor-network. Then, the selected or non-masked words generate an optimized input sequence $x' = \{x_1, x_2, \ldots, x_{k'}\}$ of $k'$ words. Finally, this shorter $x'$ is processed by the three components of the LSTM intent classifier to predict the gold intent with higher accuracy. When the shorter optimized structure $x'$ by $k' \leq k$ was found correctly, a higher prediction of golden intent $y$ was expected in pre-trained LSTM, by $p(y|x', \theta_{Pre}) \geq p(y|x, \theta_{Pre})$. The core contribution of DRL for the NLP intent classification domain is to select intent-related words as the optimized
input sequence $x'$ in Figure 5.4C from a given news piece $x$ with a known intent $y$. Since the previous LSTM classifier serves as a fixed environment to support the state transitions, their parameters $\theta_{Pr}$ are frozen during the DRL learning of $\theta$.

Since the NLP problem differs from DRL’s traditional game-solving problems, we need to unify and clarify some essential terms used in the DRL domain. A training episode in DRL stands for one trial of the Markov decision process (MDP) by masking non-intent words from steps 1 to $k$ of a news piece and collecting the intent prediction accuracy rewards. In addition, an episode is applied based on only one news piece data $x$. We use five episodes for one news data as a mini-batch and collect loss for all the news in a batch, each running five episodes, to update $\theta$. After all the batches are processed in one training epoch, each news piece will try another five episodes in the following training epoch. When we describe our DRL model with an example of one specific episode, we suppose this episode is based on one specific news piece $x$, with an annotated golden intent class $y$. For example, we use the input sequence as “poll taken months ago found percent approval higher act collective bargaining law.” The intent label is class 2 as socialization. But the optimized input sequence $x'$ is unknown and be optimized from DRL algorithms.

**DRL Model** By setting $x' = x$, the DRL model extends an actor network to each LSTM cell $\theta_{lstm}$ in the previous LSTM intent classifier with the following details:

- **State**: At each step $t$, an input feature vector $w_t$ from the embedding layer $\theta_{em}$ represents the tokens of an input word $x_t$. The LSTM cell $\theta_{lstm}$ processes $w_t$ with $c_{t-1}$ and $h_{t-1}$ and passes $c_t$ and $h_t$. This hidden state vector $h_t$ serves as the state $s_t$ for the actor network at the current step as $s_t = \pi(w_t, c_{t-1}, h_{t-1}; \theta_{lstm})$.

- **Actor**: The actor network has one dense linear layer followed by a ReLU activation and another dense linear layer followed by softmax activation of two nodes. The parameters of all the neurons are in $\theta$ and can be tuned by the DRL. Given a state $s_t$ from one LSTM cell, the actor generates a policy $\pi(a_t|s_t, \theta)$ for two actions.

- **Action**: If an actor chooses $a_t = 1$, the ‘keeping’ action maintains $x_t$ in the optimized input sequence $x'$. Otherwise, if the actor chooses $a_t = 0$, the ‘masking’ action removes $x_t$ from $x'$.

- **State Transition**: The state transition determines how each action $a_t$ controls the state $s_{t+1}$ for the next step. Since $s_{t+1}$ is generated from the next LSTM cell, $a_t$ decides the inputs for the LSTM cell of step $t + 1$ by:

$$
    s_{t+1} = \begin{cases} 
        \pi(w_{t+1}, c_{t-1}, h_{t-1}; \theta_{lstm}) & \text{if } a_t = 0 \text{'masking'}; \\
        \pi(w_{t+1}, c_t, h_t; \theta_{lstm}) & \text{if } a_t = 1 \text{'keeping'}. 
    \end{cases}
$$

(5.2)
Delayed Reward  The intent prediction of the pre-trained LSTM network is improved by
masking non-intent words and keeping the temporal relationship of the intent-related words.
This goal can be reflected by a delayed reward from the intent class distribution of the
optimized input sequence $x'$ of one news piece, which is calculated by the LSTM classifier
after the last word $w_k'$. This delayed reward is unique in some NLP problems when
the standard immediate reward is unavailable. Three steps generate a delayed reward: (1) The
actor network makes $k$ decisions to generate a time-series of actions $\tau = \{a_1, \ldots, a_k\}$; (2) If
$a_t = 1$ exists in $\tau$, the corresponding input word vector $w_t$ will be added to the optimized
input sequence $x'$; and (3) The optimized sequence $x' = \{x_1, x_2, \ldots, x_k'\}$ is validated by the
pre-trained LSTM classifier in Section 5.5.2 to collect the delayed reward.

One delayed reward is the prediction of the true intent class label, as $R_{\text{pred}} = p(y|x', \theta_{P re})$.
Another reward is based on the number of masked words to select a small number of relevant
tokens in the DRL, as $R_{\text{len}} = (k - k')/k$. The total delayed reward is defined by:

$$R = R_{\text{pred}} + \lambda R_{\text{len}} = p(y|x', \theta_{P re}) + \lambda (k - k')/k,$$

where $\theta_{P re}$ is the parameter set for the LSTM intent classifier with a critic network, $y$ is the
golden label for a news piece $x$, $k'$ is the length of selected words in $x'$, and $\lambda$ is a weight.

Policy Gradient  Our DRL model maximizes the delayed reward in Eq. (5.3). Like many
other NLP problems where an immediate reward is unavailable after deciding an individual
step’s action, we will learn the actor’s policy by policy gradient (PG)-based models. Also,
the Q-learning models cannot calculate the Q values because of a lack of immediate reward.

The PG-related RL algorithms can directly tweak an established policy $\pi(a_t|s_t, \theta)$ of the
action probabilities. The big difference between PG and Q-learning is PG’s on-policy property.
The chosen action in a step must follow the current policy to tweak and slightly update the
current policy’s parameters through iterations. For example, the current stochastic policy
maximizes the expected delayed reward by calculating the gradients $R \log \pi(a_t|s_t, \theta)$ in RE-
INFORCE algorithm [288]. Based on the gradients for all $k$ transition steps in one episode
following PG’s on-policy property, REINFORCE calculates the negative log loss as:

$$\mathcal{L}_{PG} = -\sum_t R \log \pi(a_t|s_t, \theta),$$

5.5.3 Gradient Variance Reduction by Advantage Actor-Critic (A2C)

PG depends on the value of $R$ for each training episode, and the rewards ranges may cause
high variances of gradients. To avoid the influences from varying delayed rewards, the PG
models need to balance the variances of gradients $R \log \pi(a_t|s_t, \theta)$. We have several ways to
balance the gradients by deducting a value from $R$. One easy way is to use the baseline
function, the average value of $R$ from several mini-batch training episodes, to distinguish the
episodes with higher \( R \). The other way is to learn the value of the critic network. Because of the delayed reward in our intent classification task, we borrow the pre-trained critic network in Section 5.5.2 and apply a special policy-based Advantage Actor-Critic (A2C) model.

**Baseline Function**

As shown in Eq. (5.4), the identical \( R \) is multiplied at each step in an episode to calculate the gradients. There are five delayed rewards when we process one news piece by trying five mini-batch episodes. Our REINFORCE finds a baseline value \( b \) as the average of five episodes by the following loss function:

\[
\mathcal{L}_{PGB} = -\sum_{t} (R - b) \log \pi(a_t|s_t, \theta). \tag{5.5}
\]

When the average value balances the five episodes, the new rewards \( R - b \) may be positive or negative when all five episodes calculate the loss and gradients. This negative advantage over baseline can suppress the probability of choosing \( a_t \) in future episodes. It is also ubiquitous in all the models with balanced gradients, such as A2C, in the next section.

**Advantage by Pre-trained Critic Network**

The baseline value in Section 5.5.3 helps calculate the advantage over a reward. Furthermore, we can replace the baseline value with a critic value from a critic network to more accurately validate the new episodes’ rewards. The common A2C model predicts a value \( V(s_t) \) from a critic network for the current state \( s_t \). This critic value validates the discounted total rewards of the actor network’s current action \( a_t \). The Advantage is defined by the distance of the current critic value to the selected action’s discounted total reward. The A2C model aims to balance the variance of the discounted total rewards from different training episodes.

However, the delayed reward is an immediate reward at the last step. An Advantage value derived from LSTM’s critic network and the delayed reward is a special case. After investigating the existing LSTM classifier, we find that the previously defined critic network (\( \theta_{\text{critic}} \)) can serve this role. We can observe that the classification layer is connected to the LSTM cell after the last output \( h_k \) or \( h_k’ \) as the critic network to validate the quality of intent prediction of both LSTM cell outputs and the DRL actions \( \tau \). Each episode’s new reward is based on the optimized input sequence \( x’ \). When we try a new episode, the action \( a_t \) is determined by random choice following the current policy \( \pi(s_t, \theta) \). In this case, this policy exploration may choose the action with a lower probability. If the actor network exploits the policy at each step, the deterministic action will be selected to form \( \tau_\pi \), representing the current policy’s quality. The optimized input sequence of fully exploited policy is \( x’_\pi \) with \( k’_\pi \) words based on the number of ‘keeping’ actions in \( \tau_\pi \), which replaces the \( \tau \) in Figure 5.4. In the end, when the sequence \( x’_\pi \) is processed by the LSTM classifier, the critic network \( \theta_{\text{critic}} \)
can generate the quality of the current policy by a delayed reward $R_{\pi}$ from Eq. (5.3) as:

$$V(\tau_{\pi}) = R_{\pi} = p(y|x', \theta_{pre}) + \lambda(k - k')/k,$$

(5.6)

**Policy-based A2C**

Since the critic value of the current fully exploited policy $V(\tau_{\pi})$ is a delayed reward, the advantage of this reward from a new episode with policy exploration to the critic value $V(\tau_{\pi})$ is the same for all the steps in one episode, as $A = R - V(\tau_{\pi})$. The A2C model loss function is defined by:

$$L_{A2C} = -\sum_t [R - V(\tau_{\pi})] \log \pi(a_t | s_t, \theta).$$

(5.7)

When calculating the gradients and the loss, if the trial episode’s reward $R$ is very close to the critic value, the loss is close to 0, meaning the model converges and stops updating. When the advantage is positive, the trial episode’s reward of $R$ can improve the current policy. When updating a model with a positive advantage, the current policy will learn to increase the chosen action’s probability. When the advantage is negative, the trial episode fails to achieve the policy’s critic value, which will update the policy to decrease the probability of the chosen actions.

**5.5.4 PG with Multi-dimensional Uncertainty Reward Function**

The previous model variants have a constraint that the reward or advantage of the reward value in Eqs. (5.4), (5.5), and (5.7) are identical for all the steps in a single episode. The equal rewards indicate that the DRL agent trusts each selected action in $\tau$ equally because the agent does not know which step is more reliable to encode $x'$ at the final step. Our model leverages the pre-trained critic network $\theta_{critic}$ at each time $t$, as illustrated by Figure 5.5. In each local step $t$ of the DRL, the output hidden vector $h_t = s_t$ is passed directly to the critic network $\theta_{critic}$. This local critic network generates an intent class distribution $p(s_t, \theta_{critic})$, which can be represented by Subjective Logic [132] via vacuity maximization in Figure 5.5A. Owing to the uncertainty metrics in the SL-based opinion in Figure 5.5B, we can suggest a local certainty level $r_t$ to existing rewards $R$. In Figure 5.5C, the loss value at each step $t$ can consider a different amount of the delayed reward with certainty $r_t$, which provides an uncertainty-aware reward function (mUR) and allows the critic to influence the update of the actor’s model with $r_t R$ or $r_t A$.

**Critic Network at Local Steps**

In Figure 5.5A, the actor and critic are two parallel-running but independent networks, although they share the same input $s_t = h_t$. This local critic network can generate the
5.5. Proposed Approach: mudRIA

Figure 5.5: The uncertainty-aware reward function mUR for DRL models.

intent class distribution representing the policy \( \pi(a_t|s_t, \theta) \) at early steps. The critic network’s role is not to directly manipulate the actor’s decision. The critic only indirectly influences updating the actor’s policy and model parameters \( \theta \) by a reward function. In addition, the critic network’s parameters \( \theta_{\text{critic}} \) are pre-trained in Section 5.5.2 and kept frozen in this step. However, the actor network’s parameters \( \theta \) are learned and updated by trying new episodes for fake news data.

Uncertainties from Subjective Logic Opinion

These local intent probabilities \( \pi(a_t|s_t, \theta) \) can be considered a multinomial opinion in SL corresponding to five beliefs towards the intent classes. The key benefit of using an SL opinion is to offer ways to estimate several uncertainty estimates, such as *vacuity* caused by a lack of evidence and *dissonance* introduced by conflicting evidence. Since the local intent probabilities cannot reflect the level of vacuity, we apply the vacuity maximization technique [132] on the local probabilities of \( P \) classes and generate a vacuity maximized opinion \( \omega_t = [b_t, u_t, g] \). In Figure 5.5B, there are \( P \) belief masses in SL opinion as \( b_t \). \( u_t = [u^{\text{vac}}_t, u^{\text{diss}}_t] \) refers to two considered uncertainty metrics. The base rates in vector \( g \) are the distribution of \( P \) classes in annotated set \( D \). The uncertainty quantification in this work is primarily epistemic uncertainty, which is related to a situation that we cannot predict exactly an event because of a lack of knowledge [94]. There is also aleatoric uncertainty related to statistical uncertainty from long-time observations. Aleatoric uncertainty is due to randomness, and reducing the statistical uncertainty has a role in accuracy or quality.
evaluations. However, the nature of uncertainty determines the interweaving of two types so that they are not easily separated in the SL subjective opinion.

**Multi-dimensional Uncertainty-Aware Reward Function (mUR)**

In one DRL training episode, each step has its own uncertainty metrics $u_t$ from a local critic-transformed SL opinion $\omega_t$. Uncertainties in $u_t = [u_t^{vac}, u_t^{diss}]$ decide an overall uncertainty level by accessing vacuity and dissonance sequentially, with a threshold $\eta$. This certainty/uncertainty level decision process is because dissonance plays more roles in decision-making in the SL opinion with a high vacuity value. In the reward function $r_t R$, we discuss the three steps to decide a certainty level $r_t$ as follows:

1. **High certainty level**: The $u_t$ has a low uncertainty level simply because vacuity is lower by $u_t^{vac} < \eta$. The critic-derived SL opinion indicates a high certainty level toward the actor by $r_t = 1 + \eta - u_t^{vac}$. It is bounded by $r_t \in (1, 1 + \eta]$, which weighs more to the delayed reward for the steps with a higher certainty level.

2. **High certainty level**: The $u_t$ can still have a low uncertainty level if the first step fails, but the dissonance is lower, by $u_t^{diss} < \eta$ and $u_t^{vac} \geq \eta$. Therefore, a high certainty level is calculated as $r_t = 1 + \eta - u_t^{diss}$ and is bounded by $r_t \in (1, 1 + \eta]$.

3. **Low certainty level**: The $u_t$ has a high uncertainty level when both steps are not met by $u_t^{vac} \geq \eta$ and $u_t^{diss} \geq \eta$. When under the conditions of a high $u_t$, belief masses $b_t$ are nearly uniform in each class. There is no satisfying level of certainty of the opinion generated from the vector $s_t$. Thus, the critic-derived SL opinion provides a low certainty by $r_t = 1$ without modifying the delayed reward for step $t$ as $r_t R = R$.

The above rules apply to REINFORCE with baseline and A2C, as $r_t A \geq A$. For the positive advantage values when $A > 0$, $r_t$ is defined as in the above steps. However, the advantage can be negative, requiring $r_t \leq 1$ to satisfy $r_t A \geq A$. This case means a high certainty step receiving a negative $A$ will trust this negative $A$ less to help the model update correctly. Thus, we modify the previously defined $r_t$ to the range of $[1 - \eta, 1 + \eta]$. If $A < 0$, the desired $r_t$ range is $[1 - \eta, 1)$ when $r_t = 1 - \eta + u_t^{vac}$ in step 1 or $r_t = 1 - \eta + u_t^{diss}$ in step 2.

**Loss Function**

Considering the uncertainty-based reward function as a belief of each step’s action, the previous loss functions of the REINFORCE, REINFORCE with baseline, and A2C models are updated by:

$$ L_{mUR} = -\sum_t r_t R \log \pi(a_t|s_t, \theta), \quad \text{or} \quad L_{mUR} = -\sum_t r_t A \log \pi(a_t|s_t, \theta), \quad (5.8) $$
Table 5.2: Model Hyper-Parameters Setting

<table>
<thead>
<tr>
<th>Model</th>
<th>Parameter</th>
<th>Value</th>
<th>Parameter</th>
<th>Value</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSTM/LSTM+TF</td>
<td>Training epochs</td>
<td>30</td>
<td>L2 regularization</td>
<td>1e-4</td>
<td>$h_t$ and $c_t$ size</td>
<td>128, 128</td>
</tr>
<tr>
<td></td>
<td>Batch size</td>
<td>16</td>
<td>Dropout rate</td>
<td>0.25</td>
<td>LSTM dense layer dims</td>
<td>[128, 257, 5]</td>
</tr>
<tr>
<td></td>
<td>Padding length $k$</td>
<td>20</td>
<td>Learning rate</td>
<td>1e-4</td>
<td>Text mining tool features</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td>Embedding dims</td>
<td>128</td>
<td>Dictionary size</td>
<td>10000</td>
<td>LSTM+TF dense layer dims</td>
<td>[153, 257, 5]</td>
</tr>
<tr>
<td>PG, PGB, A2C</td>
<td>Training epochs</td>
<td>100</td>
<td>Learning rate</td>
<td>0.02</td>
<td>DRL actor dense layers dims</td>
<td>[128, 257, 2]</td>
</tr>
<tr>
<td></td>
<td>$R^{cen}$ weight $\lambda$</td>
<td>0.4</td>
<td>Batch size</td>
<td>32</td>
<td>Mini-batch episodes</td>
<td>5</td>
</tr>
<tr>
<td>mUR</td>
<td>Training episodes</td>
<td>100</td>
<td>Training episodes</td>
<td>600 K</td>
<td>Uncertainty threshold $\eta$</td>
<td>0.3, 0.5, 0.6</td>
</tr>
</tbody>
</table>

where $r_t R$ is for mUR with REINFORCE, $A = R - b$ is for mUR with REINFORCE with baseline, and $A = R - V(\tau_\pi)$ is for mUR with A2C model. If $r_t = 1$ at each step, the loss function of the uncertainty-aware reward function is the same as in the previous models.

### 5.6 Experimental Setup

This section describes datasets, metrics, and comparing schemes used for our experiments.

#### 5.6.1 Dataset

We use the publicly available dataset *LIAR 2015* [282]. This dataset has 12.8K short statements by political parties with labels from a fact-checking agency, where we only keep the labels of 2,511 fake and 2,073 true news. We annotate each piece of news with one intent class based on the five intent classes discussed in Section 5.5.1. We only annotate 33% of fake and true news in LIAR 2015 to form a dataset $\mathcal{D}$ of 1,500 tweets to pre-train the LSTM intent classifier and fine-tune the proposed DRL models in Figure 5.4. There are 835 fake news and 665 true news. The training dataset $\mathcal{D}$ relies on three annotators’ manual annotations of each news data. Since each news collected labels from three sources, we assign the dominant intent class to each piece of news data, both fake and true. We separate dataset $\mathcal{D}$ into fake news only or true news only.

#### 5.6.2 Metrics

The performance of the DRL models in the mudRIA framework can be evaluated by:

- **Accuracy**: This multi-class classification accuracy is estimated based on the ratio of correctly predicted data over the total testing data. It measures correct detection for true positives (TP) and true negatives (TN) out of all tried cases and is given by:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}, \quad (5.9)$$

where FP refers to false positives and FN is false negatives.
• **Prediction Rate**: This metric is the effectiveness of DRL and is computed by the prediction accuracy of the gold intent class from the critic network.

• **The Optimal Length of Words**: This indicates the efficiency for DRL and is obtained by the length of the optimized sequence.

• **The Reward of Episodes**: The average delayed reward across the training episodes.

### 5.6.3 Comparing Schemes

Our proposed models aim to predict the intent classes for fake news datasets by DRL algorithms. The pre-trained LSTM intent classifier is a non-DRL baseline model. All the proposed fine-tuned DRL models are built on the pre-trained LSTM framework with frozen parameters \( \theta_{pre} \). The intent-related linguistic features are added to the LSTM model as an LSTM classifier with lexical features. Each DRL algorithm only fine-tunes the actor network’s parameters \( \theta \) by new loss functions. The REINFORCE model is the basic PG model for comparison. Built from this basic PG model, we compare the REINFORCE with baseline and A2C. Then we add the multidimensional uncertainty-aware reward function to each PG model and check the performance of this new reward function. Those models’ design features of rewards and loss functions can be evaluated as ablation studies. We use the following names when discussing the comparative performance results in Section 5.7: LSTM, LSTM+TF, PG, PGB, A2C, PG+mUR, PGB+mUR, and A2C+mUR. Table 5.2 summarizes the model hyper-parameters and their default values used in this work.

### 5.7 Experimental Results & Analyses

In each epoch of training LSTM and DRL models, we separate the dataset to 80\% as a training set and 20\% as a testing set. In the testing episodes, the actor always chooses the optimal action based on each step’s policy \( \pi(a_t|s_t, \theta) \).

#### 5.7.1 Multi-Class Intent Prediction Accuracy

Table 5.3 compares the accuracy between the non-RL LSTM and non-RL LSTM with text features and six DRL-based models. The weight \( \lambda = 0.4 \) in Eq. (5.3) is described in Table 5.2, and the uncertainty threshold \( \eta \) uses the best settings for each mUR model. From all news data, DRL-related models bring a higher multi-class accuracy than LSTM. However, the DRL models only increase the prediction accuracy for intent class 1 ‘information sharing’ and class 2 ‘political campaign’. The baseline in PGB shows the same accuracy level as PG, but A2C improved 1\% from PG. The mUR function increases the accuracy from PG’s 90.3\% to 91\%, from PGB’s 90.3\% to 92.7\%, and A2C’s 91.3\% to 93.3\%. Overall, the best results
5.7. EXPERIMENTAL RESULTS & ANALYSES

Table 5.3: Multi-Class Intent Testing Accuracy

<table>
<thead>
<tr>
<th>Model</th>
<th>All Testing Data (each class)</th>
<th>Fake News Data (each class)</th>
<th>True News Data (each class)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Model</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LSTM</td>
<td>0.86</td>
<td>0.817</td>
<td>0.925</td>
</tr>
<tr>
<td></td>
<td>(0.772, 0.942, 0.874, 0.943, 0.8)</td>
<td>(0.676, 0.934, 0.864, 1.0, 0.667)</td>
<td>(0.915, 0.955, 0.889, 0.857, 1.0)</td>
</tr>
<tr>
<td>LSTM-TF</td>
<td>0.893</td>
<td>0.889</td>
<td>0.9</td>
</tr>
<tr>
<td></td>
<td>(0.974, 0.952, 0.846, 0.63, 0.557)</td>
<td>(0.956, 0.951, 0.818, 0.765, 0.5)</td>
<td>(1.0, 0.955, 0.889, 0.429, 0.643)</td>
</tr>
<tr>
<td>PG</td>
<td>0.903</td>
<td>0.894</td>
<td>0.917</td>
</tr>
<tr>
<td>$\lambda = 0.4$</td>
<td>(0.909, 0.99, 0.846, 0.815, 0.714)</td>
<td>(0.882, 0.983, 0.818, 0.882, 0.667)</td>
<td>(0.949, 1.0, 0.889, 0.714, 0.786)</td>
</tr>
<tr>
<td>PGB</td>
<td>0.903</td>
<td>0.906</td>
<td>0.9</td>
</tr>
<tr>
<td>$\lambda = 0.4$</td>
<td>(0.936, 0.99, 0.874, 0.687, 0.686)</td>
<td>(0.926, 0.984, 0.864, 0.765, 0.667)</td>
<td>(0.949, 1.0, 0.889, 0.571, 0.714)</td>
</tr>
<tr>
<td>A2C</td>
<td>0.913</td>
<td>0.922</td>
<td>0.9</td>
</tr>
<tr>
<td>$\lambda = 0.4$</td>
<td>(0.944, 0.99, 0.874, 0.758, 0.686)</td>
<td>(0.941, 0.984, 0.864, 0.882, 0.667)</td>
<td>(0.949, 1.0, 0.889, 0.571, 0.714)</td>
</tr>
<tr>
<td>PG+mUR</td>
<td>0.91</td>
<td>0.906</td>
<td>0.917</td>
</tr>
<tr>
<td>$\eta = 0.3$</td>
<td>(0.927, 0.99, 0.819, 0.758, 0.793)</td>
<td>(0.912, 0.984, 0.773, 0.882, 0.75)</td>
<td>(0.949, 1.0, 0.889, 0.571, 0.857)</td>
</tr>
<tr>
<td>PGB+mUR</td>
<td>0.927</td>
<td>0.933</td>
<td>0.917</td>
</tr>
<tr>
<td>$\eta = 0.5$</td>
<td>(0.944, 0.99, 0.923, 0.758, 0.764)</td>
<td>(0.941, 0.984, 0.864, 0.882, 0.667)</td>
<td>(0.949, 1.0, 0.944, 0.571, 0.786)</td>
</tr>
<tr>
<td>A2C+mUR</td>
<td>0.933</td>
<td>0.928</td>
<td>0.966</td>
</tr>
<tr>
<td>$\eta = 0.6$</td>
<td>(0.942, 0.99, 0.923, 0.815, 0.793)</td>
<td>(0.926, 0.983, 0.909, 0.882, 0.75)</td>
<td>(0.966, 1.0, 0.944, 0.714, 0.857)</td>
</tr>
</tbody>
</table>

from A2C+mUR have a 3% increase in the intent classification, improving both fake and true news spreaders' intent prediction. All the DRL models consistently learn to predict the fake news in testing datasets with better accuracy than the pre-trained LSTM. However, DRL models, except A2C+mUR, fail to predict true news data. Our mUR function helps A2C to fit the true news data better with a great improvement of 6.6% from A2C and 4.1% from LSTM.

5.7.2 Reward based on Model Effectiveness and Efficiency

Besides the accuracy metric for the intent classifications, DRL models also learn to reach a higher reward through iterations. Figure 5.6 shows the total delayed reward and the efficiency component by the length of words in the optimized sequence. We keep weight $\lambda = 0.4$ as a constant value to compare $R_s$ in the same scale in Eq. (5.3). The length of optimized words is captured when the model reaches the highest accuracy with the minimum number of kept words in $x'$. Figure 5.6b shows the length reduction from LSTM’s $k = 20$ to DRL’s $k'$ from training episodes and testing. From the total testing data, the total reward increases when the more advanced models are used for training and testing episodes in Figure 5.6a.
CHAPTER 5. mudRIA: Multidimensional Uncertainty-Aware Deep Reinforcement Learning-based Intent Analysis of Fake News Spreaders

This performance trend is A2C>PGB>PG. In addition, our new reward function, mUR, leads to a large increase in the achieved reward via training. The length of words is part of the reward function; accordingly, using shorter words leads to a higher reward. The masked length reward in Eq. (5.6) is DRL’s main contribution in reducing the number of noisy words in the news tweets. As shown in Figure 5.6b, the length shows the same trend as the total delayed reward, and all six models reduce the length from PG’s 9.9 to A2C+mUR’s 8.2 by 1.7 words for each testing data. Adding a baseline in PGB and an uncertainty-aware function in PG+mUR, PGB+mUR, and A2C+mUR found fewer optimized words. In conclusion, the multi-dimensional uncertainty-aware reward function efficiently reduces the noisy words in predicting intent. A2C+mUR predicts true news in testing data better than fake news, whereas all other models predict fake news with higher rewards.

5.7.3 Sensitivity of Varying Uncertainty Threshold, $\eta$

To compare against each DRL model, we choose the best $\eta$ in our mUR models because model-specific $\eta$ demonstrates a higher accuracy with a reduced number of optimized words. To further investigate if certain ranges of SL-based certainty contribute more to the delayed reward by $r_t R$ or $r_t A$, we analyze the sensitivity of various uncertainty thresholds for PG+mUR, PGB+mUR, and A2C+mUR in Figures 5.7, 5.8 and 5.9. Those three figures demonstrate the effects of $\eta$ on the accuracy and length of optimized sequences. In partic-
5.7. Experimental Results & Analyses

Figure 5.7: The effects of varying threshold $\eta$ in PG and PG+mUR on the prediction accuracy and the length of optimized words (PG is $\eta = 0$).

ular, when $\eta = 0$, the uncertainty is always high, resulting in $r_t R = R$ for PG and $r_t A = A$ for PGB and A2C. In general, when $\eta < 0.6$, this uncertainty-aware reward function can support finding a shorter length than the non-uncertainty models. However, removing more words does not always guarantee a higher accuracy. This is because the model may remove meaningful or intent-related words. In addition, the model may reach a convergence state of a high reward of short length and low accuracy.

Fig 5.7 suggests that if $r_t$ is applied on $R$ rather than a balanced $R$, a certain range of $\eta$ effectively improves accuracy. For PG+mUR, $\eta = 0.3$ has higher accuracy of 0.91 than PG, and fake news has higher accuracy than true news. Since using $\eta = 0.4$ reveals the same accuracy as PG with a reduced length, $\eta \in [0.3, 0.4]$ is an effective range for PG+mUR. In addition, true news can be predicted with higher accuracy within this effective range than fake news. The parameter value $\eta = 0.3$ demonstrates the best fake and true news accuracy. An optimal threshold of certainty level $r_t \in [1, 1 + \eta]$ follows our expectation for $r_t R$’s role because our design aims to add variances to $R$ for each step’s gradients and pushes the policy to mask more words. If more words are masked based on a higher $\eta$ value in Figure 5.7, the optimized news piece is expected to have a better intent accuracy. However, if the variance caused by $r_t R$ is too large, it will cause a large gradient variance problem for the policy gradient models. They also show in this accuracy plot, even if the policy masks more words for $\eta > 0.4$, those masking decisions are not always correct and cause a reduction of accuracy when the variance of $r_t R$ is large.

If $r_t$ is applied on a balanced $R$, such as for PGB and A2C, the role of $\eta$ is different from PG. Overall, mUR is more effective in improving the total delayed reward and reducing the number of optimized words than in PG+mUR because improvements are found for all the testing $\eta$ values for the whole testing data. Removing more words by mUR can still train the model with higher accuracy for PGB+mUR and A2C+mUR. In Figure 5.8, $\eta = 0.5$
achieves the best accuracy for total testing and fake news data. In Figure 5.9, the accuracy for total testing data still increases for $\eta = 0.1 - 0.6$. However, A2C+mUR shows a unique trend that the true news spreaders’ intent is predicted with a large hop after $\eta = 0.4$. When $\eta = 0.5$, the true news and fake news have the same accuracy, and the true news continues to learn well and reaches a higher accuracy than the fake news when $\eta = 0.6$. In summary, our proposed reward function with $\eta$ for predicting fake news intents suggests the promising role of SL-based certainty.
5.8 Key Findings

In this work, we proposed a multidimensional uncertainty-aware deep reinforcement learning-based intent analysis framework, named mudRIA, to analyze the intent of fake news. We applied several DRL policy gradient models to conduct an intent classification tale for a given set of fake news based on real datasets. We formulated a delayed reward in the DRL models to achieve the model’s effectiveness and efficiency in terms of prediction accuracy and the number of words used in the optimized sequence. To enhance the limitations of the delayed reward, we devised a local response using multi-dimensional uncertainty estimated by a belief model called Subjective Logic. This design allowed the agents to trust delayed rewards early and led to the DRL model’s prediction accuracy being higher. We also used reliable datasets labeled by three human annotators for the ground truth intent labels. We annotated the intent class for the fake and true news from the dataset LIAR15. We used this annotated dataset to train the LSTM intent classifier and then created DRL models to help reduce the noisy words.

Via extensive experiments using a real dataset, we had the following findings:

- The DRL models with optimized structure representation significantly improved the multi-class classification accuracy from the pre-trained LSTM intent classifier.
- Our uncertainty-aware reward mUR, with two uncertainty metrics, vacuity, and dissonance could help the DRL models achieve higher gold intent class classification accuracy while reducing the number of noisy words.
- The A2C with multi-dimensional uncertainty reward combined the strengths of gradient reduction and a local certainty level to accomplish our intent classification goals.
- The conducted sensitivity analysis demonstrated that PGB+mUR and A2C+mUR outperformed other counterparts when using an optimal uncertainty threshold, $\eta$, to predict fake news spreaders’ intent.

In future work, we plan to conduct the following tasks: (1) dig into more relevant metrics for the rewards by effectiveness and efficiency; (2) validate our models and the multi-dimensional uncertainty reward function from other fake news datasets; and (3) analyze more aspects on how to apply our DRL intent classifier to identify fake news spreaders and true news spreaders to minimize influences of fake news spreaders in online social networks.
Chapter 6

Conclusions

In this chapter, I summarize the completed research items and the generated publications during my Ph.D. studies by the final defense exams.

6.1 Summary of the Key Contributions & Findings

In summary, we made the following contributions and findings in my dissertation research:

- We conducted extensive literature review to investigate what types of online social deception attacks exist and what countermeasure approaches have been examined in the literature.

- We built social capital (SC)-based friend recommendation framework to combat phishing attacks in online social network platforms and proved its effectiveness in defending against phishing attacks via real dataset-based, extensive experiments.

- We found that users having friends with high SC can self-defend against phishing attacks better than users having friends with the same topic interests or social attributes.

- We identified that SC-based FRSs could enable users to combat phishing attacks better than non-SC-based counterparts because the friends of the users with high social capital can help them defend against phishing attacks.

- We obtained the result that Bot-based phishing attacks can be more easily detected and defended than human-based phishing attacks under all FRSs because bot attackers show more distinctive characteristics than human attackers in social capital.

- We developed two SL-based opinion models, including uncertainty-based or homophily-based opinion models (OMs), and refined three exiting baseline OMs to be compared against the proposed two opinion models. All five OMs are devised to represent users’ uncertain opinions based on Subjective Logic to deal with uncertain opinions explicitly. By modeling real users’ attributes and behaviors, we demonstrated the impacts of divergent and convergent opinion models to investigate how a different OM can contribute to combating disinformation differently in OSNs.
• We found that uncertainty-based OM can assist users in excluding false, contradicting, and uncertain information and to believe and accept true information. However, other OMs based on homophily, assertion, and encounter can easily mislead users to believe false information.

• We observed that a user’s game-theoretic process of disinformation propagated in a network can significantly affect the dynamics of the network in terms of network topology and network influence, which is represented by communities and polarization. Compared to all other OMs, homophily-based OM can cause the highest polarization in the network, while uncertainty-based OM can help users access true information best.

• We designed the intent analysis framework of fake news spreaders based on a multidimensional uncertainty-based deep reinforcement learning technique. The proposed framework is the first that is designed to minimize the number of words used to obtain an intent classification while maximizing the accuracy of the intent classification.

• We reduced gradient variances in text classification policy gradient DRL by A2C with a multidimensional uncertainty reward function from a pre-trained critic network. We combined the strengths of gradient reduction and a local certainty level to the delayed reward.

6.2 Publications Accepted, Published, Submitted, and In Preparation

During my Ph.D. studies, I have the following papers published (P), accepted (A), submitted (S), revised (R), archived (C), and in preparation (PR):


6.3 Future Research

I plan to conduct future research based on the following directions:

- Validate our mudRIA models and the multi-dimensional uncertainty reward function from more fake news datasets; Replace the recurrent unit with the Transformers unit for better performance from the pre-trained language models; Analyze more aspects on how to apply our DRL intent classifier to minimize influences of fake news spreaders in online social networks.

- Study the intent and tactics of cybergroomers performing cybergrooming (i.e., attackers aiming to achieve online sexual exploitation via online social network platforms) attacks to adolescents; Build a robust chatbot to protect youth victims in the education
programs; Conduct thorough evaluations including human evaluation of the developed chatbots.

- Analyze the intent and tactics of crowdturfing (i.e., malicious, paid human workers performing malicious behaviors to achieve their employers’ goals) attackers.

- Conduct uncertainty quantification research by SL focusing on the role of accuracy or quality evaluations. This statistical role of uncertainty belongs to aleatoric uncertainty due to randomness. The completed work mainly focused on epistemic uncertainty, where collecting more evidence can update the SL opinion.

- Explore bot detections based on our SC quantification and findings. Study the influences of personality traits and demographic features on social capital quantification and the behaviors of legitimate users and human attackers types.

- Develop a social capital-based defense mechanism for users to combat false information and employ their online social capital for fast and robust crisis recovery; Discuss the different roles of three dimensions of social capital, including structural, cognitive, and relational subtypes, in the crisis event recovery.

- Address the effectiveness of using social capital in terms of recommending ‘useful friends’, emphasizing resourcefulness, which is well-aligned with the core concept of social capital.

- Conduct case studies in multiple OSN platforms for applicability. Collect event-related datasets to predict users’ opinion scales from real situations. These real datasets can demonstrate network characteristics.
Bibliography


[156] M. Latonero, “Human trafficking online: The role of social networking sites and online classifieds,” Available at SSRN 2045851, 2011.


Appendices
Appendix A

Supplement Document: Mitigating Influence of Disinformation Propagation Using Uncertainty-Based Opinion Interactions

A.1 Nash Equilibria Analyses

The solution of a non-cooperative game is called a Nash equilibrium (NE), where each player has the same knowledge of the opponents’ strategies and chooses the best response according to the knowledge of all the opponents [249]. In our opinion model game, we can also analyze attackers, users, and a defender’s NE solutions, which are the best strategies based on the accurate beliefs of the opponent’s strategies. This game is a game of incomplete information because each player needs to estimate the types of opponents as the prior assumption of nature selecting a player type. For instance, the user needs to guess the type of the opponent as either a legitimate user or an attacker because different types of players have different strategies, preferences, and payoff functions. The defender also needs to decide if the reported user is an attacker or a legitimate user. NE solutions are analyzed from a game tree and a corresponding normal-form game matrix to calculate the payoffs of all possible choices of both players. However, the payoffs between a pair of players are highly dependent on the current opinions of the players in Eqs. (4.14), (4.17), and (4.20) so that NE solutions are not fixed for all cases of opinion interactions.

<table>
<thead>
<tr>
<th>Attacker</th>
<th>SU</th>
<th>U</th>
<th>NU</th>
</tr>
</thead>
<tbody>
<tr>
<td>DG</td>
<td>$-0.67, 0.11$</td>
<td>$-0.67, 0.11$</td>
<td>$-0.67, 0$</td>
</tr>
<tr>
<td>C</td>
<td>$-0.84, 0.13$</td>
<td>$-0.84, 0.13$</td>
<td>$-0.85, 0$</td>
</tr>
<tr>
<td>DN</td>
<td>$-0.48, 0.12$</td>
<td>$-0.48, 0.12$</td>
<td>$-0.49, 0$</td>
</tr>
<tr>
<td>S</td>
<td>$-0.35, 0.14$</td>
<td>$-0.35, 0.14$</td>
<td>$-0.36, 0$</td>
</tr>
</tbody>
</table>

In a particular interaction of players in the game framework, one player knows the opponent’s opinion, strategies, and payoffs and makes decisions by considering the payoffs of all players.
A.1. Nash Equilibria Analyses

Figure A.1: A normal-form game tree of an attacker with an opinion \((b = 0.11, d = 0.72, u = 0.17, a = 0.13)\) spreading disinformation to a user with opinion \((b = 0.09, d = 0.25, u = 0.66, a = 0.27)\). The attacker and user make simultaneous moves denoted by the dotted lines. The attacker’s strategies are \(a^A_1\)s, and the user’s ones are \(a^U_1\)s, respectively. The two utility values under each tree node are the attacker’s utility in Eq. (4.15) and the user’s utility in Eq. (4.21) without using \(P^A_j\).

to reach the Nash equilibrium. The decisions by Nash Equilibrium (NE) are demonstrated in Tables I, II, and III, where the highest payoffs for both the row player (underlines) and column player (overlines) are considered. These decisions are different from our proposed game model and the payoffs proposed in Eqs. (14), (17), and (20) because the NE solutions are based on the assumption that the players have correct beliefs, which is unrealistic and thus not reflected in our approach. The normal-form matrix game for attackers in Table I considers the strategies of the user or the defender. The attacker’s payoff of strategy \(k\) and the user’s strategy \(m\) is

\[ EP^{A_i}_{k,m} = \sum_{\ell \in D} p^D \cdot u_{k\ell m} \]

The attacker’s payoff of strategy \(k\) and the defender’s strategy \(\ell\) is

\[ EP^{A_i}_{k,\ell} = \sum_{m \in U} p^U \cdot u_{k\ell m} \]

The \(u_{ij\ell m}\) is defined by Eq. (15) as defined previously, and this \(u_{ij\ell m}\) depends on the specific opinion values of the attacker and the user. In Table I, the Nash equilibrium solutions are \(\{S, SU\}\), \(\{S, U\}\), and \(\{S, M\}\).

The normal-form matrix game for the row player (i.e., users) in Table II considers the role and strategy of the column player (i.e., either an attacker or a user), which is determined by Nature with probability \(P^A_i\), as shown in Figure 2. When Nature decides the column player’s type, the column player’s strategies are known by the row player, so the utility values on the bottom lines of Figure 2 are calculated under each condition of both players’ strategies. On the left half, when the column player is an attacker, the utility is \(un_{m,k} = -s(m, \omega_F, \omega_i, \omega_j)\) (as a part of Eq. (21)) if the row user decides to update the current opinion \((m = a^U_1\) or \(a^U_2\)). If the row user refuses to update, the utility is \(un_{m,k} = 0\). On the right half, when the column player is a user, the utility is \(un_{m,m'} = uc^U_j\) (as a part of Eq.(22)) if the column player is U-type. For column player \(j\) in other types, the utility is \(un_{m,m'} = hc^U_j\). In the normal-form matrix game in Table II, the row player user’s payoff of strategy \(m\) and column player’s
strategy \( k, m' \) combines the probability of interacting an attacker by \( P^A_U \) and interacting a user by \( 1 - P^A_U \), as \( EPN^U_{m,k,m'} = P^A_U \cdot un_{m,k} + (1 - P^A_U) \cdot un_{m,m'} \). When the row and column players can observe all the payoffs in Table II, the row player chooses the highest payoff value, as the first payoff in each cell, highlighted by an underline, in each column. The column player chooses the highest payoff value, as the second payoff in each cell, highlighted by an overline, in each row. The cells selected by both the row and column players are the NE solutions, which are four solutions in Table II, as \{SU, (S, SU)\}, \{SU, (S, U)\}, \{U, (S, SU)\}, and \{U, (S, U)\}.

The normal-form matrix game for the row player defender in Table III also considers the role of the column player as either an attacker or a user. The defender makes a decision after an interaction when a user is reported by other users at least three times. Similarly, Nature decides if the reported user is an attacker or a normal user by \( P^A_U \). In the left half of Figure 3, the defender’s utility toward an attacker’s strategy \( k \) is \( un_{\ell,k} = u^D_{\ell,k} \) and \( u^D_{\ell,k} \) is defined in Eq. (18). In the right half of Figure 3, if a user chooses \( NU \), the defender’s utility is \( un_{\ell,m=\ell k} = 0 \). If a user decides to update opinion, the defender will consider all the attacker’s strategies to compute the utility towards a user, as \( un_{\ell,m \neq \ell k} = \sum_{k \in A} p^A_k \cdot u^D_{\ell,k} \). In the normal-form matrix game of Table III, the defender’s payoffs in each cell combines the utilities of interacting an attacker or a user, as \( EPN^D_{\ell,k,m} = P^A_U \cdot un_{\ell,k} + (1 - P^A_U) \cdot un_{\ell,m} \). In Table III, when the row and column players choose their best strategies, the NE solutions are \( \{T, (S, SU)\} \) and \( \{T, (S, U)\} \).

Figure A.1 demonstrates the game tree and utilities of the normal-form game as an example of an attacker \( \{b = 0.11, d = 0.72, u = 0.17, a = 0.13\} \) attacking a U-type user \( \{b = 0.86, d = 0.11, u = 0.03, a = 0.89\} \). When an attacker selects a user, he can observe the user’s strategies and payoffs. In this game tree, an attacker and a user make simultaneous moves, the attacker chooses one of the four strategies, and the user chooses one of the three strategies. Table A.1 presents all possible interactions of an attacker and a user. The two payoffs in one cell are the utility payoff of a row player’s and column player’s strategy. The upper and lower bars indicate that this cell is optimal for the column or row player so that the NE solutions of \( (S, SU) \) and \( (S, U) \) are selected by meeting both row and column optimal. Under this interaction case, the NE solution of an attacker is the deception strategy, Subversion, while the best response of the U-type user is to update the opinion by either \( SU \) or \( U \).

Figure A.2 provides an example of the game tree and the payoffs of a U-type user \( \{b =
Figure A.2: A normal-form game tree of a user with an opinion ($b = 0.48, d = 0.24, u = 0.28, a = 0.67$) interacting with a column player with an opinion ($b = 0.09, d = 0.25, u = 0.66, a = 0.27$). Nature chooses the type of this column player as an attacker by $p^A_i$ or a user otherwise. The normal-form trees on the left and right sides are the game tree with a specific type of column player. The two utility values under each tree node are the utilities of the two players, either a user vs. an attacker (i.e., Eq. (4.21) and Eq. (4.15), respectively) on the left tree or a user vs. a user ($u_{i,n}^j$ in Eq. (4.22)) on the right tree.

Table A.3: The Expected Payoffs of (Defender) vs. (Attacker, User)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>0.20, 0.14</td>
<td>0.20, 0.14</td>
<td>-0.01, -0.07</td>
<td>0.19, 0.13</td>
<td>0.19, 0.13</td>
<td>-0.02, -0.09</td>
<td>0.22, 0.15</td>
<td>0.22, 0.15</td>
<td>0.02, -0.07</td>
<td>0.24, 0.17</td>
<td>0.24, 0.17</td>
<td>0.03, -0.04</td>
</tr>
<tr>
<td>M</td>
<td>0, -0.07</td>
<td>0, -0.07</td>
<td>0, -0.07</td>
<td>0, -0.08</td>
<td>0, -0.08</td>
<td>0, -0.06</td>
<td>0, -0.06</td>
<td>0, -0.06</td>
<td>0, -0.06</td>
<td>0, -0.04</td>
<td>0, -0.04</td>
<td>0, -0.04</td>
</tr>
</tbody>
</table>

$0.48, d = 0.24, u = 0.28, a = 0.67$ with the opponent of either a user or an attacker $\{b = 0.09, d = 0.25, u = 0.66, a = 0.27\}$. The row player, a U-type user, obtains the current opinion by interacting with several true informers and attackers, whereas the column player, an attacker, obtains the current opinion from the previously interacted users.

Nature determines the probability of an attacker by a prior assumption of $p^A_i = 0.1$ in our setting. If the column player is an attacker, it will have four strategies with payoffs. However, if the column player is a user, it will have three strategies with payoffs. Based on the utilities of each type, Table A.2 shows the payoffs of two players. The column strategies have two strategies from each type of possible player, and the payoffs are also based on the probability of nature selecting each type of player. The NE solutions are $(SU, (S, SU))$, $(SU, (S, U))$, $(U, (S, SU))$, and $(U, (S, U))$. The row player user will update his opinion by $SU$ or $U$, and the column player attacker will take deception strategy $S$ if he is an attacker. Otherwise, the row player user will take strategy $SU$ or $U$ to update his/her opinion.

Figure A.3 provides an example of the game tree of a defender taking actions on the opponent of either a U-type user or an attacker with the opinion $\{b = 0.08, d = 0.60, u = 0.32, a = 0.11\}$. The defender can guess the opponent to be an attacker by the probability $p^A_i$, and
Figure A.3: A normal-form game tree of a defender recognizing a player with an opinion 
(b = 0.09, d = 0.25, u = 0.66, a = 0.27) as malicious. Nature chooses the type of this player
as an attacker by a rate $p_D$ or a user otherwise. The normal-form trees on the left and right
sides are the game tree with a specific type of column player. The two utility values under
each tree node are the utility of the two players, the first is a defender in Eq. (4.18), and the
second is an attacker in Eq. (4.15) on the left tree or user in Eq. (4.21) on the right tree.

the payoffs from each column player type are weighted by $p_D$ as shown in Table A.3. From
the normal-game matrix, the NE solutions are $(T, (S, SU))$ and $(T, (S, U))$, which indicates
that, for the column player, in this example, the defender will recognize it as a malicious
user and terminate this account by strategy $T$.

A.2 Analysis of Strategy Selection of Varying the Ratio
of H-DM and Other Users

Figure A.4: The probability distributions of the best strategies taken by three player roles
during 200 interactions.
A.3 Analysis of Varying the Ratio of Attackers

Figure A.4 compared the strategy choices of each player’s roles in our proposed game model when there were mixtures of H-DMs and other types of DMs. All settings had 10% attackers and 10% true informers. For the other 80% legitimate users, the three mixtures were considered: (1) H-DM: other DMs = 0.8 : 0; (2) H-DM: other DMs = 0.4 : 0.4; (3) H-DM: other DMs = 0 : 0.8. The other DM types meant that users could have equal chances to be either U-DM, E-DM, A-DM, or HE-DM. In a single user type network, all the game players knew the user’s opinion update model. However, when there were multiple user types, all the other players needed to guess the type of the interacting user’s opinion model type.

In the H-DM network, the attackers chose each of the four attacker strategies without much difference in Figure A.4a, while the attackers in the other DMs network favored S and C. In Figure A.4b, H-DM in the network H-DM: other DMs = (80:0) had the least NU and most U and SU than other networks. When there were half H-DM and half other DMs, H-DM and other DMs showed the lowest SU and U and highest NU. This implied that H-DM users’ decisions were influenced by the other users when there was uncertainty in network user types. The defender in the H-DM network had a rare chance to choose T to terminate a malicious account in Figure A.4c. In mixed H-DM and other DMs network, the defender had the same chance of action M or T. However, in the network of other DMs except for H-DM, the defender always chose T to terminate a suspect account. From these findings, we can observe: (1) The attackers in the non-H-DM network tried to more directly change non-H-DM type users’ opinions because their opinions could be affected more by opinions with low uncertainty. On the other hand, in the H-DM network (i.e., all H-DM users), the attackers indiscriminately used their strategies, causing high uncertainty to mislead the opinions of H-DM users; (2) In the non-H-DM network, the defender tended to terminate more users. On the other hand, in the H-DM network, the defender chose to monitor a user further. This implied that in the non-H-DM network, attackers were more likely to be removed, and their effect on propagating disinformation was substantially mitigated. This could also well explain the trends of the user strategies; and (3) The H-DM type users update more often and contribute to propagating disinformation more than the other DMs users.

A.3 Analysis of Varying the Ratio of Attackers

Further experiments are performed to investigate the influence of initial attackers, i.e., $P_{false}$ in each type of network dynamics. Figure A.5 plots the dynamics of S, I, R, and Removed nodes after 200 interactions by increasing the $P_{false}$ from 5% to 25%. The results show a clear trend that U-DMs in Figure A.5a are the most resistant to the number of attackers because the I is close to 0 for all the ratios of attackers. The I increases significantly for H-DM users, and most users believe disinformation in 25% of initial attackers. The other E-DM, A-DM, and HE-DM type users show similar patterns: the I has a low level when there are fewer attackers, but the I increases along with the increase of $P_{false}$. 
A.4 Additional Results Under the Dataset Cresci15

In this section, all the results are generated using the Cresci15 dataset [46] to analyze the proposed opinion model framework. Figure A.6 plots the dynamics of opinions after 200 user interactions using the five OM types by plotting \( b \) (belief), \( d \) (disbelief), and \( u \) (uncertainty). In general, the results are highly similar to those based on the dataset 1KS10KN [296, 298] discussed in the main file. Although users updating opinions under different OMs show distinctive dynamics, U-type users can reach the highest beliefs, while H-type users have the most diverse and polarized opinions.

Figure A.7 illustrates the details of dimensions \( b \) (belief), \( d \) (disbelief), \( u \) (uncertainty), \( a \) (base rate), and \( P(b) \) (projected belief) by histograms after 200 interactions and compares the opinion dynamics of the five types of OMs. These results confirm that U-type users show the highest belief, lowest disbelief, and the highest projected belief among all the OMs. The advantage of beliefs in Figure A.7a from U-type OM to other OM types is more noticeable in the OSN using the Cresci dataset than in the 1KS10KN dataset.
A.4. ADDITIONAL RESULTS UNDER THE DATASET CRESCI15

(a) Uncertainty-based OM  
(b) Homophily-based OM  
(c) Assertion-based OM  
(d) Herding-based OM  
(e) Encounter-based OM

Figure A.6: The evolution of SL-based opinions of all legitimate users over 200 interactions

Figure A.7: The histograms of three opinion masses (i.e., \((b, d, u)\)), the base rate \(a\), and the projected belief \(P(b)\) for all legitimate users after 200 interactions: Each subfigure shows the histograms for the five OMs in a given element under the dataset Cresci [46].
APPENDIX A. SUPPLEMENT DOCUMENT: MITIGATING INFLUENCE OF DISINFORMATION PROPAGATION USING UNCERTAINTY-BASED OPINION INTERACTIONS

Figure A.8: The probability distributions of the taken strategies by each player type based on NE solutions and the solutions by our proposed game under Cresci [46].

Figure A.9: The comparison of the average payoffs of the strategies taken by each player type based on the NE solutions and our proposed game under Cresci [46].
A.4. ADDITIONAL RESULTS UNDER THE DATASET CRESCI15

Figure A.10: The community plots of network topology based on three community detection algorithms, bipartitions, modularity, and label propagation, to show networks before and after disinformation propagation under the five OMs under the dataset Cresci [46].
Figure A.11: The polarization scores from the four polarization metrics measuring the graph partition of three community algorithms under the dataset Cresci [46]: Each metric compares the OSN networks before and after disinformation is propagated over 200 user interactions under the five opinion models.

Figure A.8 shows the best strategies chosen during 200 user interactions under each player type in our proposed game and NE. Attackers’ NE solutions for U-type, A-type, HE-type, and E-type users are mostly $S$. In contrast, the NE solutions for H-type users are $C$ more than 50% times, as shown in Figure A.8a. The H-type OM produces more similar probability distributions of the attacker’s strategies taken under both our and NE games than other OMs. Under our taken strategies, the attackers choose all strategies with a tiny difference in Figure A.8a. Still, the attackers in all other networks adopt $S$ and $C$ more frequently. The legitimate users’ NE solutions in Figure A.8b under all user types significantly increase $U$ and $SU$ to update their opinions, compared to $U$ and $SU$ taken in our proposed game. The users’ NE solutions indicate that HE-type users have the slightest chance to update their opinions. Our game model has the opposite situation for the users, where all user types take strategy $NU$ much more than $U$ and $SU$. U-type users have the slightest chance to update, while H-type users update their opinions more frequently, as shown in Figure A.8b. The defender’s choices from both NE solutions and our game model fit well except for the A-type user network in Figure A.8c. Under both conditions, the defenders in the U-type and HE-type networks prefer $T$ to remove malicious accounts, and those in the H-type and E-type networks barely take $T$. However, the defender in the A-type user network has a relatively equal chance of $T$ or $M$ from NE solutions rather than a strong bias to $T$ from
A.4. ADDITIONAL RESULTS UNDER THE DATASET CRESCI15

![Graphs showing betweenness scores](image)

Figure A.12: The betweenness scores before and after 200 user interactions and opinions updates with respect to the five opinion models (OMs) under the dataset Cresci [46]. In (f), we also indicated the mean betweenness of an individual user in the label on the top of each bar.

![Graphs showing trust scores](image)

Figure A.13: The trust scores before and after 200 user interactions under the five opinion models (OMs) under the dataset Cresci [46]. In (f), we also indicated the mean trust of an individual user in the label on each bar.
our proposed game.

Figure A.9 plots the payoff values of each strategy between NE solutions and the solutions taken by the players to confirm the preferences of best strategies under a specific user type player in Fig A.8. For all three player types, the higher utilities correlate with stronger strategy preferences.

Figure A.10 shows the comparative analyses of network topology changes from the communities of the initial state and disinformation propagation based on the five opinion models (OMs). In Figure A.10, we observe distinct communities formed depending on a different community detection algorithm with disparate clusters in the plots. The node colors reflect each node’s projected belief \( P(b_i) \) in a color bar, where belief in true information is in blue and belief in disinformation is in red. Green in subgraphs (a), (g), and (m) represents the degree of the initially projected belief, i.e., \( P(b_i) = 0.5 \). Although various community detection algorithms generate different communities, they all have the same trends to reveal users’ opinion polarization caused by disinformation. Most U-type users form their opinions as true information. On the contrary, H-type users in two distinct communities believe either highly true or highly false information in the communities detected based on all three community detection algorithms. Herding-based OM also forms communities with high beliefs similar to the U-type OM.

Figure A.11 demonstrates the polarization measures under three community algorithms. After U-type users update their opinions, the network decreases the polarization scores under all cases except for the community boundary score in label propagation communities. The polarization scores of the H-type user network in Figure A.11c increase under all four methods compared to the initial polarization scores. In the other two communities, the polarization scores increase under two methods while decreasing under the other two methods, as shown in Figures A.11a and A.11b. However, the H-type users have the highest polarization scores compared to other OMs. Higher polarization means a more polarized network when diffusing disinformation in the network of U-type users.

Figure A.12 plots the distribution of all users’ structural social capital (STC) in the five OMs before and after disinformation propagation. For the H-type user network in Figure A.12b, the users have the largest betweenness compared to other types. This implies that H-type users gain more information from the network structure and have more structural holes to achieve higher STC. This result also supports a more partial distribution of network power among users in the H-type user network.

Figure A.13 plots the distribution of all users’ trust, as relational social capital (RC), after 200 user interactions using the five OMs upon disinformation propagation. All five OMs show a lower RC after the interactions in Figure A.13f. Similar to the STC, trust values in H-type networks are the highest among all five OM networks. This represents that groups with high homophily also have higher trust within the group. This can introduce more opinion/network polarization due to the strong acceptance of only similar minds.