UNIVERSITY OF SOUTHAMPTON

Strategic and Adaptive Behaviours in Trust Systems

by

Taha D. Güneş

A thesis submitted in partial fulfillment for the
degree of Doctor of Philosophy

in the
Faculty of Engineering, Science and Mathematics
School of Electronics and Computer Science

March 2021

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF ENGINEERING, SCIENCE AND MATHEMATICS
SCHOOL OF ELECTRONICS AND COMPUTER SCIENCE

Doctor of Philosophy

by Taha D. Güneş

Intelligent systems are having significant impact on our daily lives in many ways. These systems can help guide human decisions, act on our behalf and cooperate within mixed-initiative teams. This inter-dependency between humans and AI systems inherently includes a level of risk about the outcomes of actions. To mitigate this risk, the concepts of trust and reputation have received significant attention in multi-agent systems (MASs). Numerous techniques have been proposed to answer the interrelated questions of how to reliably assess the trustworthiness of autonomous systems, how to make robust decisions under uncertainty, and how to establish trust between agents and between agents and humans. Computational models of trust typically focus on evaluating the trustworthiness of others using direct observations and the opinions of others in order to select partners for delegation or to form and maintain relationships. Significantly less attention, however, has been given to understanding how these systems can reliably operate under budgetary constraints, or their vulnerabilities to external attacks. In this thesis, we propose and evaluate a suite of new decision-making strategies to progressively select trustworthy partners under budgetary constraints. First, we show how this decision-making problem maps to budget-limited multi-armed bandit problems. We then present new decision-making models that incorporate both observations and opinions from others. Finally, we show how these approaches can minimise costs associated with, and the risks involved in interaction with agents with varying and uncertain reliability.

In order to better understand the performance and reliability of such algorithms, we propose a novel, generic method to automate the process of identifying vulnerabilities in Trust and Reputation systems. We do this by mapping the vulnerability analysis problem to an optimisation problem, and show how efficient sampling methods can be used to search the attack space. We devise an attack model and generate attacks that involve the injection of false evidence to identify vulnerabilities in existing trust models. In this way, we provide an objective means to assess how robust trust and reputation algorithms are to different kinds of attacks and conduct comparative analyses.

Contents

List of Tables

List of Figures

Acknowledgements

Firstly, I would like to thank my supervisors Tim Norman and Long Tran-Thanh for all their support, guidance and patience throughout my PhD. Their help encouraged me to strive for academic excellence and broaden my vision in many different research areas. I am very grateful for everything.

I would like to especially thank my undergraduate academic advisor, Murat Şensoy, who have been supporting me throughout my academic career. Getting prepared for this work would not be possible without his help. Another major support was from Reyhan Aydoğan, who pushed me to attend some academic competitions and introduced me to interesting areas of research. Of course, I would like to extend my gratitude to all of my other professors from my undergraduate studies. I hope they would never lose their enthusiasm for teaching.

My apologies to those I may miss (due to space), but I would like to mention the people from AIC group, Edoardo, Paolo, Tin, Dorota, Alex and all other members who always provided intriguing ideas about my research, also provided bibliographic materials, and made my life easier in Southampton. My long time friends, Okan, Furkan, Bahadır, Enes, Cihan, and Ömer deserve a big thanks as well.

My studies were funded by the University of Southampton ECS Doctoral Training Programme. This research was possible by the use of the IRIDIS HPC facility at the University of Southampton.

I am very grateful to have such a supporting family. Harika, Taner (my mom and dad) and of course Kağan, my brother, never stopped encouraging and providing support even in the most difficult times of my PhD. They never stop inspiring me.

The final thanks go to Hsiao-Hsuan. I deeply appreciate your love, understanding, moral support and care.

Chapter 1

Introduction

The use of automation in support of individual activities that are dependent on each other, and team-based tasks have been a complex problem that is tackled in computing. Solving issues around this automation can be considered under *social computing* problems. Solutions to such problems are inspired from the intersection of social concepts that are from major fields such as psychology, sociology, philosophy and economics. This can be seen as a result of increasing needs in challenging automatisation tasks which are usually interconnected and interdependent. While social concepts take humans as their central focus, the ramifications of these needs have led to the idea of creating virtual societies to solve such challenging problems where in some instances the social concepts may apply. Delegation of these problems to virtual agents (for instance, delegating a purchase order of a cheap flight ticket to a software agent) requires communication and interaction in many levels between various parties. These interactions can be complex and can include several subtasks such as negotiation (software agent interacts with multiple vendors), auctioning (or software agent enters into auctions with other software agents), acquisition of services (or software agents collects more information) and more. Therefore, a decentralized approach was posited to tackle these inherent characteristics.

The decentralized approach is having multiple parties (i.e. *agents*) in virtual societies which may depend on each other in a distributed manner. These virtual societies and their participants have been studied as multi-agent systems (MASs) (Jennings, 1993). The MAS abstraction offers a distributed representation of tasks which allows many features include re-usability, parallelization and better problem solving by decoupling various components. However, this way of reducing complexity has some drawbacks as well, for instance, the need for communication within society. This is essential since the agents need to coordinate with each other in order to accomplish tasks. Because of the nature of these engagements, including the *social* concepts is a pragmatic choice (Gasser, 1991). One such social concept is *trust*.

The notion of having trust within societies is posited to be an important concept to act as a supportive layer (Lewis and Weigert, 1985; Bromley, 1996). This layer is studied in varying human-centric fields: Psychology, sociology (for instance, the management of trust relationships in network structures (Buskens, 1998)), philosophy (for instance, reciprocity in trust and free-riding (Hume, 1978)) and economics (for instance, game theoretical models that study the effect of reputation between two players (Celetani et al., 1996)). The main premise of having a trust is that an unknown portion of members can behave in an unexpected manner, meaning that they may have malicious intentions, or they may be incompetent (i.e. unreliable) in varying levels. This is problematic in many MAS applications, such as automated negotiations where agents interact with each other to reach agreements and common understanding where agents are tasked to aggregate information from multiple sources.

These ideas resulted in a subsystem within MASs where this concept can be modelled and used, which is known as Trust and Reputation Systems (TRSs, also known as trust models). These computational models are designed to support agents to make predictions of others' future performance before making a decision on whom to rely. These types of support components of agents' decision-making are also known to be a *soft security*[1] that can be seen as collective enforcement of specific norms by the participants of a community. Such systems incentivise the behaviours that are generally accepted and sanction those that are not (Josang and Haller, 2007). A variety of methods have been preposed to implement TRSs, some of which we discuss in Sections 2 and 3, and concrete examples of their use include electronic markets such as Amazon[2], Ebay[3], Airbnb[4], ride-sharing companies such as Uber[5], Lyft[6] and many more use TRSs including autonomous vehicles (Schneider, 2017) to enhance their overall service quality, not only for the consumers but also for service providers as well.

In this research, we focus on advancing general-purpose TRSs by addressing challenges in the *decision-making process* of agents and *vulnerabilities* that may affect this process. Our main concern is to look at particular environmental settings that decision-makers may be in and devise techniques to enhance the robustness of TRSs. We put our focus on trust approaches in MASs and treat trust-related transactions as abstract as possible, thus allowing them to be applicable in a wide variety of MASs applications.

[1] In this context, *hard security* is mostly associated in the related work with infrastructure level security that involves areas such as encryption, authorization, and authentication of MASs (Barber and Kim, 2003; Bertocco and Ferrari, 2008).

[2] `https://www.amazon.com`

[3] `https://www.ebay.com`

[4] `https://www.airbnb.com`

[5] `https://www.uber.com`

[6] `https://www.lyft.com`

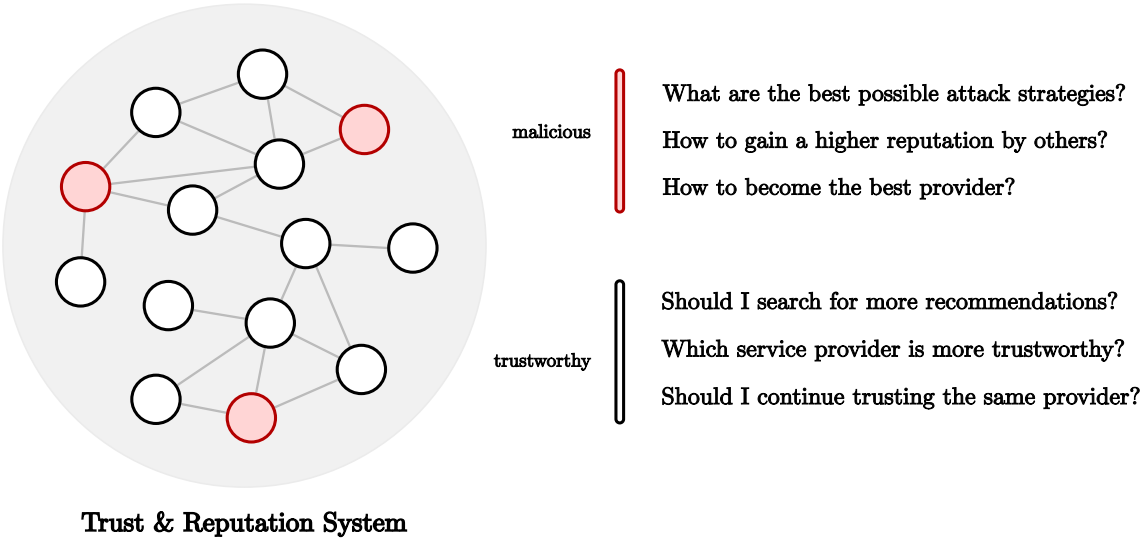

FIGURE 1.1: Conflicting challenges that agents face in Trust and Reputation Systems

1.1 Motivation

The underlying philosophy of utilizing *trust* is to drive increased service quality and to increase agents' confidence in the outcomes of future transactions. This is achieved by encouraging agents to provide feedback on services and goods that are visible to others in aggregate, with commentaries often associated with individual ratings. However, there are inherent challenges in such systems where typical trust models implicitly rely on simplifications. We elaborate on these challenges by a hypothetical example as follows.

Assume that an agent is regularly given a task, which is to purchase a common cloud computing service for a specific usage. We call this as a consumer agent. The service that this agent is interested in is provided by many vendors with varying prices. This is a challenging problem, including cases where the agent has no prior knowledge about how *good* the services are provided by different vendors (for instance, AWS[7], Microsoft Azure[8] or Google Cloud[9]). Any commitment to a service can be *costly*. Not only the monetary costs, but also possible opportunity costs due to time required for integration and development to utilize their service can be present. The agent can query other agents to get their opinion about the vendors (for instance, checking reviews in TrustPilot[10]) before making any commitments. Collecting such information may be useful and sometimes *free*, however, it can be costly in terms of time, and they may be tailored to be manipulative as well. In addition, opinions from more reliable sources can be purchased, which is, of course, has monetary costs. Alongside with this conflicting challenges as

[7] `https://aws.amazon.com/`
[8] `https://azure.microsoft.com/en-gb/`
[9] `https://cloud.google.com/`
[10] `https://uk.trustpilot.com/categories/cloud_computing_service`

shown in Figure 1.1 arises a fundamental question: how an agent with a limited budget can make purchases of these services, meanwhile robust to manipulations.

This hypothetical scenario is not an isolated case. The problem of deciding whom to trust exists in many domains. Managing risks in operational activities in supply chain management can be given as an example where decision-makers need to achieve a level of performance on several objectives, such as on-time delivery, order completeness, order correctness and defect-free delivery (Gaudenzi and Borghesi, 2006). Any decrease of the level of performance on these objectives requires strategic balancing with the given the resources of the decision-maker for both short-term and long-term profitability of organizations (Wu and Pagell, 2011).

In terms of manipulation, there are many similar situations in e-commerce where users have a choice among a range of similar options, *relative* ratings can have a big impact on decisions. This, of course, introduces a strong incentive for companies and individual services/goods providers to game the system. For instance, many sellers in Amazon struggle with increasing fake reviews (Clayton, 2020), which is an ongoing challenge in many platforms. In response, platform owners introduce controls; for example, only to permit reviews from confirmed customers. This has led to more sophisticated attacks, such as those reported by the Wall Street Journal (Emont and Bürge, 2018), where items are purchased and then returned to qualify to inject negative reviews. Users may report such incidents, but the moderation process is manual, time-consuming, and maybe equally used by dishonest sellers.

Such challenges also exist in multi-agent systems research. While many MAS challenges tend to be specific to the domain of interest (Dorri et al., 2018), there are common features in domains such as task allocation, learning, organization, coordination and security where trust is applicable. We summarize the main challenges regarding the usage of trust in MASs as follows:

- *Variety of information*: Ideally, a trust model should be able to recognize and use various types of information sources strategically with respect to the present situation. Many types of information may be utilized in trust formation including an agent's observations of others, third party opinions, sociological information (Burnett, 2011). The question of when and how each relevant information source can be exactly utilized or fused together for assessments is vital in decision-making. There are many intrinsic characteristics that need to be taken in to account. The first-hand experience may be costly, but more reliable only if it is considerable (Sabater and Sierra, 2005). In cases where this is not possible, other types of information can be more effective (Sutcliffe and Wang, 2010), which include exploitation of information through some centralized or decentralised MAS mechanisms (Seuken

and Parkes, 2014), such as aggregated information (such as reputation) or published features of agents (such as the owner of the agent or the capabilities of the agent).

- *Constraints*: As articulated in our example scenario, information gathered throughout the decision-making process is useful for trust formation. However, collecting evidence may require investment of resources. Thus, rather than simply needing to know which providers are more or less trustworthy, customer agents need to make sequential and strategic decisions in selecting partners (Sen, 2013). The decision-making must be general enough to satisfy various requirements, including handling varying costs to acquiring different types of information. Furthermore, decision-makers will not know how *useful* their investment was at the time of their investment.

- *Manipulation*: Traditional multi-agent security mechanisms usually focus on protection from outsider attacks; i.e. protection from third-party entities, which originate out of the system (Josang and Haller, 2007). On the other hand, these systems are known to be vulnerable to internal threats. These adversarial agents may use the actions that MASs provide instead of malicious behaviour by the means of *hard security*. This shows the need for trust management in MASs to aid traditional security mechanisms.

  Identification of agents in MASs (e.g. peer-to-peer networks) is one of the possible mitigation strategies. Typically, for instance, when a malicious agent is detected in the system, the agent can be removed and any further attempts to join the system with the same identity can be disabled. However, one can generate several new identities to enter the system again (Douceur, 2002) to damage service providers' reputation by giving biased ratings. As a result, overall performance of the system suffers an unfair advantage can be given to some specific service providers.

  Agents that are trustworthy for a long time may start to behave maliciously (Sun et al., 2006). Then how significant the previous evidence of an agents' behaviour with regards to its current behaviour? Which part of historical data must reflect the trustworthiness of the entity? Lastly, how can the system demotivate this unexpected behaviour? Rather than removing the agent completely from the system, it might be feasible to exploit this inconsistent behaviour (i.e. delegating tasks only if the malicious entity is predicted to behave as expected.)

1.2 Problem Statement

In this thesis, we focus on the problem of:

How to make proactive decisions regarding who to interact with by the use of generalized trust evaluations in environments where strategic attackers and resource constraints exist?

Proactive decisions that we aim to make in our problem setting must consider future events with the features observable in the present. This is parallel to making decision step by step, since the decisions that are made earlier can influence the future decisions. We will now give an overview of the contributions of this thesis towards our research into this problem.

1.3 Contributions

Devising mechanisms to make strategic decisions is a challenging problem. Firstly, the system can have multiple self-interested agents with unknown behaviours. Secondly, each of them might have varying resource constraints. Finally, each system can provide different types of evidence. We argue that dealing with these complexities requires a strategic decision process that takes into account the constraints of decision-makers and possible adversarial behaviours. Therefore, what vulnerabilities that adversaries may use is also our another focus. For this, we now go through our contributions that are done in pursuit of our research aim in the following section.

- *A general decision model on trust and budgetary constraints* (Chapter 4): We introduce a selection of decision processes that are compatible the statistical trust models in resource-constrained environments. These processes are designed to make the discovery of truthful partners alongside continuing to engage with the known partners while exploiting third-party information.
- *A mechanism to search for strategic attacks* (Chapter 5): Motivated by making these decision processes resilient to possible attacks, we go beyond the known ways to attack such trust mechanisms by introducing a novel mechanism for exploring ways to sample attacks. The mechanism allows attackers to strategise such that they can set objectives and introduce orchestrated attacks.
- *Vulnerability analysis of trust models* (Chapter 5): This mechanism's realization is used to explore vulnerabilities in a selection of trust models. This exploration was not done in a principled manner before. We demonstrate that such orchestrated attacks rely on several features of the environment, which we show by simulated and real-world data sets.

1.4 Thesis Outline

The remainder of this thesis is structured as follows:

- In Chapter 2, we provide a survey of related work in multi-agent systems with a focus on trust and reputation. We discuss the problems of trust estimation, decision-making with trust and compare existing approaches that attempt to address these issues.
- In Chapter 3, we provide a background on the fundamental techniques that we exploit in building our contributions.
- In Chapter 4, we present our decision-making processes and evaluate its performance with varying information sources. We discuss the problems that can arise environment-specific features differ, and how a decision-maker adapt to the processes to support different environments.
- In Chapter 5, we present our attack search mechanism and our vulnerability analysis of a selection of trust models. We evaluate this model within our simulated and real-world data sets.
- In Chapter 6, we provide some potential application domains in our approach alongside with how they can be used or extended.
- In Chapter 7, we discuss our approach with future avenues of research.

1.5 Related Publications

The work presented in this thesis resulted in a number of peer-reviewed academic publications.

- Güneş T.D., Norman T.J., Tran-Thanh L. (2017) Budget Limited Trust-Aware Decision Making. In: Sukthankar G., Rodriguez-Aguilar J. (eds) Autonomous Agents and Multiagent Systems (AAMAS) 2017. Lecture Notes in Computer Science, vol 10643. Springer, Cham.

  Workshop/Conference paper presented in TRUST Workshop at the Autonomous Agents and Multiagents Systems Conference. This paper was selected as one of the most visionary workshop papers in AAMAS 2017 and published as a book chapter following the conference. The publication refers to the initial work that underpins our contributions in Chapter 4.

- Güneş T.D., Tran-Thanh L., Norman T.J. (2018) Strategic Attacks on Trust Models via Bandit Optimization. In: International TRUST Workshop at AAMAS/IJCAI/ECAI/ICML 2018, Stockholm, Sweden on July 14, 2018.

  Workshop paper presented in TRUST Workshop 2018 at the joint conference that held Autonomous Agents and Multiagents (AAMAS) 2018. The publication refers to the initial work that underpins our contributions in Chapter 5.

- Güneş T.D., Tran-Thanh L., Norman T.J. (2019) Identifying vulnerabilities in trust and reputation systems. In: International Joint Conference on Artificial Intelligence (IJCAI) 2019. Macao, China on August 10-16, 2019.

  Conference paper presented in the main track of International Joint Conference on Artificial Intelligence (IJCAI) 2019. The publication refers to the initial work that underpins our contributions in Chapter 5.

Chapter 2

Related Work

Trust and reputation concepts have been widely popular in multi-agent systems. Generally, the main areas of research have been looking into the questions of how to *evaluate others* and how to use these evaluations to *make decisions*. There have been a variety of applications in the forms of community building, human-agent and agent-agent transactions, service provision. As a consequence, there have been various types of models and system designs. In light of our research aims, we narrow our focus to key models in this area to argue the shortcomings and challenges that the decision makers have. Then, we discuss the state-of-the-art approaches towards our objectives to motivate the contributions that we present in Chapter 4 and 5.

2.1 Trust

Statements, such as *"I trust you."*, *"This is an untrustworthy vendor."* or *"This provider has a good reputation."* are made in daily conversations frequently. The meaning of these statements is known to be different in each domain and context. Generally, the common motivator in these sentences is the aim of *reducing the difference between expectations and reality*. This phenomenon is known as maintaining a *psychological contract* (Robinson, 1996) in human-human interactions. When we transition to multi-agent systems, the earliest signs of this uncertainty minimization concept dates back to the dissertation of Marsh (1994). Marsh characterises this phenomenon as:

> "We arrive at the concept of trust as choosing to put ourselves in another's hands, in that the behaviour of the other determines what we get out of a situation."

Ramchurn et al. (2004a)'s definition covers both reciprocity and the game-theoretical dimension of trust:

> "Trust is a belief an agent has that the other party will do what it says it will (being honest and reliable) or reciprocate (being reciprocative for the common good of both), given an opportunity to defect to get higher payoffs."

It is important to mention that there is no consensus on a definition of trust. If we look at the most recent extensive reviews, the definition is on the probabilistic side:

> "An agent's trust is a subjective belief that the selected party acts according to the agent's expectation during an interaction" (Yu et al., 2013a; Cho et al., 2015)

This is generally the well accepted definition in the multi-agent systems research. Trust that an agent has about another is thought to be a subjective belief (typically modelled as a subjective probability), that each party relies on to evaluate others. On top of that, we will elaborate in the following sections why we decided to see trust as in subjective probabilities. Our reasoning comes primarily comes from the challenges in other types of approaches (for instance, cognitive approaches) that have a different view on trust.

Before elaborating on how we might compute the trustworthiness of an agent, it is important to discuss the types of *inputs* that are used in trust models. Different environments engender a different sets of inputs. Specific models, therefore, tend to cover different realizations given the environment concerned. This also serves to fill the conceptual gaps in the definition that we provided.

2.2 Sources of Trust

Forming subjective trust beliefs (i.e. trust assessments) inherently requires *some* information. Decentralised models typically utilise observations from direct interaction and information from contacts that can act as witnesses. In contrast centralized systems have access to all *inputs*. All systems need to take into account subjective biases in the available information, address the *cold-start* problem, and consider complex attacks. To address these challenges, the studies have explored additional information that might be present in the environment. Here, we give an overview of the literature focussing on the types of information considered in existing models. We adapt and combine the taxonomy from extensive reviews (Ramchurn et al., 2004a; Sabater and Sierra, 2005; Yu et al., 2013a; Granatyr et al., 2015; Ruan and Durresi, 2016).

2.2.1 Direct Interactions

Direct Interactions (DI) of an agent is defined as the collected historical evidence concerning its transactions with other agents. This is also referred to as *direct evidence*. The

collection of this evidence in multi-agent systems can be through service requests and responses. In terms of what is a *service request*, we give concrete examples from four major domains: in cloud computing, an agent's action to acquire computational resources from another agent, in online communities (i.e. social networks), a user's transaction with another user, in cybersecurity, an agent's request to retrieve information from a sensor agent, and in peer-to-peer networks, a peer's action to request a file from another peer (Dorri et al., 2018). The collected historical data is then used for estimating the outcome of subsequent interactions. This is widely used as source of information that is used by the models for especially as a means of *first-hand evidence*.

A short example of direct interactions can be given in the context of e-commerce. Assume that a product is purchased by a buyer agent. After the product is received from the seller, the buyer agent evaluates the transaction whether the buyer agent would purchase another item again from the seller. We are not interested in the preferences of buyers and how much the purchased item fulfils these preferences. Formally, this transaction refers to *Agent A* interacting with *Agent B*. The outcome is the information that is valuable for assessing *Agent B* for *Agent A*. *Agent A* commonly called as *truster* or *consumer* (Sometimes we refer this role as the decision-maker agent). While *Agent B* is called *trustee* or *provider*.

This kind of evidence is used by different trust models in different ways. Marsh (1994) use a set of rules, such as reciprocation to update the trustworthiness of others. Griffiths (2005) adopts a weighted product of beliefs in a multi-dimensional manner. For instance, while one dimension would be *trust* on the availability of bandwidth, the other dimension would be the *trust* on the service quality. Ghanea-Hercock (2007) uses incremental updates to *trust* estimation based on whether a provider would *defect* or *cooperate*.

Temporal Information

The temporal information of interactions is complimentary to direct evidence. This is known to be useful when agent behaviour is considered to be *dynamic*. A well known malicious behaviour, *oscillation* attack, where providers alternate between desirable and undesirable behaviour, is an example of where the models that don't consider temporal information fail. Especially, a sudden change of behaviour is a complex problem, where popular models fail to handle.

The earliest and most popular of related work to deal with this complexity that was proposed is to have a moving time window. This is called the *forgetting factor*. According to this factor, the effect of previous interactions is discarded or reduced. The earlier work does not cover how to determine this value (Jøsang and Ismail, 2002).

2.2.2 Direct Observation

One of the drawbacks of relying on direct interactions completely in the models is *cold-starting*. The trust models face the *cold-starting* problem when there is no historical evidence available. This may be due to making a series of direct interactions is not possible. Therefore, agents in such systems would end up assuming that *trustworthiness* of all other agents as equal. In such situations, *direct observations* are proposed that can be useful (Sabater and Sierra, 2005). The idea is to have a repository where the system that agents are in *trust system* all direct observations are recorded. This repository can be in the forms of a *distributed ledger*. This is analogue to each party having a copy of the available data outputted by the system. Thus, this source be provided in centralized and decentralized systems. HABIT model can be given as an example that uses this information (Teacy et al., 2012).

The models that incorporate this type of information are few. The ones that they do are using aggregated metrics from the system, such as the number of direct interactions made, the roles of agents, and the number of *unique* interactions that were made (for instance, the number of interactions with various parties, instead of a clique) (Carter and Ghorbani, 2003; Klejnowski et al., 2010). One of the reason is this type of information may be biased or altered. By this, the trust system may need an extra mechanism to reduce the effect of manipulation (for instance by the use of cryptographic signatures). The next information type handles this drawback taking into account third-party information in a more decentralized manner.

2.2.3 Witness Information

Referrals, advises, reputation, word-of-mouth and indirect evidence (i.e. indirect knowledge) are known to be *witness information* (WI) (Sabater and Sierra, 2005; Yu et al., 2013a; Granatyr et al., 2015). Witness information refers to the third-party information that the decision-makers collect from other members of the system. This type of information is known to be useful when agents do not have any direct interactions or direct observation knowledge. Going back to our previous example of two agents, *Agent A* estimating the *trustworthiness* of *Agent B*, *Agent A* can get *Agent C*'s opinion about *Agent B* without committing to a transaction. If we compare the previous information source and witness information, the difference is that where the type and the source. In witness information, the decision-makers query on the third party's opinions. However, in direct observations, the decision-makers query the system. Aggregate types of information such as the number of interactions between a pair of agents or direct interactions between parties can be collected by this source.

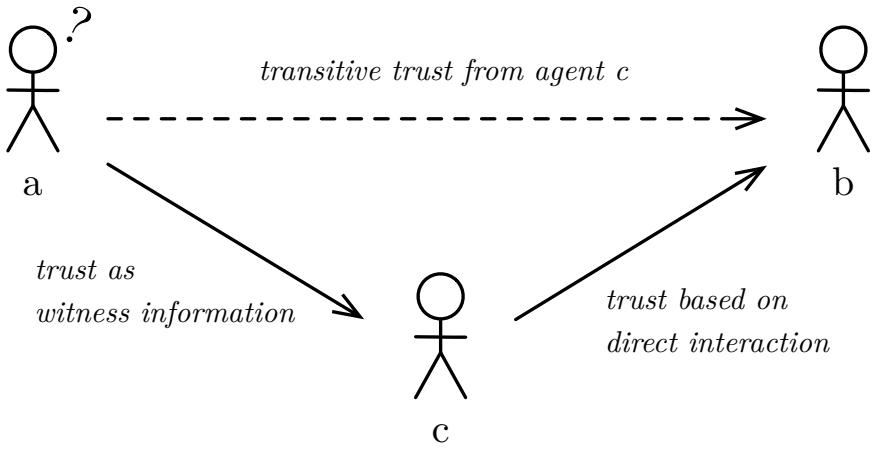

FIGURE 2.1: Trust transitivity relationship between agents

Direct observations and witness information suffer from *bad-mouthing*. This is where several malicious parties continuously report false information. This attack can be extended to a more complex form: malicious cliques. In these, malicious parties increase their reputation by witness information continuously within themselves and reject interacting others that are not from the clique. After the reputation-building phase, agents that are outside from the clique may perceive them as *trustworthy*. Thus, using this source requires defensive strategies this type of attacks. We will elaborate on these in the next sections.

The consumer agents may ask the provider agent to show evidence about their trustworthiness. One of this evidence can be *referrals*. This is called *certified reputation* (Huynh et al., 2006), which resembles the recommendation letters of people that are for job applications. Before *Agent A* interacts with *Agent B*, *Agent A* queries agent B and receives a set of reports about her. It can be seen as a short way to get witness information about agent B. As a result, the traffic of querying witness information in the network decreases in an ad-hoc MAS. In the cases where *Agent B* mimic another agent's identification, Hard security measures can be used while collecting witness information to make sure the identities of advisor agents are not copied (Botelho et al., 2009). Implementing this mechanism empirically has been shown to have benefits in addressing the *cold-start* problem (Huynh et al., 2004).

Beta Reputation System (Jøsang and Ismail, 2002), BLADE (Regan et al., 2006a) and TRAVOS (Teacy et al., 2006) can be given as an example trust models that use this information alongside direct interactions. The models use of witness information ranges from weighting heuristics to probabilistic graphical models. Witness information is usually combined with other information types. There are a few models that only use witness information (Granatyr et al., 2015).

2.2.4 Sociological Information

The relationships between agents can be modelled as an undirected graph. These graphs are often called a *trust network*. Edges of this graph denote trust relationships between agents. Intuitively, witness information can be thought of as equal to sociological information in the following scenario. Assume two agents that are not connected (i.e. never had a direct interaction before) in a network (for instance, two hops away) uses intermediate (i.e. middle, advisor) agents to determine the trustworthiness of another. This *trust transitivity* can be used in the computation of trustworthiness of others. However, the main difference in the related work of usage of *sociological information* is the extraction features from the network (Sabater and Sierra, 2005).

The roles of agents, ontologies, stereotypes and reputations of nearby agents are examples of sociological information. Sometimes the combination of all is called the context of the trust relationships. These features can be learned and exploited to be used as an indicator of the trustworthiness of unknown agents. Burnett et al. (2010) shows that each agent can have a *stereotype*. These stereotypes can be learned by the decision-making agent with the use of decision-tree algorithms. The estimated stereotype is used as the prior probability estimation in Bayesian trust models. This is shown to be useful in ad-hoc agent groups (Burnett et al., 2013). Similarly, Liu and Datta (2012) shows usage of contextual information in e-commerce setting. The features that were used include, for example, the average age of the profiles, average delivery time and the degree of contact information profile.

Chhogyal et al. (2019) uses *values* that agents' share when there is no other information available. The values of agents are considered to be *abstract* that represent compatibility of agents, the degree of conflict between agents. Compared to *stereotypes*, their model makes trust assessment between two agents by computing the number of values that agents share minus the number of their values that contradict each other. They explain the use in the transitivity in trust sequences for tasks that require multiple agents cooperation. The situations such as *Agent A* delegates a task to *Agent B*, and *Agent B* delegates a dependent task to *Agent C*. At each link, they use their assessment method. Thus, values of *Agent A* and *Agent C* are not compared. After the trust assessments, each agent selects the most trustworthy agent for delegation.

2.3 Computational Models of Trust

In this section, we focus on numerical models of trust and reputation systems (TRSs) instead of cognitive methods Falcone and Castelfranchi (2001); Pinyol et al. (2012); Piunti et al. (2012). The cognitive methods are proposed to incorporate the use of characteristics of agents to form the trust. They may, for example, verify the ability of

an agent using beliefs about that agent. We exclude these models from our scope for several reasons. First, these high-level models are justified theoretically and there is no concrete instantiation that we are aware of. As Ramchurn et al. (2004a) pointed out these models require learning aforementioned factors. However, the data that is related to the behaviour of parties to form a set of beliefs in different dimensions (beliefs such as competence, willingness, persistence, motivation) are not evident in the environment. Therefore, we put our focus on computational models of trust.

One of the earliest statistical models of trust and reputation system that proposed is Beta Reputation System (BRS) Jøsang and Ismail (2002). It is a simple model that uses Beta probability density functions to represent posterior distributions of binary events. This is a practical design choice, where Beta distributions are conjugate prior to the binary events. The reputation of others are combined cumulatively by the use of the parameters of the Beta distribution. To reduce the effect of the previous interactions, their effect is reduced by a forgetting factor (aka. *longevity* factor, *ageing* factor). For example, the decision maker agent stores all the outcomes of a series of interactions with provider j as $O^T_{\delta \to j}$ until time T. The *pseudo-counts* of the associated beta distribution (r, s), where r is for positive and s is for negative transactions, are changed with a forgetting factor λ by:

$$r_j^T = \sum_i^t [O^i_{\delta \to j} = 1] \lambda^{(t-i)} \text{ and } s_j^T = \sum_i^t [O^i_{\delta \to j} = 0] \lambda^{(t-i)} \tag{2.1}$$

The *forgetting factor* is bounded with in the range $[0, 1]$. $[O^i_{\delta \to j} = 1]$ is an Iverson bracket (Iverson, 1962) that returns 1 if the condition inside is satisfied, otherwise 0. Afterwards, the final reputation, τ_j, is given to a provider by:

$$E(\tau_j | r_j^T, s_j^T) = (r_j^T + 1)/(r_j^T + s_j^T + 2) \tag{2.2}$$

This model has been extended with Dirichlet Reputation Systems (DRS) where Dirichlet distributions Josang and Haller (2007) are used for generalizing the binomial Beta functions, allowing any number of discrete rating levels. Similar to BRS, Dirichlet distributions are also conjugate before multinomial distributions. The expectation of posterior probability of behaviour of the parties is used as the final rating for decision makers. To mitigate deceptive agents, Beta reputation system was extended to filter agents who deviate from the majority by Whitby et al. (2004). This work includes a majority filtering mechanism to Beta reputation system by proposing a quantile parameter to filter parties that deviate from the majority. At this point, information gathered from direct observations and others were treated as same. They are added cumulatively and providers that remain in the quantile are kept. Iteratively, this is done from every decision maker perspective, until the set of filtered parties converges. They provide

appropriate values for the quantile parameters that they found empirically given their experimental settings.

Instead of these methods which incorporates Bayesian Learning, FIRE (Huynh et al., 2004) model handles each of the information source individually and computes trustworthiness of an agent by taking a weighed average amongst every information source. In our terms, these are direct evidence, witness information and sociological information, and include a weighting function, similar to the *forgetting factor*. They represent opinions of agents (i.e. ratings) in the range of $[-1, 1]$. Regarding direct interactions, they multiply two metrics: *reliability measure*, and *rating deviation reliability*. These are the ratio of the number of opinions given before with respect to a given threshold and the deviation of these opinions to the average respectively. The way in which they handle direct evidence is adopted from the REGRET model (Sabater and Sierra, 2001). Briefly, REGRET model uses a weighting averaging method and weights are determined by an ontology. FIRE uses the witness information, which is collected cumulatively by after discounting the historical opinions. The sociological information that the uses is *roles* that agents may have. The idea stems from if two agents are from the same organization, they can set a trustworthy prior for each other. These are defined as predefined rules for each decision maker agent. All these components are weighted and averaged for the final *composite* trust value, which is used in decision-making.

TRAVOS (Teacy et al., 2006) adopts the similar mechanisms as Whitby et al. (2004)'s BRS model, but calculates weights of how similar is the decision's makers direct evidence with respect to information from others. This means that TRAVOS does not have to rely on the majority of opinions as in BRS. This comparison is done by a heuristic which compares these two distributions in terms of regions. The trust in these opinions from others relies on the expectation of direct evidence and the witness information being in the same region. The total density mass in this region then is used for the degree of similarity. This approach has shown to be better in cases where there is more false evidence, which makes the majority filtering not feasible.

POYRAZ (Şensoy et al., 2009) introduces the use of ontologies in the service selection domain. In this setting, the deceptive agents are handled differently from TRAVOS. Direct evidence and witness information are weighted according to the level of error that decision-maker can tolerate similar to FIRE. The final trustworthiness of agents is calculated as the weighted average of both of the derived score derived from each information source. These are called *public* and *private credit scores* of agents. The underlying derivation of these is based on BRS for both sources. The use of ontological knowledge with this discounting mechanism, results in higher performance when compared to TRAVOS and BRS, especially in the early stages of simulations. This is useful for mitigating the *cold-start* problem.

Regan et al. (2006b) avoids this heuristic by relying solely on principled theories in BLADE. In this model, trust relationships are represented by Bayesian Networks. Compared to TRAVOS, BLADE supports any number of discrete rating levels similar to DRS. BLADE also removes the filtering mechanism and introduces an implicit discounting mechanism by learning the correlation between direct observations and witness information. It is empirically shown to be outperform TRAVOS when the ratio of deceptive agents is higher. One of the largest limitations of this model is to compute the estimations efficiently every input in the model must have all features available. We will elaborate on this assumption in Chapter 5. The solution does not work in cases where the decision-maker does not have all observations in the historical data (i.e. when it is missing some parts of the data).

HABIT (Teacy et al., 2012) relaxes one of these assumptions of having only discrete observations. Their hierarchical model can allow both discrete or continuous observations. To capture group behaviour in the system, their hierarchical Bayesian model sets a hyper-parameter that denotes the behaviours in the system. In comparison to BLADE model where, their model have $O_{i\to j} \sim \theta_{i\to j}$ dependence (Each observation, $O_{i\to j}$, is distributed by $\theta_{i\to j}$). Each observation between i and j depends on the parameter θ. HABIT proposes the behaviours are distributed according to a hyper-parameter ϕ: For all i and j pairs, this is $\theta_{i\to j} \sim \phi$. To draw inferences using this model, they recommend using approximation methods, such as Markov Chain Monte Carlo (MCMC) and variational methods. The closed-form computations are only provided for specific cases, for instance, discrete observations with a Dirichlet Process (DP) prior. Their results show through simulations that when there is a behaviour correlation between agents, the model outperforms BLADE. The group behaviour that they take into account is simulated by the sampling of a Dirichlet distribution for θ where the sum of the concentration parameters (α) is varied.

More recent models of trust revisit some previous ideas and/or included more complexity to the models. PGTM (Fang et al., 2013) proposes a probabilistic graphical model that was inspired by trust between humans while considering binary outcomes. The social concepts such as competence, benevolence, integrity and subjectivity difference are included in the model. While some of these factors are *observable*, and some are included in the model as latent factors. These latent factors are learned by the closed-form derivations. The inferences are made approximately by collapsed Gibbs sampling (Liu, 1994). The model is designed for e-commerce trust transactions, where users rate *entities*, and advisors provide opinions about those entities.

Jiang et al. (2013)'s MET model (known as multiagent evolutionary trust model) focuses on creating a trust network which consists of a set of advisors for a decision-maker by direct evidence and witness information. They define a fitness function, which is computed by the average difference of direct evidence and witness information. Therefore, smaller the value of fitness the higher the quality of the trust network (i.e. a set of trust

values for each provider). In each generation, they first select three advisors (these are selected uniformly if there is no prior information). Second, a *crossover* operation is invoked stochastically. The operation computes a new trust network at each generation by combining witness information from all advisors from the previous generation. Later, a *mutation* operation is invoked which adds some perturbation (i.e. noise) to the generated trust network. Finally, the resulting trust network is kept only if its fitness value is lower than the previous generation. This process is done until a specified limit. The model assumes that the decision-maker agents have historical information both direct and indirect. It is not clear in their work how to collect this information when there is a *cold-start* in the system.

The models mentioned previously mostly evaluate the performance of their assessments by the use of simulations. In these evaluations, the common metric is *the mean absolute error* of the distance between the predicted behaviour and the assigned behaviour Jøsang and Golbeck (2009) of providers. The challenge in comparing the performance of models in practice starts from the fact that there is no *true* ground truth available. As the ground truth is unknown, the use of real-data tend to use aggregated metrics. The availability of the information that has *true* ground truth is rare. This requires participants to provide some information about their trust relationship. However, this is a challenging problem. For instance, Epinions dataset (*Epinions.com*) (Leskovec and Krevl, 2014) used by PGTM is an example to this. In Epinions dataset, users rate other user' entities (articles). Also users can directly indicate if they trust another users. These indications are used as a ground truth in their evaluation. Therefore, the performance evaluation in trust models is mostly done via the use of simulations. This is by generating various sets of different behaviour parameters and market settings. In fact, this makes comparing different trust models challenging.

2.4 Decision Making with Trust

The computation of trust does not cover the problem of making progressive decisions to who to interact with at a certain time. Specifically, computation of these subjective beliefs does not necessarily allow when to *explore* others, and how much to invest in *exploring*. By exploring, we mean engaging in interactions with agents that are unknown to the decision-maker agent. Although the popular method is to select the most trustworthy agent amongst others (known as *greedy*), this approach does not take into account the constraints that bound the decision-maker: risks, rewards, the amount of budget, and the time. These approaches that adapt to varying environmental conditions are called *dynamic* approaches. They are designed to manage this exploration-exploitation trade-off. The decision-maker has to decide when to search for better alternative providers versus when to continue engaging with known providers.

Griffiths (2006) uses a threshold and grouping based decision-making model. Rather than marking agents as trustworthy or not, the model refers to provider agents that they can be in either *untrust* or *undistrust* (insufficient trust) category, which is determined by thresholds. The model choses to explore others if there are not any trustworthy agent that is determined. The exploration phase of this model is not explicitly mentioned, but a number of the interactions at the beginning are allocated for a uniform random exploration (each agent has an equal probability of being selected).

The extended FIRE model Huynh et al. (2006) proposes a two-step decision-making process that incorporates a Boltzmann strategy (Kaelbling et al., 1996) for exploration and exploitation. At first, the truster calculates the trustworthiness of trustees by witness information which is provided by other parties. If no reputational information is available, the agents concerned are stored in a set called *NoTrustValue* and the remaining trustees are placed in *HasTrustValue* set. After this initial step, the truster determines which action to take by using a Boltzmann exploration strategy. The probability of selecting an agent a_i is determined by:

$$P(a_i) = \frac{e^{\frac{u_i}{T}}}{\sum_{j=1}^{n} e^{\frac{u_j}{T}}} \tag{2.3}$$

where u_i denotes the expected utility and T is the temperature parameter. There are two actions that a truster can take according to FIRE with Boltzmann exploration: select a random trustee from *NoTrustValue*, or select the trustee with the highest trust value from *HasTrustValue*. In the second phase, the temperature parameter, T, is set to a high value. This makes the algorithm *biased* towards to exploration over exploitation. Throughout this phase, the T parameter decreases over time according to a decay parameter, and so the decision-maker shifts into *exploitation* in later interactions. However, the question of what is the decay rate for the temperature value and the optimum value of T is not analysed in a rigorous manner.

Ahn et al. (2008) proposes a simple method, which takes inspirations from Reinforcement Learning and proposes using an ϵ-*greedy* algorithm. Each time decision-maker agent uses their multi-dimensional trust model to make trust assessments over all provider agents. This list is kept sorted with only *known* agents. They probabilistically make between two choices: using an agent from the list, or selecting an unknown agent randomly. The probability of these actions are set $1-\epsilon$ and ϵ respectively. Throughout the transactions, ϵ is made to decay similar to FIRE model. This increases the probability of selecting agents from the list in the later stages.

Fullam et al. (2005) propose a testbed and a competition for trust models and decision-making processes to be evaluated under similar conditions. This was known in the trust community under initiative *Agent, Reputation and Trust* (ART). ART has been discontinued, however, two key approaches were designed for the competition. The testbed had

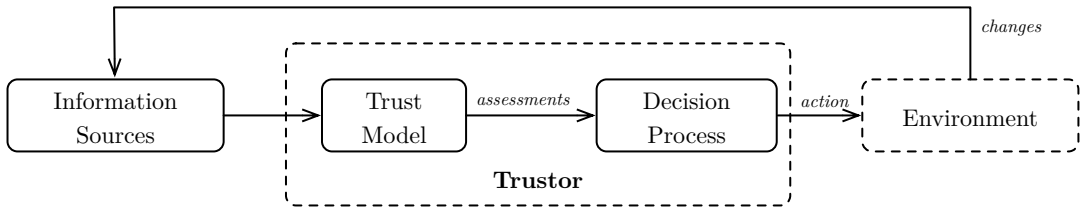

FIGURE 2.2: Trust-aware decision making agent

two operation modes: *competition* and *experimentation*. These two approaches were designed for the competition mode, where decision-maker agents compete with each other. The competition mode is based on an imaginary domain, *art appraisal domain*. The task of consumers is to make a profit from art pieces by betting on their price. However, consumers do not have any knowledge about the price of an art piece. They request opinions from appraisers (i.e. advisors, potential opinion provider). These requests have a fixed cost, and the opinions can be deceptive. The final offer is calculated by the simulation platform by the weighted average of the opinions received. These weights are expected to be computed by consumer agents. Consumer agents have a bank account (i.e. budget) on how much they can invest to witness information. Direct evidence is encapsulated in the amount of profit the consumers make. Each art piece is sequentially appraised. This is to allow consumers to update their trust on advisors after they start appraising the next art piece.

The winner of ART competition in 2006 and 2007, Teacy et al. (2008) models the competition's domain as a multi-agent RL problem. VPI Chalkiadakis and Boutilier (2003) algorithm is adapted to this setting. Their Bayesian approach aims to approximate an optimal solution for the exploration-exploitation trade-off. The winner of the next competition in 2008 was by the agent developed by Muñoz et al. (2009) instead of an RL method, their approach introduces a simple dynamic approach that divides agents into four groups based on the knowledge degree that a decision-maker has. This is based on their multi-dimensional trust model. Similar to previous models, these only use direct evidence and witness information. The value of this metric ranges from 0 to 1. Trustees with degree 1 are denoted as *perfectly known* and therefore are exploited the most after the exploration phase. The model uses various levels of thresholds to decide these categories and phases of exploration and exploitation. For instance, in case, not enough trustees are *perfectly known*, additional rounds are allocated for exploration according to a threshold. Both these last two approaches are specifically designed for the competition. The competition was used as a test-bed in their empirical evaluations. Therefore, the results are affected by other agents' strategies. This limits their generalizability. The considerations over budgetary and usage of various trust models are not explored.

Hoogendoorn et al. (2010) propose an algorithm that looks into temporal information of a provider. According to this, decision-maker computes two trust metrics along and uses a forgetting factor: the long term $LT_i(t)$ and the short term $ST_i(t)$ trust values.

Intuitively the major difference between these types is the decay parameter γ. These metrics are defined by the varying γ values: γ is set lower for $LT_i(t)$ and higher for $ST_i(t)$. By taking the average absolute difference between these values, the estimated change $C(t)$ is calculated in each time step. If $C(t) = 0$, this means that there is no change in the behaviour of the agent. Thus, the truster acts as in *greedily*: picking the most trustworthy party. Otherwise, a selection probability value $RP_i(t)$ is calculated based on the witness information for each agent. A trustee is picked by a Monte Carlo method based on $RP_i(t)$ values.

One of the most recent *dynamic* approaches proposed by Sen et al. (2015b) uses budget-limited multi-armed bandit algorithms, BL-MAB, as a decision process. This is the only paper that explicitly considers budgetary constraints of a truster in environments where only direct interactions can be utilized. The authors improve Tran-Thanh (2012)'s work by introducing new BL-MAB algorithms for trust. They apply these algorithms for building trust in a trust-based supply chain environment. For each step in the supply chain, the corresponding decision-maker uses these algorithms to build trust. The model only considers direct evidence. These are used to compute the reward/cost ratio of each provider. This is the main criterion to decide which provider is better to pick at a particular time. In Chapter 4, we argue that witness model and any trust models with certain properties can be included to extend the generalizability of this model.

The multi-armed bandit literature includes other approaches that focus on the shortcomings of BL-MAB algorithms, which assume the cost of interacting is known beforehand. However, in domains such as real-time bidding in ad exchange and particular service provider problems, the cost of pulling an arm is discovered later. Closely related to this limitation, Ding et al. (2013) propose *Multi-Armed Bandit problems with Budget constraint and Variable costs* (MAB-BV). Here the cost of an arm is only known after the arm is pulled. A further assumption underpinning the models described so far is that interactions occur in a sequence, not in parallel. Xia et al. (2016) study an extension of MAB-BV that enables a player to pull an arm multiple times in each round. Their *multiple ratio confidence bound algorithm* (MRCB) provides better empirical and theoretical results than all other algorithms in the literature, including the algorithms designed for MAB-BF and MAB-BV. This highlights the fact that single interaction decisions enacted in a sequence are one (often unstated) assumption of existing models.

2.4.1 Learning from expert advice

The problem of *learning from expert advice* has been investigated in Online Learning. The setting that was considered is: Assume that there is a set of experts (similar to advisors in trust engagements) and several trials (i.e. occurrence of an event multiple times). These experts are known to be capable of providing predictions regarding the outcome of an event. The goal is to develop a learner (i.e. master, predictor), which

combines these predictions to generate an accurate prediction on outcomes of subsequent trials as close as possible to the *best* expert (i.e. oracle).

The halving algorithm is one of the simplest methods (Angluin, 1988). In this algorithm, the learner keeps a set of experts. With equal weight, experts vote on the outcome of the next event. Then, the set is updated after each trial. The experts that predicted the outcome successfully are kept for the prediction of the next trial. This assumes that there is at least a single expert (i.e. the best expert) that does not make any mistakes in predictions. With this assumption and assuming that more than half of the experts are accurate in their predictions, the maximum number of mistakes increases logarithmically by the number of experts. If there are no *best* experts, a *naive* approach is to reset the set (i.e. consider all agents again) when the set of experts is empty.

Limitations of the Halving algorithm are tackled with a weighted majority algorithm, also called exponential weighting (Littlestone and Warmuth, 1994). This generalizes the halving algorithm by initially setting weights of the votes to be $\frac{1}{N}$, where N is the number of experts. These weights are updated in each round and the set of experts is kept unchanged. The prediction of a trial at the time t is then computed by:

$$p_t = \frac{\sum_{i=1}^{N} w_{i,t-1} f_{i,t}}{\sum_{i=1}^{N} w_{i,t-1}} \tag{2.4}$$

where $f_{i,t}$ is the expert i's prediction and $w_{i,t-1}$ is the weight assigned to the expert. These weights are updated at each round by:

$$w_{i,t} = w_{i,t} \cdot \exp(\eta \ell(f_{i,t}, y_t)) \tag{2.5}$$

where $\eta > 0$ is the learning parameter and $\ell(f_{i,t}, y_t)$ is the preferred loss function.

The ITEA algorithm (Parhizkar et al., 2019) is a recent adaptation of the weighted majority algorithm in a trust setting. Their algorithm assumes that the set of providers and the set of advisors are disjoint (i.e. a provider agent can not have an advisory role at the same time). Initially, the weights of advisors are set to $\frac{1}{N}$. Next, the decision-maker agent queries all the advisors to receive a single binary opinion about the trustworthiness of each provider. Then, these are combined for each provider by Equation 2.4 (i.e. for each provider there is a p_t, called as $p_t(i, j)$, which denotes the prediction of consumer i about provider j at the time t). The provider with the highest $p_t(i, j)$ is selected for direct interaction. The outcome of the direct interaction then is used for updating the weights similar to Equation 2.5. They show that this method outperforms the TRAVOS and MET models, and has a smaller footprint in terms of computational time and memory.

2.4.2 Parameter optimisation

The performance of decision processes that we described above depends on the properties of the systems. For instance, the performance of multi-arm bandit algorithms vary significantly in different populations, budgets and costs. There are various ways to address this challenge; for example, a truster may select from a set of decision processes or combine the outputs of multiple decision processes. The algorithm selection method (Rice, 1976) can be employed where one from among a set of algorithms is selected on the basis of certain criteria. One common approach is "winner-takes-all" [Lazzaro et al. (1989)], in which the algorithm whose overall performance is the best on a given problem distribution.

Alpaydin (1998) combines multiple learning algorithms in order to increase the performance of the overall learning process. In automated negotiation, Ilany and Gal (2016) developed a meta-agent, which aims to predict the performance of a set of negotiation strategies based on distinctive features of domains. These features are, for instance, the size of the outcome space, the competitiveness of the scenario, degree of conflicts among preferences and so on, and accordingly adopt the strategy expected to perform best for the given scenario. This was inspired by the "*wisdom of the crowd*" principle (Surowiecki, 2005) and the "*algorithm portfolio*" approach (Leyton-Brown et al., 2003). Gunes et al. (2017) show that this approach can work well in automated negotiation. Their approach outperformed state-of-the-art negotiation agents in the recent international automated negotiating agents competition. The approach competed in International Competition Automated Negotiating Agents Competition (ANAC) 2016 and got the first place amongst ten finalist agents[1].

A model grounded on this approach can offer mechanisms in a trust-based decision-making problem is an open question. For example, before a decision-maker starts interacting with providers, an appropriate decision process can be selected, given the context: number of potential trustees, number of interactions, and overall budget. Having a set of approaches defined and mapped to domain characteristics beforehand is a major disadvantage for these method, however even if these approaches are known, their performance in new systems requires an evaluation phase to rank which one performs better than others. To this end, there are existing techniques that predict the performance of each approach based on properties of systems.

2.5 Robustness and Security

Robustness of trust and reputation systems is an important issue that has not got much attention in the trust community. As discussed trust models are getting complex, for

[1] (`http://web.tuat.ac.jp/%7Ekatfuji/ANAC2016/`)

instance: Burnett et al. (2010) incorporates stereotypical information, Goel and Faltings (2019) introduce mechanisms to encourage truthful interactions in crowdsourcing settings and Meo et al. (2017) uses topological information available in the environment. Yet, the work on investigating the robustness of the models are few. Jøsang (2012) argues:

> "Many studies on robustness in the research literature suffer from the authors desire to put their own trust and reputation system designs in a positive light, with the result that the robustness analyses often are too superficial and fail to consider realistic attacks. Publications providing comprehensive robustness analyses are rare."

Thus, the overall focus has been on accurately assessing/calculating the trust, while making weak assumptions about the cheating behaviours of the parties involved. Addressing this gap is pivotal for devising robust trust and reputation systems against attacks. For this reason, this section reviews potential attacks and existing attempts to investigate robustness in existing trust models and other related algorithms.

2.5.1 Trust and Reputation Systems

There are few reported studies that analyse robustness TRSs against realistic attacks. In fact, Granatyr et al. (2015) investigated 230 papers related to trust and reputation systems in multi-agent systems. A large proportion (68%) of these models do not assume any form of attack. The ones that look into this problem tend to, first theoretically identify the types of vulnerabilities of interest, then empirically assess the candidate models to measure the robustness against attacks. We categorize these vulnerabilities and their properties in Table 2.1.

The types of attacks that are considered in the literature consists of actions, or a combination of those actions, that an attacker or a set of attackers can take in the system. We investigated these identifications and derived their properties. Starting with simple strategies to the complex ones, the attack strategies identified are:

- *Self-promoting* (Hoffman et al., 2009), (also known as unfair ratings (Jøsang, 2012) and ballot-stuffing (Such, 2013), where an attacker by potentially creating multiple identities falsely increases a target's reputation. The target can be the attacker or another agent.
- *Slandering* (Hoffman et al., 2009; Such, 2013), (also known as unfair ratings (Whitby et al., 2004)) is where the attacker disseminates negative feedback to potential competitors.

	Controlled									
	seller(s)		advisor(s)		**Objective**			**Properties**		
Attack	I	C	I	C	LT profit	damage TRS	ST profit	SI	time	MI
Self-promoting			•	•	•					•
Slandering			•	•	•	•				•
Whitewashing	•	•	•	•	•			•		
Orchestrated	•	•	•	•	•	•	•	•	•	•
Oscilllation		•		•	•				•	
Camouflage			•	•	•					•
Value-Imbalance	•	•	•	•			•			
Reputation Lag	•	•					•			
Initial Window	•						•	•		
Exit	•						•			
Discrimination	•	•	•	•	•					•

TABLE 2.1: The properties of identified attacks for TRSs. Attackers use seller(s) or/and advisor(s) either independently (I) and/or collaboratively (C). LT is long term, and ST is short term. SI stands for switching identities, and MI for multiple identities.

- *Whitewashing* (Hoffman et al., 2009) (also known as re-entry (Kerr and Cohen, 2006, 2009) and new-comer (Wang et al., 2014)) is used by the attackers to leave the system, once the reputation of the attacker is not high enough to further profit from the malicious behaviour.

- *Sybil attacks* in peer to peer systems, where the attacker forges multiple identities to gain the majority control over the system. Kerr and Cohen (2006) identified the *cold-start* problem (also known as initial window (Kerr and Cohen, 2006, 2009)) where the attackers can make profits in the initial start of the system. Within no information available to the consumers, attackers can misbehave to make a short term profit.

- *Exit* attack is the steps that an attacker take to misbehave as long as the attacker profit from the system. Then, they leave the system.

- *Value-imbalance* attack (Wang et al., 2014) is the process of increasing the reputation of a provider with many low-cost interactions. In many trust models, the cost of the interaction is not taken into account. This makes attackers to increase their reputation, then make a large profit by having a high-cost interaction.

- *Reputation lag* (Sirur and Muller, 2019) exploits the assumption that TRSs assume the interactions are instantaneous. The other one is attackers aim to increase the reputation of their controlled providers by making many low-value non-deceitful interactions. Once the good reputation is achieved, the attackers switch to making high-value interactions with deceitful behaviour.

- *Discrimination attack* Jøsang (2012); Such (2013); Wang et al. (2014) can be combined with the attacks mentioned above can by use of other parties. In this

attack, the assumption that the reputation values are not relative to each party. Therefore, attackers create their cliques with sellers and consumers, which they only interact with the ones that they are in their clique. The increase in the reputation within the clique can be used with others who are not inside the clique later, to achieve higher gain in high value interactions.

- *Camouflage* attack (also known as dynamic unfair rating attack), Muller et al. (2016) aims to use advisors to first build reputation by being truthful initially and then acting maliciously to use the gained reputation.

- *Oscillation* attack from Srivatsa et al. (2005), in which attackers create two teams in which they switch between malicious and non-malicious behaviour. They can also incorporate the methods mentioned earlier to increase the reputation of the attackers in non-malicious behaviour.

- *Orchestrated* attacks (also known as playbooks (Jøsang, 2012) and collusion attacks (Such, 2013; Ruan and Durresi, 2016)) are the combinations of multiple attacks.

The designers of predominant TRS that we reviewed consider degrees of bias or noise in their models. The studies done by others try to answer: How vulnerable the models are under these identified attacks? Kerr and Cohen (2009) answers this question by creating strategies of attackers by materializing the given attack strategy definitions. In a simulated marketplace, the degree of vulnerability is measured by the amount of profit the attackers can achieve. They consider a *advertised-price* marketplace. A fixed set of products are sold by sellers. The prices of these items are sampled from the right portion of a Gaussian distribution where the median is 0. (The prices of items can not be lower than zero.) Each seller uniformly takes a set of products to sell. The sellers are selected from three different groups of behaviours: *honest sellers*, *random cheaters* and *cheaters*. Sellers from cheaters group select a strategy according to the experimental setting. The strategies can be in the forms of *proliferation* attack (a malicious seller creates multiple identities, thus inflating the number of products that are on sale), *reputation lag* attack, *re-entry attack* and *value-imbalance* attack. Each round of the simulation represents a day. After 14 days, consumers learn the outcome of a transaction. The amount of profit gained by cheaters from each attack and a simultaneous version of the attacks are measured and presented across five trust models. The authors point out that these attacks have defeated numerous TRSs, which include TRAVOS and BRS.

There have been studies on the theoretical aspects of some attacks. Wang et al. (2015a), for example, investigate *slandering* attacks from an information-theoretic perspective. They use *information leakage* to measure the strengths of the attacks conducted. This is a measure that they define to show how dependent a set of random variables are to each other. Information leakage is set to zero if they are independent. Based on this metric, they compute the strongest possible attacks that can be conducted by malicious advisors

(Wang et al., 2015b). They argue that the worst-case attacks in terms of *slandering* are a better measure than specific variations of the attacks. The continuation of their investigation shows the theoretical findings on the effect of initial honest behaviour of malicious advisors to gain the trust of consumers first (*reputation lag attack*) Wang et al. (2016). In addition, authors assume that any trust model that is in question in terms of its robustness treats the opinions from advisors under an information theoretical framework in their investigation. We will elaborate on this in Chapter 5

Bidgoly and Ladani (2016) provides a more complex approach, where various attack strategies are considered with a general planning mechanism (POMDP) that learns effective attack strategies through trial and error. The use of a partially observable MDP is relevant in designing a single attacker attempting to exploit an unknown TRS, where the ordering of the attacker's actions influences the outcome. It is another step into discovering the unknown space of *orchestrated attacks.* However, one of the practical issues is the search space is substantial. The considered attackers are behaving independently, whereas, in practice, it is known that the attackers incorporate other parties in their attack strategies.

Sirur and Muller (2019) provides a theoretical analysis on the *reputation lag* attack. Their analysis includes varying levels of capabilities of the attackers. These are *oracle* attackers which know the past and future transactions of the system, *eavesdropper* attackers which only know past transactions, and *blind* attackers which can see only their transactions. Within these configurations, they theoretically show that injection of a specified number of negative feedback to any TRS optimally is an NP-hard problem. Thus, their main finding is that an optimal strategy in the space of *reputation lag* attacks is not computationally feasible.

Defensive strategies against such attacks are not common in the literature. The key trust models that we presented earlier tend to not provide an explicit defensive strategy, but rather show the empirical performance of their models with the assumption that attackers exist. The ones that have a strict defensive strategy, tend to focus on putting additional mechanisms in the *dissemination* instead of the *formulation* and the *assessment* calculation of the model (Hoffman et al., 2009). Here, the *dissemination* refers to the method of sharing reputational information amongst the agents. To give an example regarding the fake identities in TRSs, such mechanisms are, for instance, making sure the agents are *unique* by associating them with cryptological and/or network-related properties (e.g. internet protocol or media access control addresses, public keys, and network coordinates to detect clusters). For the generation of false information (by the means of *self-promoting*, *slandering* and *orchestrated* attacks), the defensive strategies were based on techniques such as discounting the bad behaviour (for instance, TRAVOS (Teacy et al., 2006)), completely removing the outliers (for instance, BRS with filtering (Whitby et al., 2004)), or transforming the behaviour (i.e. utilizing the malicious behaviour to form the true distribution (Regan et al., 2006b)).

2.5.2 Adversarial Behaviours in Learning

Adversarial behaviours in *learning* and *classification* tasks is similar to the verifying robustness of models from TRSs. Although the application domain is different, there are similarities in terms of treating the models as *black-boxes*, or usage of *empirical methods* and *benchmarks*. Starting with real-world classifications, these tasks tend to require a lot of data for training. The process of collecting training data may be expensive and time-consuming. A common way to solve this problem is to acquire data sets from different sources. This can be done as an offline process, for instance, multiple workers in crowdsourcing platforms can provide a large number of training examples or an online process where on-the-fly new instances can be accepted by the users of this system. However, the disadvantage of having this procedure is that the algorithms are open to being influenced by *data poisoning* (Barreno et al., 2006). Data poisoning is a technique to strategically manipulate the training data set of a classifier to achieve an adversarial objective (Barreno et al., 2010). These attacks can target the overall performance (maximizing test error) of the system or focus on misclassification of a set of inputs.

The majority of the literature looks into the problem of devising poisoning strategies to systems for both cases that assume the training data set is completely trusted. Outside of academia, in practice, there have been reports that observe that this is not a robust design choice. For instance, Microsoft Tay, a chatbot that was designed to mimic young Twitter users, was taken offline, due to the generation of offensive comments after being poisoned by other Twitter users (Biggio and Roli, 2018). Another issue is that the provenance information of each data instance, is not taken into account or assumed not to exist. There are objective-oriented approaches: Biggio et al. (2012), for example, looks directly to SVM classifiers and tries to find an attack that decreases the overall accuracy of the classifier. There are also approaches that specifically focus on the effect of noise such as Prasad et al. (2018) improves the robustness of classifiers under a large number of outliers.

Gathering a relatively small trusted data set to improve the training of an untrusted data set by taking into account different sources can show improvements in accuracy without considering the features of the sources (Konstantinov and Lampert, 2019). Further the features of the sources can be useful for further improving the predictions by aggregating the instances from groups of other sources. This is explored by Baracaldo et al. (2017, 2018) where they provide filtering mechanisms with provenance information. In their work, they propose two algorithms: removing parties that are not behaving similarly in the trusted data set, and labelling parties that are suspicious by their contribution to the overall accuracy when the trusted data set is not available. Existing work by Koh and Liang (2017) approximates the loss function of classifiers to answer the question of how important a single data point is to predictions. Treating the classifier as a black

box, they demonstrate the means to understand the behaviours of complex prediction mechanisms. As they also point out that there are open questions in looking at subsets of the training set and understanding their effect when they are provided by multiple sources.

2.5.3 Search for Complex Attacks in Information Security

Attacks in Information Security domain are primarily related to attempts to get unauthorized access to an *asset*. (The term, asset, is used for devices, data or some component of the environment that is protected.) As we pointed out the possibility of sophisticated attacks in TRSs, this is highly relevant in this domain as well. Generally, any asset in the environment has a level of impact according to how critical it is to the system. This criticality is the key motivator for defenders to introduce defensive components, such as sensors, controls, specialized network devices (i.e. data diodes) to protect their environment. A primitive attack can be an *SQL Injection* to a server, or a *Distributed Denial of Service (DDOS)* attack. These are known to be detectable by sensors, such as *Intrusion Detection Systems* (IDS) or anomaly detection sensors. To understand and describe these primitive attacks, defenders created a shared repository which contains public databases of vulnerabilities, weaknesses, and the classifications. These databases are regularly updated according to discoveries found by specialists.

The complexity of attacks has been increasing in this domain (Navarro et al., 2018). The recent attacks tend to consist of a combination of primitive attacks, known as *multi-step* attacks. The complexity stems from a series of these attacks taking place in a target environment. To detect a multi-step attack, Dain and Cunningham (2002) points out the logical progression of these steps taken place can be an indicator. This led to methods such as looking into the casual relationships between the attacks (Salah et al., 2013). Although multi-step attacks are structurally similar to orchestrated attacks in TRSs, the applicability of these methods is limited.

The types of attacks that are closely related to our research in this domain are under "Employing Probabilistic Techniques", which as categorized by the CAPEC taxonomy[2]. These attacks utilize exploration of the security properties of the target assets, identifying and exploiting the weaknesses that are found. For instance, *Fuzzing* can be given as an example. In this attack pattern, the system is treated as a black-box and the set of inputs are searched to identify the possible insecure states of the target. While many of the search mechanisms are brute-force, there are more complex search methods that resulted in some success. For example, Godefroid et al. (2008) show this method with a tuned search algorithm was able to find vulnerabilities that were skipped by other types of tools.

[2]MITRE Corporation maintains Common Attack Pattern Enumeration and Classification (CAPEC) dataset: `https://capec.mitre.org/data/definitions/223.html`

2.6 Summary

In this chapter, we have presented an overview of the relevant literature about our research goal by looking at essential components of a trust-aware agent. We first discussed the information sources that trust models use. Second, we discussed a selection of trust models, their limitations and assumptions. Then, we reviewed the decision processes from trust literature that utilize these assessments in their decision-making. Finally, we provided the related work on adversarial behaviours and robustness studies about TRSs and other related fields. In the following chapter, we introduce the background and followed by our first contribution, budget-limited decision-making agent that utilize direct interaction and witness information sources.

Chapter 3

Background

In this chapter, we will describe a set of techniques that we exploit that underpins the contributions of subsequent chapters. We give an overview of subjective logic, which we use in Chapter 4 as a trust model that is an interchangeable component of our decision-making framework in Section 3.1. Then, we discuss our use of reinforcement learning in both of our contributions (Chapter 4 and Chapter 5) in Section 3.2.

3.1 Subjective Logic

Throughout Chapter 4, we use Subjective Logic (SL) (Jøsang, 2016), which extends probabilistic logic with uncertainties about probabilities. The main motivation in SL is to capture the idea that probabilities can have *uncertainties*. This is to enable decision-makers to capture confidence in probabilities. In this way, SL based agents are informed to make decisions to collect more evidence if there is low confidence on the probability that is assigned to an event of interest.

We explain this with *coin flipping* example. Assume that there is a *biased* two-sided coin where the probabilities are unknown and two decision-makers with the same *probabilistic* framework make a different number of trials. The first one makes 3 trials and found a probability value to be 0.33 for heads. The second one achieves the same probability with 20 trials. However, the assigned probability does not represent any confidence level in the present information that decision-maker has. When confidences are compared, second decision-maker's opinion about the probability is higher than the first one. Subjective Logic enables these assessments to be represented when they are reported and to be fused. For instance, combining both probability calculations with first and second decision-maker would yield higher confidence than the second decision-maker's opinion.

3.1.1 Dempster-Shafer Theory

SL is based on Dempster-Shafer theory (DST) (Shafer, 1976) (also known as belief model and certainty factor model (Gordon and Shortliffe, 1984)). In comparison to classical Bayesian probabilities, the belief model uses a set of exclusive possible states, rather than single events. This is to be more expressive and enables the modelling of beliefs about propositions such as "I don't know", which is not possible with classical probabilities. We provide a formal explanation of relevant parts from DST with an example to give a context for SL.

Example 1

Assume that a consumer would like to make an assessment of a cloud service provider in terms of availability. A proposition can be "The provider has high availability." or "The provider has low availability.". By these propositions, we can define two states of this problem setting: low and high availability. Let X represent all possible states, $X = \{low, high\}$, are called *frame of discernment* in DST. All possible subset of X, is denoted as 2^X, are $\{\emptyset, \{low\}, \{high\}, \{low, high\}\}$. Each element in the power set is considered as a separate representation of a possible hypothesis. Single element subsets represent "The provider has high/low availability.". The empty set represents "The provider has neither high nor low availability.". Finally, two elements represent "The provider has either (high or low) availability."

DST assigns a belief mass to each element of the power set such by a function $m : 2^X \rightarrow [0, 1]$. The masses of all the members of the power set is required to add up to 1, $\sum_{A \in 2^X} m(A) = 1$. Continuing with our example, if we set $m(\emptyset) = 0.7$, we can express the degree of uncertainty of the consumer over the availability of the provider where the sum of belief mass on other states will be $m(\{low, \text{high}\}) + m(\{\text{high}\}) + m(\{low\}) = 0.3$.

3.1.2 Binomial Opinion Representation

SL adopts the belief masses from DST to represent *opinions*. They are used for expressing beliefs from different ownerships. For instance, an agent's opinion about the provider's availability. Binomial opinions in SL are denoted under a domain, $\mathbb{X} = \{x, \bar{x}\}$ where state space has x and its complement, $\bar{x}$. The opinions are represented with a quadruplet, ω_x:

$$\begin{aligned} \omega_x &= (b_x, d_x, u_x, a_x) \\ &\textit{where } b_x + d_x + u_x = 1 \end{aligned} \tag{3.1}$$

where the parameters in an opinion represent the masses for *belief* (b_x), *disbelief* (d_x), *uncertainty* (u_x) and *base rate* (a_x). They are in the range of $[0, 1]$. Belief mass denotes the amount of support against the proposition that x is being *true*. While disbelief is the opposite of belief (x being *false*), uncertainty mass represents the notion of lack of support (i.e. insufficiency) and the base rate represents the prior probability of x without any evidence.

SL defines a projected probability of a binomial opinion (i.e. the probability expectation of an opinion) about the value x as shown in Equation 3.2. A binomial opinion can be visualized with a barycentric triangle as displayed in Figure 3.1. The vertices of the triangle are the maximum points of each type of mass. Points in this triangle represent opinions. The points (opinions) that are located in vertices are called *absolute opinions*.

$$P(x) = b_x + a_x u_x \tag{3.2}$$

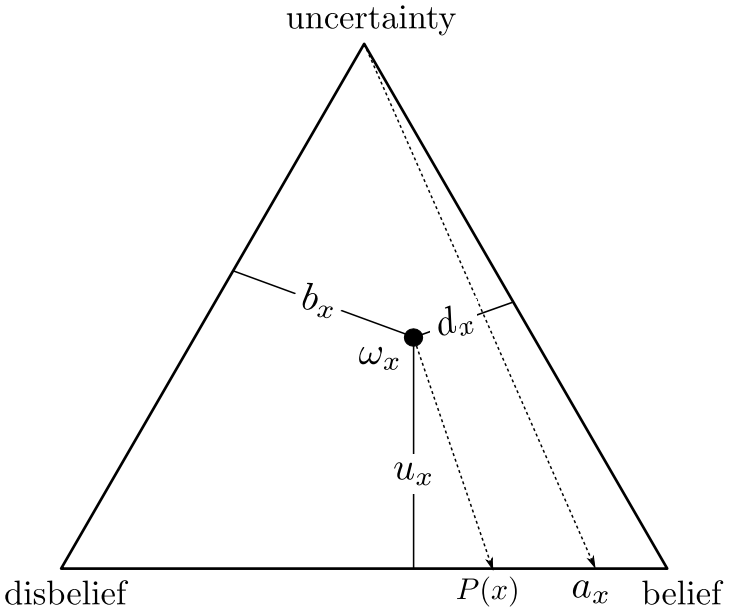

FIGURE 3.1: SL opinion triangle representation

3.1.3 The Beta Distribution

SL uses the Beta PDF (probability density function, shown in Equation 3.5) to make a mapping to binomial opinions. Assume that there is a binary event such as *coin flipping* as before, where r_x is the number of tails observed and s_x is the number of heads observed. These observations, (r_x, s_x) are mapped to the parameters of the beta distribution, α and β as in Equation 3.3.

$$\begin{aligned} \alpha &= r_x + a_x W \\ \beta &= s_x + (1 - a_x) W \end{aligned} \tag{3.3}$$

W is the prior weight which can be set to $W = 2$ to ensure a non-informative prior weight. This is similar to setting the parameters of beta distribution to $\alpha = 1$ and $\beta = 1$. The expectation of a Beta distribution computed with this mapping is:

$$E(x) = \frac{\alpha}{\alpha + \beta} = \frac{r_x + a_x W}{r_x + s_x W} \tag{3.4}$$

In Bayesian terms, this is closely related to the Beta-Bernoulli model where a random variable, X, is assumed to be distributed with Bernoulli distribution, $X \sim Ber(p)$. This is due to the fact that Beta distribution is the conjugate prior to the binomial distribution. The parameter, p, of the binomial distribution is treated as a random variable as well where p is distributed as a Beta distribution with hyper-parameters (α, β) $(p \sim Beta(\alpha, \beta))$. Therefore, Equation 3.4 is driven from the expectation of posterior predictive distribution, which is a Beta distribution with updated parameters.

$$\begin{aligned} P(x; \alpha, \beta) &= \frac{x^{\alpha-1}(1-x)^{\beta-1}}{B(\alpha, \beta)} \\ B(\alpha, \beta) &= \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha + \beta)} = \frac{(\alpha-1)!(\beta-1)!}{(\alpha+\beta-1)!} \end{aligned} \tag{3.5}$$

The PDF of uninformative prior is shown in Figure 3.2 where α and β is set to 1. This is equivalent to a uniform distribution. In SL, this is where r_x and s_x is zero and the base rate (a_x) is set to 0.5. When both parameters are equal the distribution is centred and symmetric. Uncertainty mass is high in the conflicting cases such as $\alpha = 10, \beta = 10$ and $\alpha = 3, \beta = 6$. It is lower when $\alpha = 50, \beta = 10$ as illustrated in the figure. SL, in contrast with Bayesian approaches, aggregates the data shared between agents into a compact and unified form (i.e. opinions). We use binomial opinions from SL with projected probabilities in the next chapter as a basis for our decision-making strategy. As explained in Chapter 2, we argue this trust model is interchangeable with a family of probabilistic trust models.

We extend the notation for the trust model that we implemented by introducing ownership. An opinion held by some decision maker, i, about an agent, j, regarding some issue is a tuple $\omega_{i:j} = \langle b_{i:j}, d_{i:j}, u_{i:j}, a_{i:j} \rangle$, where $b_{i:j}$ is the belief mass associated with i's view that j will succeed in the future, comparable interactions (aka. *belief*), $d_{i:j}$ is that associated with future failure (aka. *disbelief*), $u_{i:j}$ is the belief mass associated with i's *uncertainty* where $u_{i:j} = 1 - (b_{i:j} + d_{i:j})$, and $a_{i:j} \in [0, 1]$ is the prior, or base rate. The

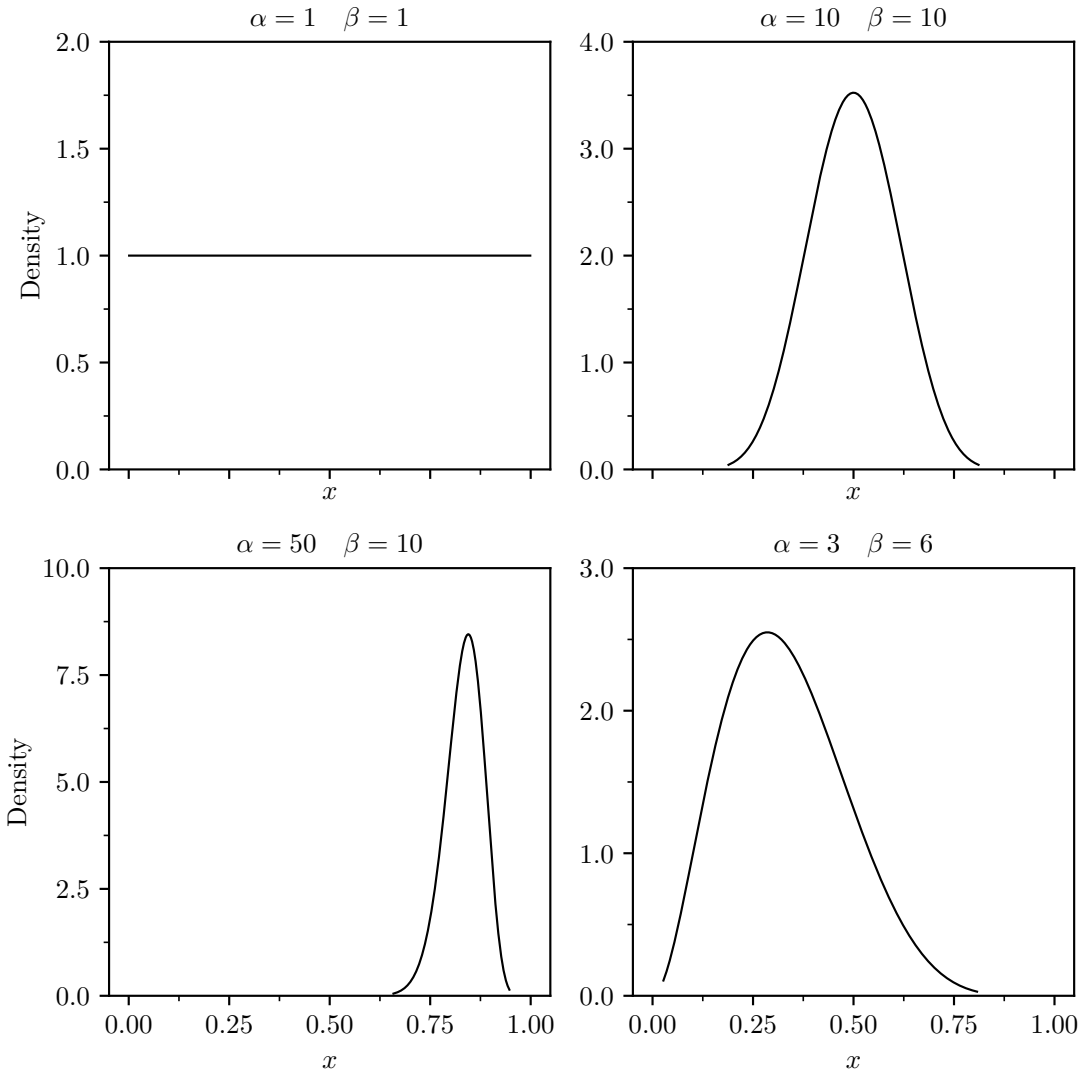

FIGURE 3.2: The PDF of beta distribution with varying α and β values

evidence used to construct binomial opinions are represented as a pair $\langle r_{i:j}, s_{i:j} \rangle$ where $r_{i:j}$ is the number of *positive* interactions that i experienced with j and $s_{i:j}$ is the number of *negative* interactions. The belief masses, $b_{i:j}$, $d_{i:j}$ and $u_{i:j}$, are computed using the formulae:

$$\begin{aligned} b_{i:j} &= \frac{r_{i:j}}{(r_{i:j} + s_{i:j} + W)} \\ d_{i:j} &= \frac{s_{i:j}}{(r_{i:j} + s_{i:j} + W)} \\ u_{i:j} &= \frac{2}{(r_{i:j} + s_{i:j} + W)} \end{aligned} \tag{3.6}$$

where W is prior weight and set to $W = 2$ to have non-information prior weight. We can

generate a single-valued, normalised trust assessment that can be used to rank and select from among individuals by distributing the uncertainty between belief and disbelief via our base rate, thus:

$$\tau_{i:j} = b_{i:j} + a_{i:j} \cdot u_{i:j} \tag{3.7}$$

Given that we consider the trust decision problem from the perspective of a single agent, we typically refer to τ_j as the trust that our decision-maker has an agent $j \in \mathcal{A}$, that r_j is the number of positive experiences our decision-maker has with provider j. Until here, Equation 3.7 is useful to measure trustworthiness from direct experiences. Opinions from advisors can be fused with a cumulative fusion function, which is simply adding up evidence parameters, $(r_{i:j}, s_{i:j})$, collected from others. There are also other ways to fuse opinions as well. For instance, In SL, opinions from others can be fused by using discounting based on our view of the trustworthiness of some witness providing opinions of others.

3.1.4 The Dirichlet Distribution

The Dirichlet distribution is the generalization of Beta distributions. Dirichlet distribution is a conjugate prior to categorical and multinomial distributions and used widely in Bayesian Learning. Equation 3.8 shows PDF of the distribution where the concentration vector ($\boldsymbol{\alpha}$) is supplied along with the input vector ($\mathbf{x}$). Various ternary plots of Dirichlet distributions where supports are 3-dimensional is shown in Figure 3.3. The darker shades in the figure represent higher density. Each black circle is a sample from the distribution. Uninformative prior setting where $\boldsymbol{\alpha} = (1, 1, 1)$ is uniform in every region. In the case of $\boldsymbol{\alpha} = (2, 2, 5)$, shaded area is larger. Thus, shows more uncertainty over the estimated probability of the categorical or multinomial distribution that is captured.

$$\begin{aligned} P(\mathbf{x}, \boldsymbol{\alpha}) &= \frac{\prod_{i=1}^{K} x_i^{\alpha_i - 1}}{B(\boldsymbol{\alpha})} \\ B(\boldsymbol{\alpha}) &= \frac{\prod_{i=1}^{K} \Gamma(\alpha_i)}{\Gamma(\prod_{i=1}^{K} \alpha_i)} \end{aligned} \tag{3.8}$$

In common with binomial opinions, SL maps the multinomial opinions to a Dirichlet Multinomial model. The major difference compared to the previous model is that masses are distributions rather than single values, with the exception of the uncertainty mass. The projected probability calculations are equal to the binomial version. Our decision-making strategy is compatible with this extension to support the family of trust models that support categorical values. SL's opinions are used under a set of operators that

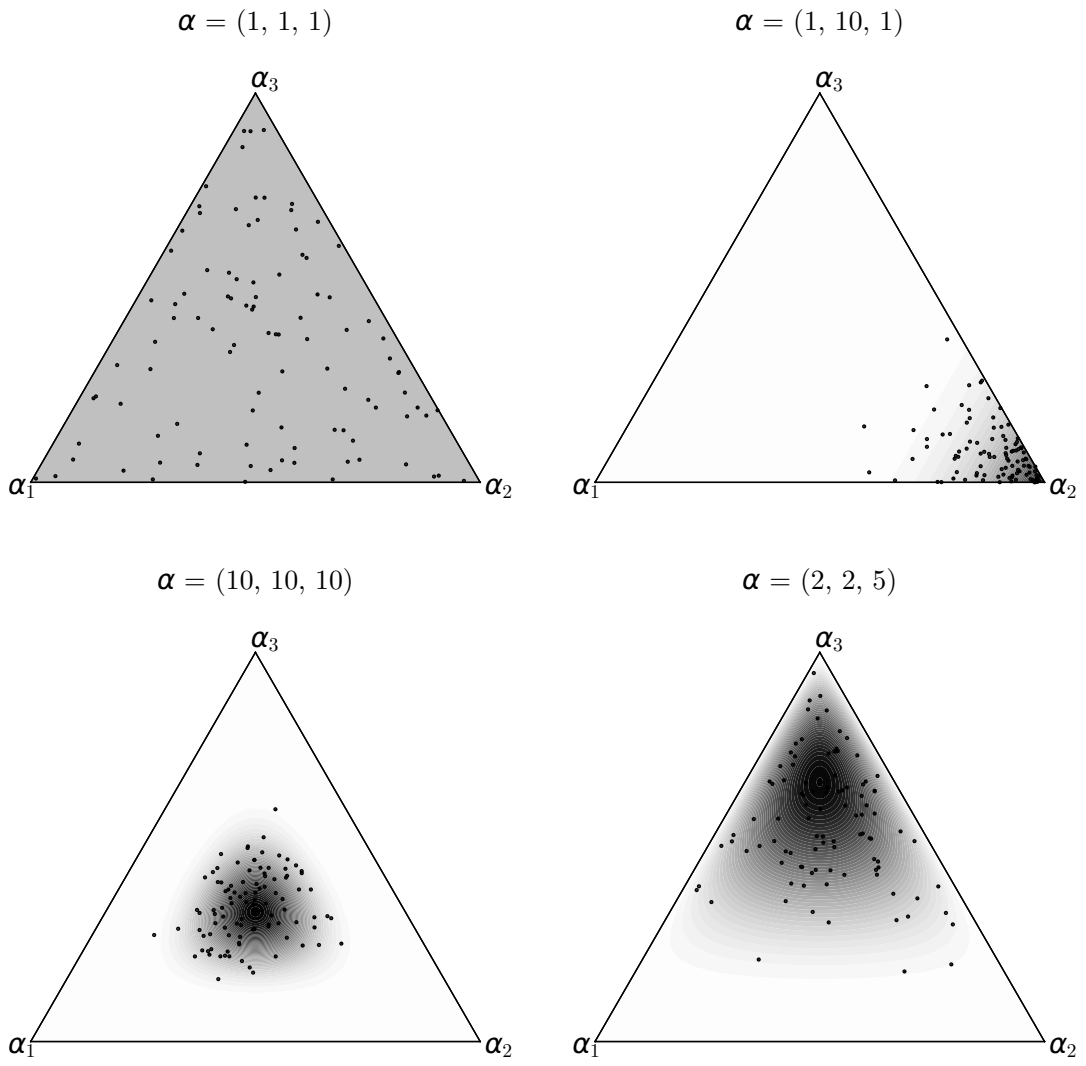

FIGURE 3.3: The ternary plots of Dirichlet distributions where supports are 3-dimensional vectors, black circles denote the samples that are taken from the distribution.

we will not be going into in thesis. We refer the reader to (Jøsang, 2016) for a more detailed description of SL and the proofs behind the algebra.

3.2 Reinforcement Learning

Reinforcement Learning (RL) is useful in the cases where a model (of the environment) is not known beforehand (Kochenderfer, 2015). Not knowing the model (i.e. unknown state transitions and rewards) requires an agent that can learn to take actions through experience. The decision-making strategy in such situations must be *dynamic*: strategy is learned while the agent interacts with the environment. Learning (making trial and

error interactions) and applying the learned knowledge are orthogonal actions. Therefore, the decision-maker agents must optimize between *exploration* versus *exploitation* actions (Sutton et al., 1998).

Throughout this thesis, we put our attention to a particular model-free learning approach: Multi-Armed Bandits (MAB) which underpins our contributions. In Chapter 4, we adapt a special case of a family of MABs to make decisions under constraints. In Chapter 5, we use Continuum-Armed Bandits (CAB) directly to find appropriate attacks that target trust models. Here, we provide an overview of these Bandit problems.

3.2.1 Multi-Armed Bandits

Multi-Armed Bandits (MAB) or K-armed Bandit(s) are one of the earliest RL approaches. This sequential decision-making problem usually includes a single state (i.e. stateless), multiple actions and unknown rewards. A typical example is given for MAB can be characterised as follows. Assume that there is a *gambler* that uses a slot machine that has K number of arms that can be pulled at each time step ($t = 1, 2, 3, ..., N$). At each time point (t) the gambler needs to decide which arm to pull given the previous plays from the time range $[0, t-1]$. The objective is to maximize the cumulative reward given the time horizon (which can be finite or infinite). We provide a selection of fundamental strategies used in selecting arms in this problem setting.

- **ϵ-greedy strategy**: ϵ portion of the time, the decision-maker explores the arms uniformly, then $1-\epsilon$ portion of the time the best rewarding arm is pulled (i.e. exploitation). The selection of when to explore or exploit is selected randomly.
- **ϵ-first strategy**: Instead of randomly selecting exploration or exploitation, this strategy divides these actions into two phases. For $N\epsilon$ number of rounds, the decision-maker pulls an arm uniformly (in some variations, the strategy is sequentially pulling all the arms). For $(1-\epsilon)N$ trials, the best performing arm is selected.
- **ϵ-decreasing strategy**: Instead of a constant ϵ, these types of strategies discount ϵ to reduce the amount of exploration throughout the decision-making. Therefore, the decision-maker is highly *explorative* initially, later the decision-maker becomes highly *exploitative*.

3.2.2 Continuum-Armed Bandits

The arms of Multi-Armed bandits are discrete, unique and usually limited. Continuum-Armed Bandits (CAB) are the problem setting where the number of arms is infinite. Assume that the decision-maker can pick any point in a real line. Each possible point is assumed to be an arm, where the reward function is stochastic. This makes CAB a

much harder problem setting than regular MABs. In Chapter 5, our considerations for a strategy to explore a space of attacks in our vulnerability analysis include CABs. We selected a hierarchical algorithm, X-armed Bandit Algorithm (Bubeck et al., 2011). We provide a brief background on this approach.

Algorithm 3.1 X-armed Bandit Algorithm (Bubeck et al., 2011)

Parameters: Two real numbers $v_1 > 0$ and $\rho \in (0, 1)$, attack space $\mathcal{X}$, all evidence $\mathcal{E}$.

Auxiliary function LEAF($\mathcal{T}$): outputs a leaf of $\mathcal{T}$

Initialization: $\mathcal{T} = \{(0, 1)\}$ and $B_{1,2} = B_{2,2} = +\infty$.

for $n = 1, 2, ...$ **do**
  $(h, i) \leftarrow (0, 1)$
  $P \leftarrow \{(h, i)\}$
  while $(h, i) \in \mathcal{T}$ **do**
    if $B_{h+1,2i-1} > B_{h+1,2i}$ **then**
      $(h, i) \leftarrow (h + 1, 2i - 1)$
    else if $B_{h+1,2i-1} < B_{h+1,2i}$ **then**
      $(h, i) \leftarrow (h + 1, 2i)$
    else
      $(h, i) \leftarrow$ choose a child randomly
    $P \leftarrow P \cup \{(h, i)\}$
  $(H, I) \leftarrow (h, i)$
  Arbitrarily choose one of arm in space $\mathcal{X}$ with respected to partition (H, I)
  $Y = \mathbb{E}[\theta_{tr \to te} | \mathcal{E}'] - \mathbb{E}[\theta_{tr \to te} | \mathcal{E}]$ ▷ Receive corresponding reward for selected arm
  $\mathcal{T} \leftarrow \mathcal{T} \cup \{(H, I)\}$
  for $(h, i) \in P$ **do**
    $T_{h,i} \leftarrow T_{h,i} + 1$
    $\hat{\mu}_{h,i} \leftarrow (1 - \frac{1}{T_{h,i}})\hat{\mu}_{h,i} + \frac{Y}{T_{h,i}}$ ▷ Update the mean $\hat{\mu}_{h,i}$ of node (h, i)
  for $(h, i) \in \mathcal{T}$ **do**
    $U_{h,i} \leftarrow \hat{\mu}_{h,i} + \sqrt{(2 \ln n)/T_{h,i}} + \nu_1 \rho^h$
  $B_{H+1,2I-1} \leftarrow +\infty$
  $B_{H+1,2I} \leftarrow +\infty$
  $\mathcal{T}' \leftarrow \mathcal{T}$
  while $\mathcal{T}' \neq \{(0, 1)\}$ **do**
    $(h, i) \leftarrow$ LEAF($\mathcal{T}'$)
    $B_{h,i} \leftarrow \min\Big\{U_{h,i}, \max\{B_{h+1,2i-1}, B_{h+1,2i}\}\Big\}$
    $\mathcal{T}' \leftarrow \mathcal{T}' \setminus \{(h, i)\}$

The algorithm divides space (i.e. real line) into regions. The divided regions are hierarchically represented by a binary tree. The nodes in binary tree is indexed with integer pairs (h, i), where the depth of a node is h and i is to denote the index of all possible nodes at the depth (in range $1 \leq i \leq 2^h$). Therefore, root node is represented as (0,1). The children of the root node is denoted as $(1, 1)$ and $(1, 2)$, in general children of (h, i) is the nodes $(h + 1, 2i - 1)$ and $(h + 1, 2i)$. Regions of $\mathcal{X}$ is associated with each node

and shown as $\mathcal{P}_{h,i} \subset \mathcal{X}$ and must satisfy the constraint, $\mathcal{P}_{h,i} = \mathcal{P}_{h+1,2i-1} \cup \mathcal{P}_{h+1,2i}$ for all $h \geq 0$ and $1 \leq i \leq 2^h$.

The statistical information that is stored in the nodes are:

- $T_{h,i}$: number of times (h, i) and its descendants are played.
- $\hat{\mu}_{h,i}$: average reward received in the region associated with (h, i)
- $U_{h,i}$: initial estimate of the region (h, i), which is sum of: $\hat{\mu}_{h,i}$,
  - $\sqrt{(2\ln n)/T_{h,i}}$: corresponding the uncertainty of rewards of the average
  - $\nu_1 \rho^h$: corresponding maximum variation of mean-payoff function in the region $\mathcal{P}_{h,i}$
- $B_{h,i}$: actual estimate of the mean-payoff function, calculated by $U_{h,i}$.

In particular, with the stored statistical information, the algorithm is progressively exploring the region in the following way. In each round, the region to play is selected by picking the node with the highest B-value (Lines 6-10 of Algorithm 3.1). The region that is selected is played and the corresponding reward is received (Lines 15-17). The tree is updated with the previous statistics given the collected reward (Lines 19-33). In detail, the path (a set of nodes) followed to select the region is updated with the reward in Lines 19-21. The tree structure of space $\mathcal{X}$, $\mathcal{T}$ is traversed and the initial estimate of subregions are updated in Lines 23-25. Initial optimistic estimates for new descendants of selected node (H, I) is set in Lines 26-27. B-values are updated in Lines 29-32.

3.3 Summary

In this chapter, we have presented the underlying techniques on which we will build our contributions in this thesis. We introduced relevant concepts from subjective logic and reinforcement learning including a discussion over our reasoning on choosing them. In the following chapter, we introduce our decision processes that use trust models in resource constrainted environments.

Chapter 4

Trust-Aware Decision Making

> Observe constantly that all things take place by change, and accustom thyself.
>
> Marcus Aurelius

In Chapter 2, we outlined the models of trust and discussed the issues and limitations when they are applied in decision-making. We found that one of the assumptions that are not relaxed is budgetary constraints that a decision-maker may have. In cases where transactions and third-party information are costly, we argue that the decision-making strategy needs to optimize towards investing in different types of information to maximize the number of trustful transactions. To achieve this, we show how this decision-making problem maps to budget-limited multi-armed bandit problems. Using the adapted algorithms that we propose, decision-making agents can explore and exploit the information gathered directly and indirectly when compared to greedy approaches.

4.1 Decision Making and Trust

Assume that a decision maker, δ, in a trust and reputation system, where the system is completely *observable* and the behaviour of each service provider is *stationary*. In this case, *observability* is an agent's knowledge about model outcomes (i.e. stochastic processes of providers). This knowledge may be modelled in terms of a metric that denotes the trustworthiness of a provider, or in terms of a preference order. From the point of a rational decision maker, this agent would interact with a provider that is the most trustworthy, or the most preferred. Formally, when decision maker is given a set of values that point towards trustworthiness of all providers, it selects one of the *strongly dominating* options (Pratt et al., 1995) in this trivial example as:

$$\tilde{p} = \arg\max_{p_i} \tau(\delta, p_i) \tag{4.1}$$

where $\tau(i, j)$ is agent i's degree of trustworthiness about agent j. The complexity of finding this provider is linear (assuming the trustworthiness of all agents are known to decision-maker in an unsorted order). The complementary addition to this simple model when the utilities are considered is as follows: each agent may have different preferences over each outcome. This could be modelled with utility theory, for instance: if the outcomes are categorical (i.e. *early*, *on time*, *late*), the cardinal utility value given by the agent of each outcome can be incorporated into decision-making as:

$$\tilde{p} = \arg\max_{p_i} \sum^{K} U(O_{\delta \to p_i}) \tau(\delta, p_i) \tag{4.2}$$

where $U(O_{\delta \to p_i})$ denotes the utility function of decision maker given the possible outcome $O_{\delta \to p_i} = k$ from provider p_i and K is the number of possible categories[1]. Subjective preference of providers (i.e. each provider changes our utility calculation) can be incorporated into the utility function in this equation. When it is not taken into account, Equation 4.2 reduces to Equation 4.1.

Until now, we assumed that there is not a cost for choosing a service provider. However, this is not necessarily the case in many domains. For instance, in e-commerce, the costs are involved while making transactions with sellers. The trust relationship involves continuous engagements between buyers and sellers. If we consider costs in the scope of an optimization problem, a buyer who wishes to maximize their utility under these costs needs to consider items that are purchased and the amount of budget available. In fact, this is a combinatorial optimization problem, a canonical *knapsack problem*. To illustrate, assume that in the environment that agent is in there are a number of items with different volumes and values. The problem is that given the available storage how many of which items to pick to have the greatest total value. The version of this problem that covers trust interactions can be considered under *unbounded knapsack problems*. These set of these knapsack problems consider the cases when a buyer can make multiple purchases of the same item (i.e. picking the same item multiple times) without limit. This addition changes the previous problem setting considerably. Although we still assume that decision-makers know the *exact* trustworthiness of everybody when transaction costs and the budget are involved, the objective now is to maximize the total utility gained overall transactions.

Following this new problem setting, instead of picking the most trustworthy agent as in Equation 4.2, the problem of the decision-maker (i.e. buyer) in this case is to consume

[1]When the domain of interactions are not categorical, the summation is switched with an integral operator in Equation 4.2.

the budget by engaging in transactions with other provider agents until the budget is depleted. (assuming that the budget is fixed at the outset.) to maximize the utility gained. We can then adapt the budget $\mathcal{B}$ into Equation 4.2. The resulting problem is that amongst all possible ways to spend this budget, the objective is to find the optimal algorithm, $\tilde{A}$, that maximizes the number of the utility gained over series of transactions:

$$\begin{aligned} \tilde{A} = \arg\max_{A} \sum_{j}^{m} N(p_i) \sum^{K} U(O_{\delta \to p_i}) \tau(\delta, p_i) \\ \text{subject to} \sum_{j}^{m} N(p_i) D(p_i) \leq \mathcal{B} \end{aligned} \tag{4.3}$$

where m is the total number of providers and $N(p_j)$ is a random variable that denotes the number of interactions made between the decision-maker agent and provider p_j. $C(p_j)$ is to cost of querying provider p_j. Total expenditure is deducted from the decision maker's budget $\mathcal{B}$ after each interaction. This is a well-studied problem, and its complexity is known to be *NP-hard* (Kellerer et al., 2004). There have been many solutions with the use of dynamic programming. Also, suboptimal heuristics are common, for instance through a greedy algorithm, where the best agent is selected throughout the decision-making.

Going back to our first problem setting, if the initial fundamental assumption is relaxed, this changes the main goal of finding the most trustworthy party when the system is not *observable*. We still assume that the behaviours of these parties are *stationary* for the time being. As we explained in detail in Chapter 2, there have been many trust models that are tackling this problem alone. These models estimate the trustworthiness metric of each party by looking into statistical correlations, opinions of others, and so on. If subjective preference of providers exists, adding the utilities into this new problem setting can be done by changing our objective function as shown in Equation 4.3. In overall, we assume a trust model exists (or selected) that provides the predictions of the behaviours of others, $\tau(\delta, p_i, k)$. Although we relaxed a fundamental assumption, another one remains. Gathering relevant information from trust systems is instantaneous. This is known as a *simplifying* assumption.

One of another known assumption is always interacting with the most trustworthy party (Yu et al., 2013a). This is in line with our starting point. However, when the assumption of decision-makers being stationary is relaxed, continuing to interact with the most trustworthy party may not be the most rational choice. Our hypothesis is that this statement is true in environments where interactions are costly, similarly to the ones we explained in Equation 4.3. When we compare two different cases: in *cost-free* environments, a natural action to do is simple. Decision-makers increase the number of

observations in the environment and reduce the uncertainty over unknown parties. Conversely, in *costly* environments, decision-makers must work within their budget, while aiming to satisfy these objectives:

- Accurately calculating the trustworthiness of others;
- Gathering relevant and significant information[2];
- Maximizing the utility (e.g. maximizing the number of positive interactions);

One possible solution is to model this problem setting is to use a more general model such as Markov decision process (MDP) (Sutton et al., 1998) or partially observable MPDs (POMDP) (Åström, 1965). To give a simple example, a direct application of MDPs to our starting problem setting is to have a single state and multiple actions where actions represent transactions with different individuals. The rewards can be modelled as the outcome of the transactions. State-transition probabilities (i.e. the probability of making a transaction with agent A after agent B) does not exist (i.e. equal to one). When we introduce uncertainty over behaviours, this mapping needs to change such that where we need to increase the number of states to have the probability estimation in our model. If we have a budget, the mapping our problem to MDPs becomes more challenging (Kochenderfer, 2015; Tran-Thanh, 2012). One direction to focus can be modelling this problem as a general multi-state Reinforcement Learning problem. However, this may lead to a dimension of a problem that may be non-existent in a trust environment in the cases where the throughout the decision-making process the preferences of the decision-makers are stationary (i.e. one-state). Throughout this chapter, we assume that the interactions made by each party do not change the state that the party is in. An example that explains this case is assuming throughout interactions the decision maker's utility function changes by considering a different outcome to be most preferred. Therefore, our focus is on the problems where the decision-maker is in a single state, which are also known as *stateless*.

Stateless (i.e.) MDPs are known under Multi-Armed Bandit (MAB) problems. When compared to the knapsack problem we introduced in Equation 4.3, we have an exploration-exploitation dilemma in our final problem setting. MABs are known to be useful in dealing with this trade-off . The *true* trustworthiness of other parties needs to be estimated progressively, meanwhile, decision-makers are interacting with others. To solve this sequential decision-making problem given the objectives above, we introduce our framework in the following section.

The remainder of the chapter is organized as follows: In Section 4.2, we define the constraints of the decision-makers may have and introduce our decision-making model.

[2]We define relevant and significant information such that it is rational for decision-makers to collect. For instance, throughout the decision-making process, if the budget spends on an advisor that deceptively reports information, we would like our model to reduce the amount of budget spend on this advisor.

We elaborate on our experimental settings and evaluate our model in Section 4.3. We discuss our results in Section 4.4, and finally, we provide a summary in Section 4.5.

4.2 A Model of Constraints

As discussed in Chapter 2, the models of trust utilize various information sources to estimate the trustworthiness of others. In our decision-making model, we relax the following simplifying assumptions made in the ongoing relationship between this estimation process and decision process: First is that the interactions are *costly*. Second is the decision-makers are constrained with a budget. The third is always selecting the most trustworthy party. Finally, the costs of each transaction to providers and the cost of the type of information varies. By doing these, we can integrate the models of trust to take the budgetary constraints into account. This means that the decision-maker can update the internal beliefs regarding the providers strategically. Our model should revert towards the processes where the previous assumptions are valid if the constraints do not exist. For instance, if collecting opinions of others is not costly, our model should not perform worse than the simple decision models. When the costs of gathering direct evidence (i.e. making interactions) are higher and gathering opinions of others (i.e. witness information) is lower, we want our model to make appropriate deviations from exploration to exploitation. An additional challenge of our decision processes is incorporating witness information into the decision-making while being under exploration-exploitation dilemma from direct evidence. As discussed in Chapter 2, trust models rely on this type of information in many cases. In the following section, we will extend the notation from Chapter 3.

4.2.1 The Problem Setting

We assume a finite set of agents where they take roles such as service providers p_j and/or consumers c_i. Throughout the lifespan of agents, $[0 : t]$, consumers interact with service providers aim to maximize their utility. Given a budget allocated to each consumer (i.e. decision-maker), we are interested in this problem: how can a decision-maker, δ, assess the given service providers and decide with whom to interact in order do maximize utility gains over time? We formally define our starting problem setting similar to Equation 4.3:

$$\tilde{A} = \arg\max_{A} \underbrace{\sum_{j}^{m} N(p_j) \sum^{K} U(O_{\delta \to p_j}) \tau(\delta, p_j)}_{G(A)} \\ \text{subject to} \sum_{j}^{m} N(p_j) D(p_j) \leq \mathcal{B} \tag{4.4}$$

We make the following assumptions, while not losing generality: utility gained after each interaction $U(O_{\delta \to p_j})$, is valued between the range $[0, 1]$. The number of providers, m, is known beforehand and constant. The behaviours of providers are stationary probabilistic (i.e. the probabilities doe not change over time). We identify each provider with a given identifier, j. The outcomes of each interaction are categorical, where the number of categories is denoted as K. In Multi-Armed Bandit (MAB) terms, our optimization problem is finding an algorithm, A, that is close to an optimal algorithm, $\tilde{A}$, maximizing our expected reward over time until $\mathcal{B}$ is exhausted for the player (i.e decision-maker), where the optimal algorithm is impossible to achieve due to exploration steps. This is also known as *regret* minimization, where we try to minimize the total regret, $R(A)$.

$$R(A) = \mathbf{E}[G(\tilde{A})] - \mathbf{E}[G(A)] \tag{4.5}$$

where the expected total reward generated by the algorithm is denoted $G(A)$. This problem is an extension of MAB problems, also known as Budget-Limited Multi-Armed Bandits (BL-MAB) (Tran-Thanh et al., 2010; Tran-Thanh, 2012). BL-MAB is, therefore, an appropriate means to model the trust decision-making process in the cases where decision-makers are interacting with multiple unknown providers under budgetary constraints. We show there is a mapping of BL-MAB to direct interactions (i.e. DI). We extend this model with actions (aka. arms) that do not return any utility gain (i.e. reward), which is needed for incorporating retrieval of witness information into the model. Unlike other approaches, we do not assume that gathering witness information is cost-free. Therefore, we would like any algorithm, A to satisfy the constraint below, that the probability of not exceeding budget is zero:

$$P\left(\sum_{j}^{m} N^{\delta}(p_j) D(p_j) + N^{\mathcal{O}}(p_j) C(p_j) \leq \mathcal{B} \right) = 1 \tag{4.6}$$

where $N^{\delta}(p_j)$ is a random variable represents the number of times the decision-maker δ interacts with p_j and $N^{\mathcal{O}}(p_j)$ is a random variable that represents the number of times the decision-maker gathers opinions about p_j from agent $\mathcal{O}$. $D(p_j)$ and $C(p_j)$ are variables that denote the costs of these actions, respectively. We consider the cases where

the cost of direct interaction is higher than witness information. If direct interaction costs less than gathering witness evidence, or $\forall j \in \mathbb{Z}^+ \; C(p_j) \leq D(p_j)$, the rational action would be to rely on collected direct evidence. This is due to the fact there is less uncertainty involved when relying on ground truth versus collected opinions and there will be a reward in the direct interactions. As discussed in Chapter 2, trust models tend to handle third parties' opinions with heuristics (such as filtering or discount) or/and exploiting correlations between similarities in what is observed versus what is collected.

4.2.2 Decision-Making

In this section, we propose a set of algorithms based on BL-MAB. First, we elaborate on these algorithms when the decision-makers are only leveraging direct evidence. We explain the algorithms in order of complexity and similarities. Second, we extend them further by allowing them to proactively use opinions from others (i.e. witness information). Throughout the algorithms, we use a stopping criterion, *the budget B_t is feasible*. This criterion, $B_t \geq \min_i C(p_i)$, is satisfied when the is remaining budget to spend. When the budget becomes not feasible for a set of providers or advisors, they are from any selection done by the algorithm.

Greedy Algorithms

The common recommendation or sometimes implicit expectation of many trust mechanisms for decision-makers is to leverage all the available information and pick the most trustworthy provider. Algorithm 4.1 shows how providers are selected when the environment only allows forming opinions from direct interaction. Initial operation of this algorithm is to first get priors from trust models. For instance, if we use Subjective Logic (SL), we derive a trust metric of each provider with the corresponding base rate a_i. Then while the budget is feasible, the algorithm selects and interacts the most trustworthy provider. Each outcome is recorded at $\mathcal{E}$. In MAB terms, this algorithm does not allocate any budget on exploration. All budget is allocated for exploitation. We introduce this algorithm since it is simple and will be useful as a baseline.

Algorithm 4.1 Greedy Algorithm - A_{greedy}

1: $t \leftarrow 1$;
2: $O_{\delta\rightarrow .} \leftarrow \{\}$;
3: **while** the budget B_t is feasible **do**
4: $\quad p_i \leftarrow \arg\max_{p_i} \frac{\tau_\delta(\mathcal{E}, p_i)}{D(p_i)}$;
5: $\quad$ *observe outcome* $O^t_{\delta\rightarrow i}$;
6: $\quad O_{\delta\rightarrow .} \leftarrow O_{\delta\rightarrow .} \cup \{O^t_{\delta\rightarrow i}\}$;
7: $\quad B_t \leftarrow B_t - D(p_i)$;
8: $\quad t \leftarrow t + 1$;

Epsilon Algorithms

These algorithms are known to follow two phases: exploration and exploitation. The budget is separated for these phases by the parameter ϵ, where $\epsilon \in [0, 1]$ and the budget for exploration is $\epsilon\mathcal{B}$ and for exploitation is the remaining budget, $(1-\epsilon)\mathcal{B}$ (Tran-Thanh et al., 2010). Throughout the exploration phase, the decision-maker uniformly interacts with service providers (i.e. uniformly pulling all arms)[3]. We introduce two variants of epsilon algorithms with same exploration phase: A_{ϵ_1} (as shown in Algorithm 4.2), where the exploitation is same as A_{greedy}, and A_{ϵ_2}, where the exploitation has done via weighted sampling. By weight, we mean the degree of trustworthiness outputted by the selected trust model. In other words, the probability of selecting p_i is proportional to $\tau_\delta(\epsilon, p_i)$. When both of these epsilon algorithms are compared, A_{ϵ_2} spends more time to trying other providers than A_{ϵ_1}, where A_{ϵ_1} only switches to other providers if the provider with the highest trustworthiness underperforms.

Algorithm 4.2 Epsilon First Algorithm - A_{ϵ_1}

1: $t \leftarrow 1$;
2: $B^{explore} \leftarrow \epsilon\mathcal{B}$;
3: $B^{exploit} \leftarrow \mathcal{B} - B^{explore}$;
4: $O_{tr\rightarrow.} \leftarrow \{\}$;
5: **Exploration phase:**
6: **while** the budget $B_t^{explore}$ is feasible **and** $A \neq \{\}$ **do**
7: *uniformly select* i and *observe outcome* $O^t_{\delta\rightarrow i}$;
8: $B^{explore}_{t+1} \leftarrow B^{explore}_t - D(p_i)$;
9: $t \leftarrow t + 1$;
10: $B^{exploit} \leftarrow B^{exploit} + B^{explore}$;
11: **Exploitation phase:**
12: Same as A_{greedy} with budget $B^{exploit}$;

We now introduce a new variant of the epsilon algorithm, which we refer to denoted as $A_{\epsilon_{\text{greedy}}}$. In this algorithm, the decision-maker alternates between exploration and exploitation phases. Instead of switching once as in previous epsilon algorithms, the phase is decided by sampling a Bernoulli distribution where the distribution's parameter is set as ϵ, in other words: $p \sim Ber(\epsilon)$. By depending on the realization of p, the phase is decided as shown in Algorithm 4.3. This is an interesting variant for showing the performance difference between when exploitation is done via after developing the model of providers (i.e. exploring the behaviours) versus the exploitation and exploration are done in parallel.

[3]There are other exploration methods, such as *Upper Confidence Bound (UCB)* exploration, however, empirical results show a minimal difference when the results of UCB is compared to uniform exploration (Tran-Thanh et al., 2010).

Algorithm 4.3 Epsilon Greedy Algorithm - $A_{\epsilon_{\text{greedy}}}$

while the budget B_t is feasible **do**
  $p \sim Ber(\epsilon)$;
  if p is True **then**
    uniformly select i and *observe outcome* $O^t_{\delta\to i}$;
    $B_{t+1} \leftarrow B_t - D(p_i)$;
    $t \leftarrow t+1$;
  else
    Same as A_{greedy} with budget B;

Filtering Algorithms

Two filtering algorithms for BL-MAB problems have been demonstrated empirical to have better performance to epsilon algorithms (Sen et al., 2015a; Karnin et al., 2013). These algorithms keep a set of potential candidates (i.e. a pool) throughout the sequential decision-making and go through a series of filtering passes. In each pass, the budget is spent on the pool, rather than all the candidates as in A_{ϵ_1} and A_{ϵ_2}. The exploitation and exploration phases are not distinct, therefore these algorithms can be seen as a version of $A_{\epsilon_{\text{greedy}}}$, where the phases are done in a reducing fashion. The number of candidates in the pool decreases throughout the decision-making. The idea is that, through this process, motivation comes from minimizing the budget spent on underperforming (i.e. malicious or dishonest) providers in the initial rounds, such that the remaining budget from these savings can be used on *known* providers. We refer to this behaviour as being *conservative*. We adapt these algorithms into our problem setting as follows: $A_{l-split}$ (as shown in Algorithm 4.4), after each pass calculates the number of providers to be stored in the next pass (Line 11). Then, the most trustworthy providers are stored (Line 12-14) and interacted with before the next pass starts (Line 4-8).

Algorithm 4.4 l-split Algorithm - $A_{l-split}$

1: $t \leftarrow 1$; $NumPasses = 0$;
2: $A^{'} = \mathcal{A}$;
3: **while** $A^{'} \neq \emptyset$; **do**
4: **for each** $i \in A^{'}$ **do**
5: *observe outcome* $O^t_{\delta\to i}$;
6: $O_{\delta\to.} \leftarrow O_{\delta\to.} \cup \{O^t_{\delta\to i}\}$;
7: $B_t \leftarrow B_t - D(p_i)$;
8: $t \leftarrow t+1$;
9: $NumPasses \leftarrow NumPasses + 1$;
10: $A^{'} = \emptyset$;
11: $PoolSize \leftarrow \left\lceil \frac{|\mathcal{P}|}{l^{NumPasses}} \right\rceil$;
12: **while** $PoolSize > 0$ **and** $\mathcal{A} \setminus A^{'} \neq \emptyset$ **do**
13: $A^{'} \leftarrow A^{'} \cup \{\arg\max_{i\in\mathcal{A}\setminus A'} \frac{\tau_\delta(\mathcal{E},p_i)}{D(p_i)}\}$;
14: $PoolSize \leftarrow PoolSize - 1$;

Instead of filtering with a number of passes, a threshold can be used to filter the pool of candidate providers. Survival of the Above Average (SOAAv) is the realization of this heuristic, wherein our case, the average trust-cost ratio is calculated and compared with a preset threshold. Depending on the changes to this metric, the candidates who are below the threshold a. SOAAv starts uniformly sampling all the providers to initialize the final threshold (i.e. $PassAverageRatio$) (Line 3-9). Then, average utility-reward ratio amongst all interactions is calculated (Line 11). Later, depending on the trustworthiness of the provider, the provider is selected according to $(1+x) * PassAverageRatio$. This parameter x is simply for tuning the threshold as desired. For instance, $x = 0$ denotes allowing agents above the threshold. This parameter is left for the implementer to tune according to the environment: behaviours of providers and costs associated.

Algorithm 4.5 *SOAAv* Algorithm - A_{SOAAV}

1: $t \leftarrow 1; A^{'} = \mathcal{A}$;
2: **while** the budget B_t is feasible **do**
3: $NumPullsInPass = 0$; $passAverageRatio = 0$;
4: **for each** $i \in A^{'}$ **do**
5: *observe outcome* $O^t_{\delta\rightarrow i}$;
6: $O_{\delta\rightarrow .} \leftarrow O_{\delta\rightarrow .} \cup \{O^t_{\delta\rightarrow i}\}$;
7: $B_t \leftarrow B_t - C(p_i)$;
8: $NumPullsInPass \leftarrow NumPullsInPass + 1$;
9: $PassAverageRatio \leftarrow PassAverageRatio + \frac{\tau_\delta(\mathcal{E},p_i)}{D(p_i)}$;
10: **if** $NumPullsInPass > 0$ **then**
11: $PassAverageRatio \leftarrow \frac{PassAverageRatio}{NumPullsInPass}$;
12: $A^{'} = \emptyset$;
13: **for each** $i \in \mathcal{A}$ **do**
14: **if** the budget is feasible and $\tau_\delta(\mathcal{E}, p_i) \geq (1+x) * PassAverageRatio$ **then**
15: $A^{'} \leftarrow A^{'} \cup \{i\}$;

We adapted these algorithms to be *discrete* BL-MAB algorithms, in which rewards are categorical. After every interaction (i.e. pulling an arm), we calculate a decision metric for all algorithms, $\frac{\tau_\delta(\mathcal{E},p_i)}{D(p_i)}$. Epsilon algorithms use this metric to sort the candidate providers in the exploitation phase. Filtering algorithms use this to choose the members for the candidate pool. This metric enables having other trust models within the algorithms introduced earlier. Until now, we showed how the direct information is collected by these algorithms. We assumed witness information is *cost-free* and available to the decision-maker in some form, formally denoted as $\mathcal{E}$. In addition, we specifically showed how we adapted these algorithms which are known to be (theoretically A_{ϵ_1}) empirically better than other possible MAB algorithms. In the following section, we show how to take into account opinions from others and the costs of these actions.

Witness Information

The opinions from others (i.e. witness information (WI)) is known to be useful to bootstrap trust assessments of providers that are unknown to a decision-maker. As discussed in Chapter 2, models often use this information exploiting correlations between direct observations and other decision-makers' opinions. We denote opinions from others in a similar way to the outcomes of direct interactions: $O^{0:t}_{c_i \to p_j}$, are the collected opinions from c_i about p_j after t rounds. We assume when c_i has any opinions, these may be transformed by the c_i, specifically for the decision-maker δ. In our model, we allow each reported opinion $O^{t}_{c_i \to p_j}$ to be categorical, rather than binary. Although, this does not restrict the usage of continuous or binary opinions.

The proposed algorithms are useful in the cases where *first-person* observations (i.e. $O^{0:t}_{\delta \to .}$) are only available. However, when *third-party* opinions exist, we can this information can be using *exploration budget* on witness information rather than sampling the behaviour of unknown providers. This is straightforward with Epsilon algorithms, where there is a clear separation of the budget, and our earlier work (Güneş et al., 2017) presented results on this idea for A_{ϵ_1} and A_{ϵ_2}. The exploration budget was spent solely on witness information gathered (as shown in Section 4.2.2), and we assumed a single *Oracle* agent which provides this information.

Algorithm 4.6 Epsilon Algorithms with Witness Information

$B^{explore} \leftarrow \epsilon B$;
$B^{exploit} \leftarrow B - B^{explore}$;
$O_{\delta \to .} \leftarrow \{\}$;
Exploration phase:
$A \leftarrow \mathcal{A}$;
while the budget $B^{explore}$ is feasible **and** $A \neq \{\}$ **do**
    uniformly select i from A;
    $O_{\delta \to .} \leftarrow O_{\delta \to .} \cup \{O^{0:t_{\mathcal{O}}}_{\mathcal{O} \to i}\}$;
    $B^{explore}_{t+1} \leftarrow B^{explore}_{t} - D(p_j)$;
    $A \leftarrow A \setminus \{i\}$;
    $t \leftarrow t + 1$;
$B^{exploit} \leftarrow B^{exploit} + B^{explore}$;
Exploitation phase:
Invoke Epsilon algorithm's exploitation phase with remaining budget $B^{exploit}$;

Filtering algorithms that switch between exploration and exploitation phases do not distinctively parametrize the proportions of budget to be spent on each phase. Therefore, the approach is taken earlier in Section 4.2.2 requires more changes in *Filtering* algorithms. Besides, there is a potential that the budget might be spent unnecessarily on witness information if the certainty of the assessments of providers is high. To take this into account and generalize how we retrieve and integrate opinions from others, we propose Algorithm 4.7. The idea is to gather opinions about the unknown providers (i.e. the decision maker's belief is uncertain about a provider), rather than uniformly

sampling as we previously proposed in Section 4.2.2. The aim is to reduce unnecessary exploration steps, and hence the budget expenditure, which can be used for exploitation steps. We assume a trust model that can be used to compute an uncertainty measure of each provider; we will elaborate on this formally in the following section. The heuristic that we propose is compatible with both Epsilon and Filtering algorithms. Whenever the decision-maker decides to interact with a provider, Algorithm 4.7 is invoked and opinions about this provider are gathered.

Algorithm 4.7 Retrieval of opinions with respect to an uncertainty threshold, h

1: **Before interacting with provider**, i
2: Calculate uncertainty mass u_i from $\tau_\delta(\mathcal{E}, p_i)$;
3: **if** $u_i > h$ **then**
4: Retrieve opinions same as in Line 8-11 from Section 4.2.2;

A key limitation with this approach is we assume a single advisor, which can provide opinions of others. In the case of multiple advisors (i.e. witnesses), making decisions about which advisors to trust to get more information about the provider of interest is more challenging. In this case, decision-makers may need to avoid, discount or transform opinions of deceptive advisors throughout the decision-making process. A common to compare direct evidence with opinions from advisors. A number of trust models discussed in Chapter 2 use different techniques so that these opinions can be used in the assessment process (i.e. calculating trustworthiness of an advisor: $\tau_\delta(\mathcal{E}, a_i)$). Although we have a process for deciding when to gather opinions about a provider, we now need to address the question from whom we should seek opinions. This problem can also be modelled as a BL-MAB problem. Assume the budget is distributed between direct interaction (DI) and witness information (WI). We formalize this problem as:

$$\begin{aligned} \tilde{A}_{\text{sub}} = & \arg\min_{A_{\text{sub}}} u_i \\ & \text{subject to} \sum_{j}^{\mathcal{W}} N^{w_j}(p_i) C^{w_j}(p_i) \leq \mathcal{B}^{\text{WI}} \tilde{\tau}_\delta(p_l) \end{aligned} \tag{4.7}$$

The reward in this sub-problem that decision-makers are looking for is minimizing total uncertainty over the provider, p_j, that decision-maker is interested in until the budget allocated for witness information is exhausted. This is to help with the main BL-MAB problem, which is making direct interactions. We invoke A_{sub} only if a decision maker's uncertainty about the expected performance of provider i is above the threshold, h. The budget allocated for all invocations of A_{sub} is distributed amongst providers based on their normalized trustworthiness: $\tilde{\tau}_\delta(p_l) = \frac{\tau_\delta(\mathcal{E}, p_i)}{\sum_l \tau_\delta(\mathcal{E}, p_l)}$. Our intuition is to allocate more budget to the more trustworthy providers to minimize the uncertainty assigned over them.

To give a concrete example on how this model is integrated with DI model, we first divide the total budget into two portions: $\mathcal{B}^{\text{WI}}$ is for witness information and $\mathcal{B}^{\text{DI}}$ is for direct interactions. If the BL-MAB algorithm, A, for DI decides to get witness information based on Algorithm 4.7, then another BL-MAB algorithm, A_{sub}, is invoked with a portion of $\mathcal{B}^{\text{WI}}$: $\mathcal{B}^{\text{WI}}\tilde{\tau}_{\delta}(p_l)$. At this point, A_{sub} uses the given budget selectively on witnesses until uncertainty is lesser than the threshold, $u_i < h$. Finally, the remaining budget is added back to $\mathcal{B}^{\text{WI}}$. A_{sub} is compatible with the introduced BL-MAB algorithms. Only difference is that instead of $\frac{\tau_{\delta}(\mathcal{E},p_i)}{D(p_i)}$ as the comparison value in DI model, we use trust values of advisors about a provider, $\frac{\tau_{\delta}(\mathcal{E},a_i,p_j)}{C(a_i)}$. The advantage of this witness model is being able to use algorithms that previously explained to solve this sub-problem that is defined in Equation 4.7. This procedure does depend, however, on a model of uncertainty about predictions of future performance of providers.

The decision model we present is a pair, (A, A_{sub}), where we allocate a particular BL-MAB algorithm for direct interaction and for witness information. We denote this as, for instance: $(A_{\epsilon_1}, A_{l-split})$ where A_{ϵ_1} is used for direct interactions and $A_{l-split}$ is used for witness information retrievals. All the adapted algorithms we introduced can be set for A or A_{sub}. In terms of budget allocation, we divide $\mathcal{B}$ such that A has $\mathcal{B}^{\text{DI}}$ and A_{sub} has $\mathcal{B}^{\text{WI}}$.

The trust values of advisors are calculated by *probability sensitive trust-discounting* approach. Assume that there are three agents: A, B and C. A wants to interact C, however, A does not have any prior knowledge about C. A can use B's opinions about C. Calculating trustworthiness of C with B's opinions for A is in SL:

$$\tau_{A:C} = \tau_{A:B} b_{B:C} + \underbrace{(1 - \tau_{A:B} b_{B:C})}_{u_{A:B}} \cdot a_{B:C} \tag{4.8}$$

Given this trust model, calculating how much trust agent A should B relies on the availability of direct experience over B. In the cases, where this not possible, we can use the following information generated from other agents who may provide their experiences to the decision-makers. For instance, assume after A collecting B's opinion and interacts with C, A devises a difference metric that would be based on the distance between what is observed (i.e. interacting C) versus what is said by others (i.e. opinions from B). *Kullback-Leiber* (KL, i.e. relative entropy) divergence can be used to measure how far is the observed behaviour from the opinions. In the case where both distributions that are compared are equal, then KL divergence is to be zero. Numerical issues from using this metric are: First, there is not an upper bound on the maximum value of the distance. Second, the metric is not *symmetric*. Also, it is known to better suited to compare the *power* of each sample (in our case, outcomes of interactions) when compared two different distributions. Instead, we use *degree of conflict* measure from SL, that compares subjective opinions via:

$$DC(\tau_{A:C}, \tau_{B:C}) = \frac{\tau_{A:C}\tau_{B:C}}{2}(1 - u_{A:C})(1 - u_{B:C}) \tag{4.9}$$

We retain uncertainty in the opinions with this model given two different opinions to calculate the conflict between opinions. Note that $DC \in [0, 1]$ where $DC = 0$ means no conflict. We set $\tau_{A:B} = 1 - DC(\tau_{A:C}, \tau_{B:C})$. This demonstrates the notion of discounting the conflicts in the situations where A does not have any observations about B. By this, the simple model explained here can use with direct interactions and opinions from others. In each round, decision-maker is going to update the trustworthiness values of advisors and providers. Specifically in each round, $\omega_{\delta:j}$ about each agent is updated with evidence pair $\langle r_{i:j}, s_{i:j} \rangle$. Next, if a direct interaction occurs and the outcome is observed, the associated evidence pair is updated and trustworthiness of the party is calculated by Equation 3.7 with SL. If an opinion from an advisor is received: First the degree of conflict, DC, is calculated by Equation 4.9. Second, this is used in Equation 4.8 to calculate final trustworthiness of the target provider.

4.3 Evaluation

In order to evaluate the contribution of our approach in various kinds of dynamic decision-making scenarios discussed in Chapter 1, we implemented a simulation environment, where decision-makers for a set of rounds interact with other agents and collect opinions until their budget is exhausted. We present our results through a series of experiments comparing the performance of our decision-making algorithms with existing models.

In evaluating our approach, our hypotheses are:

1. *Hypothesis 1:* If budgetary constraints are present, then our decision-making model will perform better than *greedy* agents. In the cases where:
   (a) *Hypothesis 1.1:* Only direct interactions are allowed.
   (b) *Hypothesis 1.2:* Direction interactions and witness information are allowed.
2. *Hypothesis 2:* If there are no budgetary constraints does not exist, then the performance of our model will be no worse than *greedy* agents.

In the following sections, we will elaborate on our simulation environment, and present our results in the scope of our hypothesis.

4.3.1 Experimental Setup

For our experiments, we create a pool of provider agent, and a single decision making agent. We limit the number of interactions that the decision maker can initiate by the given total budget ($\mathcal{B}$). Each experiment session ends when the budget of this consumer is exhausted. Providing agents persist throughout the session. We consider only binary outcomes from direct interactions (i.e. success or failure), although our model can be easily extended to support categorical (i.e. discrete) or continuous observations. A single decision maker interacts with a number of providers. The outcomes of these interactions are either *success* or *failure*. Probability of observing an outcome from a provider is set to be: $P(O_{\delta \to p_i}) = \theta_{p_i} + noise$ where θ_{p_i} is drawn from a behaviour profile distribution. We ensure $P(O_{\delta \to p_i})$ is within the range of $[0, 1]$. Particularly, these probabilities are sampled from three provider behaviours: providers with highly uncertain behaviour $\tilde{p}_1 \sim \text{Beta}(1, 1)$, reliable providers $\tilde{p}_2 \sim \text{Beta}(100, 1)$ and unreliable providers $\tilde{p}_3 \sim \text{Beta}(1, 100)$. The number of agents which are assigned these profiles are shown in Tables 4.1 and 4.2.

Costs associated with acquiring opinions from others and making interactions with others are varied with normal distributions. Providers and advisors are paired with a cost drawn from two direct interaction cost distributions and two witness information cost distributions. Costs for witness information are: low cost profile ($\tilde{c}_1 \sim \text{N}(0.1, 0.05)$) and high cost profile($\tilde{c}_2 \sim \text{N}(0.2, 0.05)$). Similarly, cost for direct interactions are: low cost profile ($\tilde{d}_1 \sim \text{N}(3, 0.5)$) and high cost profile ($\tilde{d}_2 \sim \text{N}(150, 25)$). We restrict the values drawn from these distributions to be larger than zero for numerical stability. Regarding witness information threshold and budget, we select h to be 0.01 and $\mathcal{B}^{\text{WI}}$ to be 0.005. We elaborate on this in the discussion section.

We allow advisors to *sway* the decision-maker with making false reports about the provider queried. This is done with three witness behaviour profiles ($\tilde{w}_1, \tilde{w}_2$ and $\tilde{w}_3$). Agents from *honest* profile $\tilde{w}_1$ report their opinions without any alterations. Agents from *random* profile $\tilde{w}_2$ reports random opinions that do not rely on their own experience with the provider. Finally, agents from *dishonest* profile $\tilde{w}_3$ always report the opposite of their own experience. If witness information is not available or not used, we denote this behaviour as A^o. Additionally, we will elaborate on how advisors can be coordinated to *sway* the decision-maker in an orchestrated fashion in the following chapter.

In evaluating our model, we employ these experimental conditions:

1. *Provider behaviours:* These are selected randomly from the specified profiles.
2. *Costs assignments:* Direct interaction and witness information costs are drawn from the specified profiles.
3. *Advisor behaviours*: These are selected randomly from the specified profiles.

Homogeneous Providers		
	Reliable	*Unreliable*
Low Cost	15	15
High Cost	15	15

Diverse Providers		
	Reliable	*Unreliable*
Low Cost	0	35
High Cost	15	0

TABLE 4.1: Provider behaviour profile configurations

Homogeneous Advisors			
	Honest	*Flip*	*Random*
Low Cost	10	10	10
High Cost	10	10	10

Diverse Advisors			
	Honest	*Flip*	*Random*
Low Cost	15	0	0
High Cost	0	35	0

TABLE 4.2: Advisor behaviour profile configurations

4. *Availability of information*: We select environments where consumers can use solely direct information or direct and witness information combined.

In each condition, we compare performance of the proposed epsilon-based and filtering-based algorithms with same underlying witness information handling.

4.3.2 Results

Each provider's behaviour is sampled from the distributions mentioned before. The number of providers in the different experimental conditions are shown in Table 4.2 and Table 4.1. Shapiro-Wilk (Shapiro and Wilk, 1965) test is conducted on our results to determine the appropriate statistical significance test. We selected this test because the number of samples in our results are $N < 5000$. The results that we present in this section have been found to be normally distributed with $p < 0.05$. According to this, we selected pairwise t-test (Walpole and Myers, 2012) in our results. We elaborate in detail about the statistical significance of our results with this test. The variables of adapted BL-MAB algorithms are set the same as the original author' variables. Throughout our experiments, we conducted 3000 repetitions ($N = 3000$) in each of our experiment, unless indicated otherwise.

Hypothesis 1

Figures 4.1 and 4.2 show the number of successful interactions that a decision-maker makes in conditions 1, 2 and 4. These figures show the performance of decision-making models only with direct interaction. $A_{l-split}$ in both diverse and homogenous settings outperformed the baseline model, A_{greedy} in high budget configurations. This means that the decision-making agent with $A_{l-split}$ is able to make better trust evaluations throughout the rounds than other decision-making algorithms. The error bars represent the standard error of the mean (SEM).

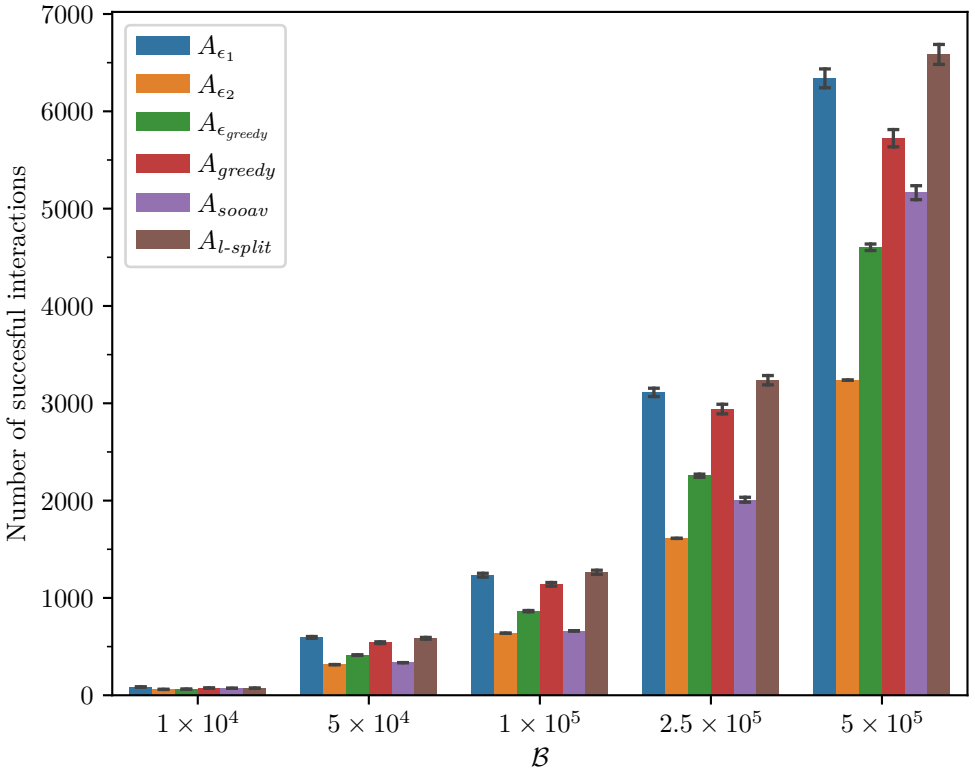

FIGURE 4.1: Diverse providers in varying budget configurations with direct interactions

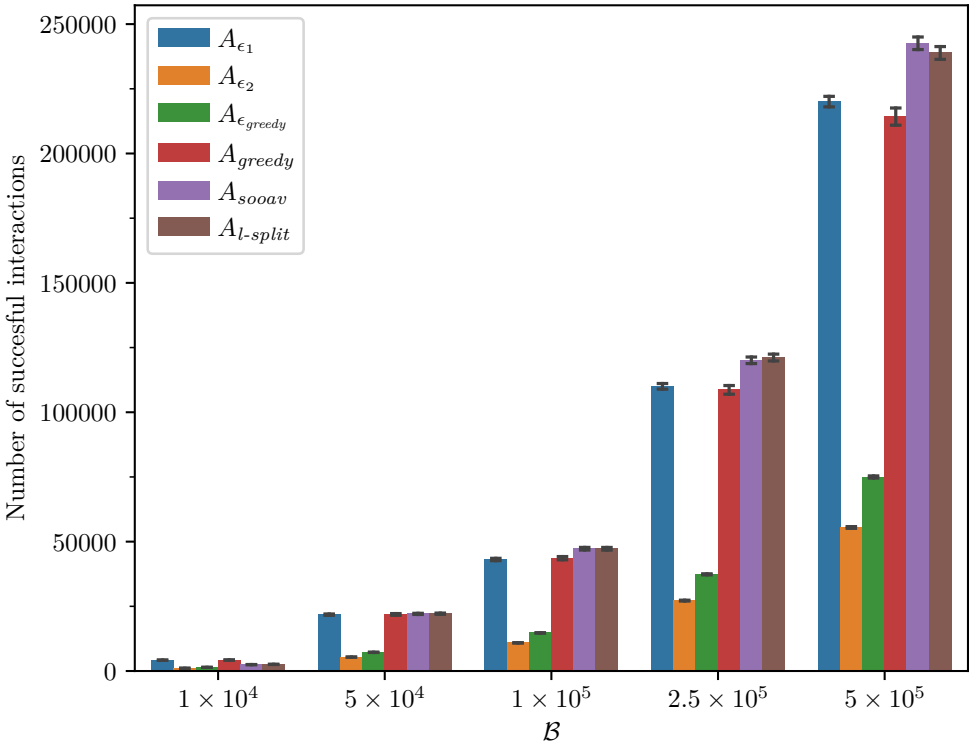

FIGURE 4.2: Homogeneous providers in varying budget configuration with direct interactions

Condition 1 and 2 when diverse provider configuration is set (Figures 4.1 and 4.2), represent the problem of having a high number of unreliable providers with low cost and a low number of reliable providers with high cost. A_{ϵ_1} and $A_{l-split}$ outperformed A_{greedy} in high budget settings. These show significant benefit to use our adapted algorithms while direct interactions in diverse populations. However, this benefit becomes minimal when the budget is reduced. A_{ϵ_2}, $A_{\epsilon_{\text{greedy}}}$ and A_{SOAAV} underperformed A_{greedy} in all budget configurations.

Condition 1 and 2 when homogenous provider configuration is set (Figure 4.2), represent the problem of having an equal number of providers from each profile. Similarly, A_{ϵ_1} and $A_{l-split}$ outperformed A_{greedy} in high budget settings. Compared to Figures 4.1 and 4.2, A_{SOAAV} also outperformed A_{greedy}. We notice that the performance of other algorithms, $A_{\epsilon_{\text{greedy}}}$ and A_{ϵ_2} performed worse in this configuration than the diverse configuration.

Witness Information with Direct Interactions

Figures 4.3 to 4.5 show the average number of successful interactions when a pair of witness information and direct interaction algorithm is used. In these figures, the budget is set to the case where the performance of A_{greedy} is similar compared to others ($\mathcal{B} = 2 \times 10^5$). Particularly, Figure 4.3 shows the performance of the algorithms when diverse providers and homogenous advisors were selected. In this experimental condition, we observe that witness information availability did not result in significant gains in the performance. However, we found to be that when witness information is collected with $A_{l-split}$ and direct interactions are made with $A_{l-split}$, $(A_{l-split}, A_{l-split})$ on average statistically outperformed the baseline pair, (A_{greedy}, A_{greedy}). This was also valid for other pairs with all other witness models where direct interaction model was set to be A_{greedy}.

Figure 4.4 shows the performance of the algorithms when diverse providers and diverse advisors were selected. In this experiment condition, the ratio of the advisors being truthful is less than the homogenous setting. The result is consistent with previous experiment condition when $(A_{l-split}, A_{l-split})$ is compared with baseline pair, on average statistically outperformed the baseline pair, (A_{greedy}, A_{greedy}). However, there is a marginal difference when $(A_{l-split}, A_{l-split})$ is compared to $(A_{l-split}, A^o)$. This behaviour is similar in terms of performance gains overall direct interaction algorithms when they are incorporating witness information into the decision-making process.

Figure 4.5 shows the performance of the algorithm when both provider and advisor profiles are homogeneous. In overall as a direct interaction algorithm, the performance of A_{greedy} is higher in this experiment condition, including when the witness information was not used. Figure 4.6 shows similar behaviour, except the marginal performance gains when witness information in cases for instance when (A_{SOAAV}, A^o) and

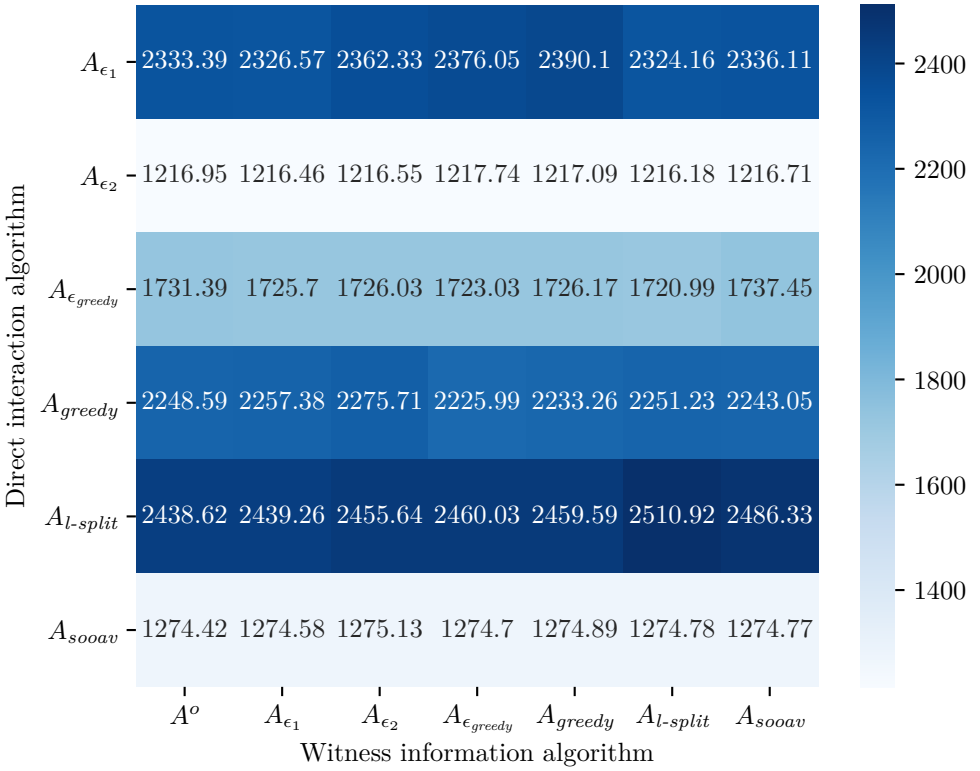

FIGURE 4.3: Diverse providers and homogenous advisors with different witness information and direct interaction algorithms

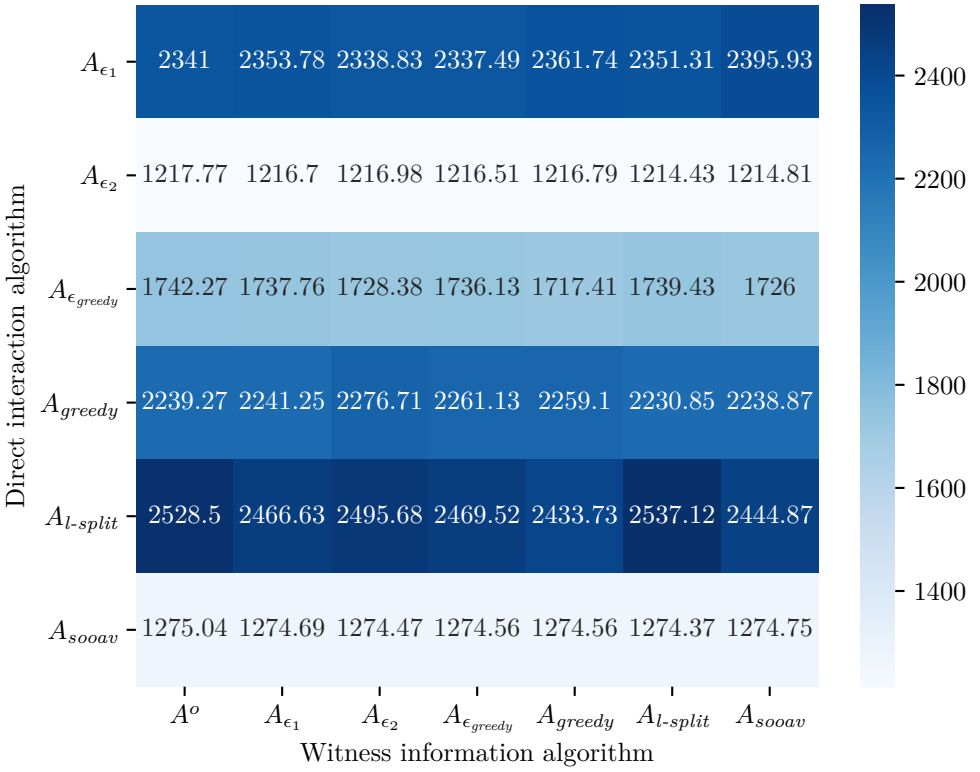

FIGURE 4.4: Diverse providers and diverse advisors with different witness information and direct interaction algorithms

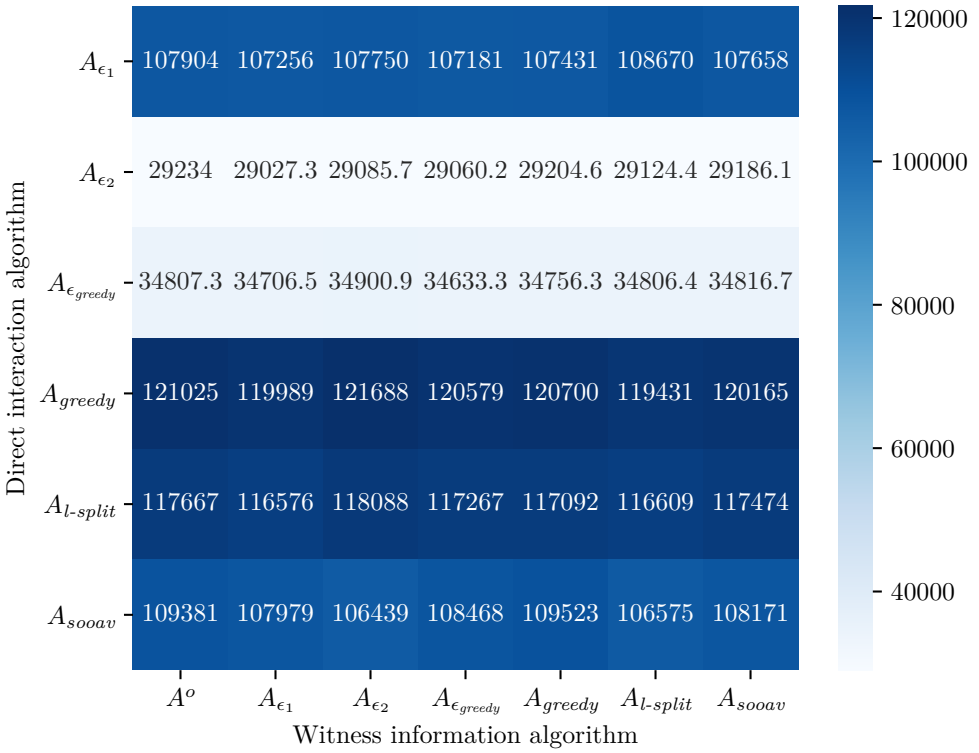

FIGURE 4.5: Homogenous providers and homogeneous advisors with different witness information and direct interaction algorithms

(A_{SOAAV}, A_{greedy}) is compared. In overall, the performance differences were minimal or no performance gains observed in this experiment condition.

Hypothesis 2

Figure 4.7 shows the performance of our algorithms with only direct interactions and homogenous providers when the costs are set to 1. A_{ϵ_1}, $A_{\epsilon_{\text{greedy}}}$, $A_{l-split}$ and A_{SOAAV} outperforms A_{greedy} as the budget (i.e. simulation time) increases. This was statistically significant when the budget is set to be $\mathcal{B} = 2.5 \times 10^5$ and $\mathcal{B} = 5 \times 10^5$. When the behaviour profiles of providers were set to be *diverse*, the results were similar. The significant difference was the higher performance of $A_{\epsilon_{\text{greedy}}}$ and A_{SOAAV} when they are compared to Figure 4.1.

Summary

To summarize, we can state the following results regarding our two hypotheses:

- *Hypothesis 1* was **supported** by our experiments: if the budgetary constraints exist, in the cases where there is only direct interaction our decision-making model

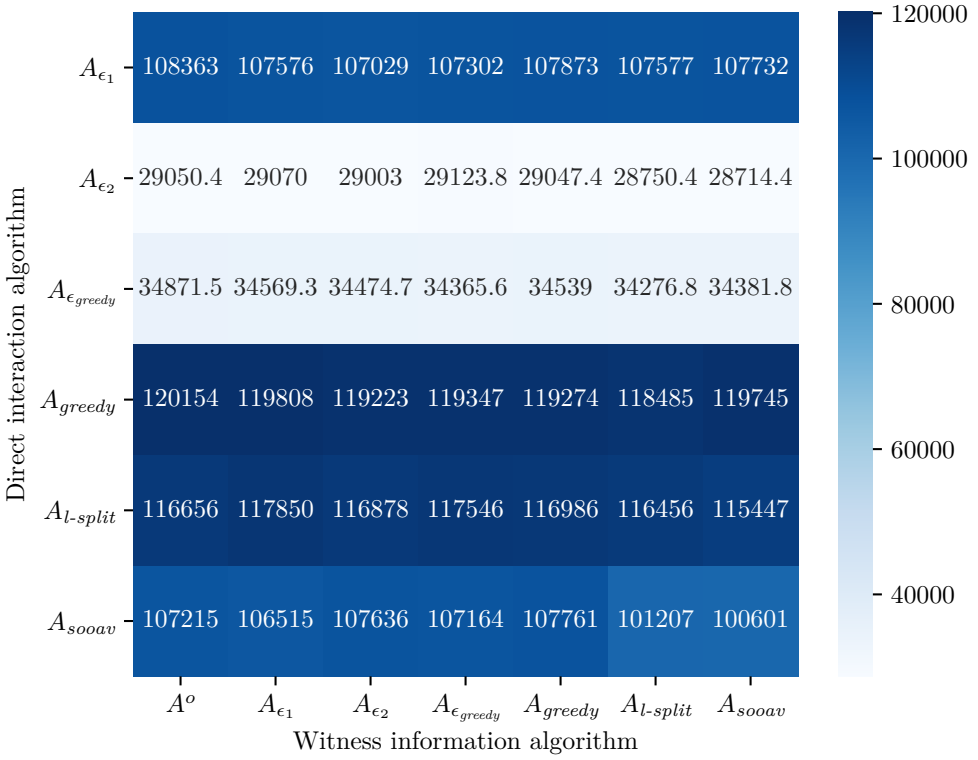

FIGURE 4.6: Homogenous providers and diverse advisors with different witness information and direct interaction algorithms

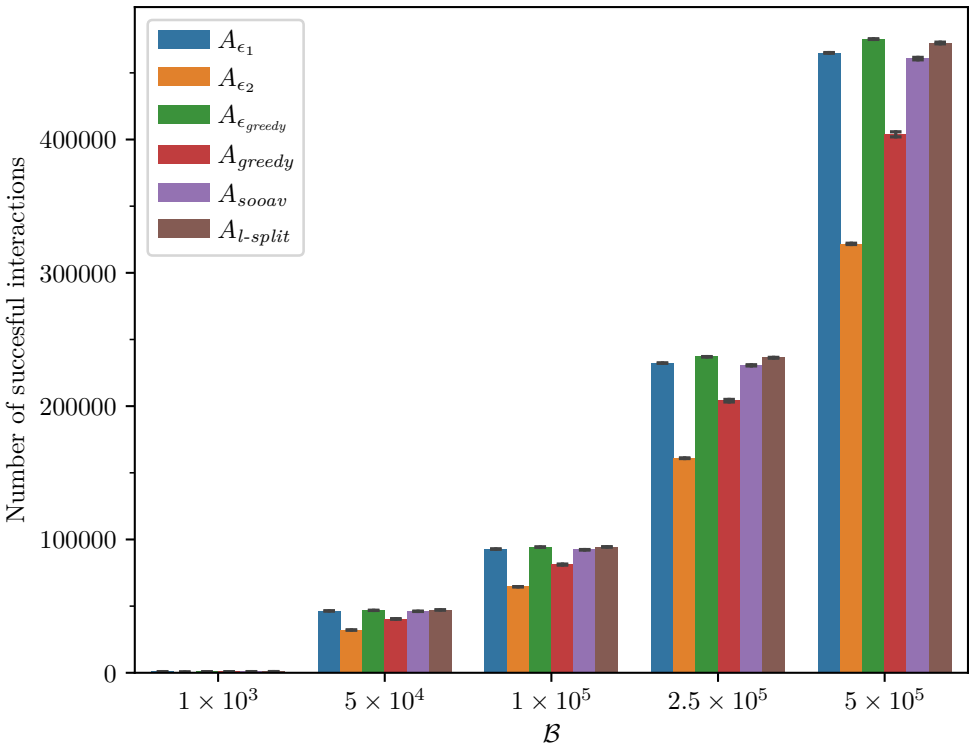

FIGURE 4.7: Homogenous providers in varying budget configuration with direct interactions, costs are set to 1.

performs significantly better than A_{greedy}. When both direct interaction and witness information exist, our decision-making model performs significantly better than (A_{greedy}, A_{greedy}) in particular settings.

- *Hypothesis 2* was **supported** by our experiments: if the budgetary constraints do not exist, then our decision-making model performs significantly better than A_{greedy}.

4.4 Discussion

Our results show that our decision-making model can offer improvements when decision-makers are under budgetary constraints. Also, we show that our model can achieve a higher number of trustworthy interactions both in information from direct interaction and witness information settings. Particularly, we observed that this performance requires a level of budget that is necessary for our model to learn about the behaviours of others. Until this level, we found out that our decision-making model performs no worse than A_{greedy}. This was the case when A_{ϵ_1} and $A_{l-split}$ was selected for direct interactions. When the witness information was included in the simulation setting, we found out that the pair $(A_{l-split}, A_{l-split})$ outperformed in all other possible pairs in diverse providers and homogenous settings.

One drawback with the results that we are presented is the underlying trust model that is used. The simplicity of the model also has an effect on the results as well. We argue that the simplicity of trust model is used was beneficial in our experimentation to put a highlight on the decision-making process rather than the behaviour estimation of providers. This helps with our generality claim where any trust model that is known to have a higher performance than is still going to benefit from performance increase in using our decision-making model.

The distribution of the budget amongst witness information and direct interaction is a challenging problem that we have not addressed in this work. We used $\epsilon = 0.1$ for direct information models, however, in our experiments this setting underperformed significantly. The further investigation is needed to find appropriate budget distribution, we identify that this relies on several factors. These are the cost distribution of the providers, the amount of available budget and the *usefulness* of witness information. We argue that the decision-maker must invest on witness information if the information is *useful*. By this, we mean that there is a statistical correlation that the selected trust model can make inferences.

4.5 Summary

In resource-constrained environments, additional mechanisms to build trust are important when decisions are made with the use of trust. As presented in our related work (in Chapter 2), the decision-making strategies often ignore such budgetary restrictions and tend to directly use the most trustworthy partner. As we have shown such strategies underperform as decision-makers miss opportunities to explore others. The approach presented here performs well in varying behavioural profiles and has a significant benefit in increasing budgetary settings.

Our usage of a standard Subjective Logic makes it compatible with any trust model that utilize sociological, direct and third-party information. Other types of information (cost-free) information that a trust model that is applied can be directly taken into account by this generalization. While we showed our decision processes' performance under a simple strategy by the use of dishonest advisors, the complex ways to influence the decision-making is not explored here. To explore this direction in the following chapter, we introduce a novel mechanism to explore attacks to trust and reputation systems.

Chapter 5

Vulnerability Analysis

> If you know your enemy and know yourself, you need not fear the result of a hundred battles.
>
> Sun Tzu

In Chapter 2, we discussed the problems posed by existence of malicious entities in trust and reputation systems. We showed that current practice in existing work is to evaluate them against known attacks by heavily relying on expert analysts. In practice, the moderation process to attacks in these systems are known to be manual and time-consuming. In addition, the complexity of attacks is expected to increase. In this chapter, we argue that vulnerabilities in trust and reputation systems can be identified in an automated manner. We formulate this problem as a black-box optimization problem in the context of online community settings and apply efficient sampling methods. In this way, we provide reliable and objective means to asses how robust trust and reputation systems are to different kind of attacks.

5.1 Adversarial Behaviours

Often, designers of trust and reputation systems rely heavily on expert analysts to evaluate their models. This gives us information about the systems when they are pitched against known attacks and their performance in limited experimental settings. As argued by Jøsang (2009; 2012), these types of analyses are limited:

> "Many studies on robustness in the research literature suffer from the authors desire to put their own trust and reputation system (TRS) designs in a positive light, with the result that the robustness analyses often are

> too superficial and fail to consider realistic attacks. Publications providing comprehensive robustness analyses are rare."

The encounters that designers of TRSs have with adversaries can be thought as a "cat-and-mouse game". Defensive mechanisms are implemented to stop or mitigate the effect of known attacks happening, meanwhile adversaries try to find new vulnerabilities. This is a pattern common to a wide range of network and system security settings. For example, injecting negative reviews for rival service providers (so called bad mouthing), or purchasing good "reviews" (ballot stuffing) are known, simple strategies. In response, TRS owners introduce controls; for example, only to permit reviews from confirmed customers. This has led to more sophisticated attacks, such as where items are purchased and then returned in order to qualify to inject negative reviews. Users may report such incidents, but the moderation process is manual, time-consuming, and may be equally used by dishonest sellers.

A concern for (near) future systems is that the complexity of attacks are expected to increase, exceeding human capabilities through the malicious use of AI algorithms (Brundage et al., 2018). TRSs may be influenced, for example, by sophisticated algorithms automating the process of finding effective combinations of attacks. As discussed in Chapter 2, the current means by which TRSs are evaluated is by assessing the accuracy of predictions across a population of simulated agents, or through the use of data sets collected from rating sites. These approaches are used to asses vulnerabilities to certain kinds of known attacks. While these methods are useful, they focus on simple attacks by single actors, eschewing the possibility of coordinated strategic attacks.

Since this chapter is concerned with how to identify vulnerabilities in trust and reputation systems, we are interested in how the search process can be automated and what can be inferred about the models that are subject to analysis. There are three main challenges that need to be addressed here:

- **Independence:** The method needs to be independent of the targeted model. In this way, the models can be compared and assessed in an objective manner.
- **Generality:** The method should, as much as possible, not presuppose the strategy of the attacker.
- **Scalability:** The method needs to scale to the numbers of participants involved in real world systems.

There are, as argued in Chapter 2, a small number of reported studies that analyse robustness of TRSs against realistic attacks (Ruan and Durresi, 2016). Furthermore, the given benchmarks are on the side of presenting the models perform in terms of accuracy in predicting the risk involved in interactions between parties. To analyse

attacks, they presume the existence of a set of known attacks (i.e specific behaviours); they lack generality and/or devised specifically for a set of models. Within the space of *all* attacks, we attempt to explore the unknown parts of the space, what an attacker can do while tackling the key challenges.

The remainder of the chapter proceeds as follows. In Section 5.2, we define our framework in terms of the trust environment, the space of attacks and our methods to search this space. In Section 5.3, we evaluate the performance of our model in simulated and real-world settings, and we conclude in Sections 5.4 and 5.5.

5.2 A Model for Vulnerability Search

Our aim is to capture the possibility of a strategic attacker that employs a coordinated attack with a specific objective. For this reason, we diverge from the general approach of starting with a model of the attacker. Instead of defining the types of attack strategy, we introduce a description of the space of possible ways to manipulate the evidence used by a TRS and propose methods for searching this space driven by a specific objective. By adopting this approach, we introduce a novel method for rigorously assessing trust and reputation systems.

5.2.1 Framework

Here, we define some notation, which aims to provide a general environment description. We first introduce a model of the environment and observations that a TRS may use as evidence, and using this we describe the space of possible attacker actions, which can be computed with strategies.

5.2.1.1 Trust Environment

For generality, we are agnostic about the specific nature of the trust model being employed, and so we characterize the trust environment in abstract terms. We assume a set of agents, $\mathcal{A} = \{a_1, ..., a_n\}$. This consists of (potentially overlapping) sets of consumers, $\mathcal{C} = \{c_1, ..., c_l\} \subseteq \mathcal{A}$ and service providers $\mathcal{P} = \{p_1, ..., p_m\} \subseteq \mathcal{A}$. Some consumers may also act as witnesses $\mathcal{W} \subseteq \mathcal{C}$. We represent opinions of agents as: A single observation at the time t is denoted as: $O^t_{c_i \to p_j}$. We assume that observations are discrete, and the number of possible values that an observation may have is bounded:

$$O^t_{c_i \to p_j} \in 1, ..., k \text{ where } k \geq 2 \tag{5.1}$$

The series of direct (or reported) observations made by a witness, $c_i \in \mathcal{W}$, of the performance of a provider, $p_j \in \mathcal{P}$, up to time t is a vector:

$$O^{1:t}_{c_i \to p_j} = (O^1_{c_i \to p_j}, ..., O^t_{c_i \to p_j}) \tag{5.2}$$

$O^t_{c_i \to p_j} = 0$ denotes that an interaction is not occurred at time t. Similarly, $O^{t:t'}_{c_i \to p_j} = 0$ denotes that c_i did not interact with p_j between the time interval t and t'. Given these environment settings, all information that is, in principle, available to form a prediction of the future behaviour of an agent (i.e. a trust assessment) at time t is, therefore:

$$\mathcal{E} = \left\{ O^{1:t}_{c_i \to p_j} \mid c_i \in \mathcal{W}, p_j \in \mathcal{P} \right\} \tag{5.3}$$

The aim here is to keep the definitions general in order to avoid restricting the applicability of our robustness analysis to other statistical trust evaluation models. Hence, the point is that the environment is characterized in terms of the fundamental (primitive) actions an attacker can take.

5.2.1.2 Decision Making

We identify a specific agent, $\delta \in \mathcal{A}$ as the decision maker. Given the evidence available, this agent needs to make assessments of the relative trustworthiness of potential providers by using a trust evaluation method. Ideally, a decision maker would have access to all the information (i.e. $\mathcal{E}$). While this is reasonable in recommender systems, this is not the case in multi-agent or peer-to-peer systems. In these settings, information that decision maker relies on can be partially observable or unreliable. Therefore, we consider decision-makers that have a partial view of the evidence available, $\mathcal{E}^\delta \subset \mathcal{E}$.

We must also consider the fact that the veracity of information available to a decision maker may vary. This is the case for *both* recommender systems and the multi-agent context, where, evidence may be misleading; i.e. a reported witness observation may differ from their real experience. Furthermore, in the multi-agent context, the veracity of each reported observation may differ for each agent collating its own viewpoint on the evidence of past interactions; e.g. $O^{0:t}_{c_i \to p_j}$ may vary among agents because $c_i \in \mathcal{W}$ provided very different witness reports to different agents. For example c_i may report that an encounter with p_j was successful to one agent and that it was unsuccessful to another. We consider the perspective of the agent that is the target of the attack (the decision maker, δ), and so $O^{0:t}_{c_i \to p_j}$ is always understood to be the observations *reported by* c_i about p_j to δ; actual observations may be missing and inaccurate ones may be added. We refer to the set of evidence available to agent δ on the basis of reported observations from other agents and its own direct experience as $\tilde{\mathcal{E}}^\delta$.

The goal of a trust assessment model is to use any observable evidence to make assessments of future performance. Assume that the decision maker uses a statistical trust model and aims to compute, for δ interested in the future performance of p_j, the probability of the outcome of the next interaction. This may be formulated as:

$$\tau(\delta, p_j, \tilde{\mathcal{E}}^{\delta}) = P(O^{t+1}_{\delta \to p_j} \mid \tilde{\mathcal{E}}^{\delta}) \tag{5.4}$$

Given that the trust evaluation method is probabilistic, the range of this function is:

$$\tau(\delta, p_j, \tilde{\mathcal{E}}^{\delta}) \in [0, 1] \tag{5.5}$$

We treat trust evaluation methods as interchangeable with other types of computational trust models. Our main reason is that while majority of the models follow probabilistic techniques (as in Equation 5.5), there are other evaluation methods that employ heuristics or combine probabilistic approaches with heuristics. Therefore, the generated value (i.e. rating) from each trust model may differ. For instance, this value can range between [-1, 1] (Marsh, 1994) or be selected from a set of categories (e.g. low trust, neutral, high trust) (Abdul-Rahman and Hailes, 2000). To satisfy our requirement for independence, however, our approach must not restrict the choice of TRS. Therefore, we treat as $\tau(\delta, p_j, \tilde{\mathcal{E}}^{\delta})$ a black-box function.

The other issue is the visibility of these values. This is not a problem for TRS owners. From an attacker's standpoint, however the implemented systems may not show the trustworthiness value of each provider. In commercial and live applications, any details about the values and their generation process tend not to be publicly provided (which is a well-known anti-pattern, "*security through obscurity*"). Users of these systems are provided with an aggregation of assessments. This can be an ordered list which shows the most trustworthy providers to the least. When this is the case, after an attack on the system, the ranking list can be observed to see the effect of an attack. This is typically a key factor defining the attacker's objective, and we elaborate on this issue in the next section.

Within a trust environment, we identify two complementary challenges:

1. The challenge for an adversary (or a set of adversaries) is to find types of attacks that significantly influence the decision maker; and
2. The challenge for a trust and reputation system is how to interpret evidence in a manner that is robust to the possibility of adversaries searching for means to exploit the system.

Addressing both challenges is necessary to develop a generalised attacker model for trust and reputation systems. Within these challenges, in the following section, we start by

identifying possible ways to influence a TRS from an attacker's perspective, then we focus on solving the problem of finding effective attack strategies.

5.2.2 Attack Space

The main goal of designing our vulnerability analysis is around the objective of *increasing ranking of a decision-making agent*. This type of behaviour can be achieved by influencing a decision maker's partial view of the evidence available (i.e $\mathcal{E}^\delta$) In terms of primitive actions, these can be a combination of some removals (e.g. $\mathcal{E}^\delta \setminus \{O^t_{w_i \to p_j}\}$), some changes (e.g. $O^{t-1}_{w_i \to p_j} = y^c$), or additions (e.g. $\mathcal{E}^\delta \cup \{O^{t:t'}_{w_i \to p_j}\}$). Removals and changes require legitimate access to the system. This is hard in practice, since TRSs tend to employ some form of authentication mechanism to prevent unauthorized access to reports generated by other parties. On the other hand, additions can be done without acquiring this level of access. These primitive actions can be combined to form an attack strategy. There are known instances of these strategies include Sybil, Whitewashing and Denial of Service attacks. Attackers can generate multiple identities and inject new reports about others to achieve their goals (aka. Sybil attacks). To avoid of consequences any malicious actions, attackers can leave or re-enter to system restore their reputations (aka. whitewashing). In addition, preventing new information entering the system (aka. denial of service) can be used by attackers to delay updates to assessments.

Within these possible primitive actions to influence the system, one of the most common classes of attack on TRSs centres on the injection of false evidence (Jøsang and Golbeck, 2009). For this reason, we primarily focus on injection of new evidence, (yet we later elaborate on the applicability of our approach given the primitive actions are different). One of the advantages of using this type of attack is that it does not require acquiring unauthorized access to the evidence available. Furthermore, the knowledge around this type of attacks are limited to basic types of attacks such as self-promoting (positive reports about the attacker) or bad-mouthing (negative reports about the competitor provider agents) (Hoffman et al., 2009). Our interest is in how to utilize such primitive strategies to form a complex one. With respect to Chapter 2, this is in the area of *orchestrated* attacks.

In this section, we provide a detailed characterisation of the means whereby an attacker can introduce false evidence. We define an attack as an injection additional witness reports to the evidence available to a decision maker. A successful attack is one for which the relative trustworthiness of the provider agents is significantly changed from the viewpoint of the decision maker, δ. If we assume that the evidence available to δ prior to the attack is $\mathcal{E}^\delta$, an attack is the introduction of $\mathcal{E}'$ so that:

$$\tilde{\mathcal{E}}^\delta = \mathcal{E}' + \mathcal{E}^\delta \tag{5.6}$$

where $\mathcal{E}'$ contains our misleading/fake reviews. We make no assumptions about the new evidence, $\mathcal{E}'$. It may be from multiple witnesses, either because it is a collaborative attack, or because an attacker can, in some way, control the generation of these reports. For this reason, identifying rewarding attacks in some context is a highly complex problem, given that the target assessment model of TRSs is unknown.

We tackle this problem by making simplifying assumptions about the space and provide our justifications for them. We consider attackers that are self-interested, and they are participating agents in TRS. The intent of these attackers is to damage overall assessments in TRS (i.e. not decreasing trust to the platform.), but to benefit from the manipulations. We restrict our attention to the trust models where the temporal information is not taken into account: $\forall O_{w_i \to p_j}, O_{w_k \to p_l} \in \mathcal{E}'$ the order which $O_{w_i \to p_j}$ and $O_{w_k \to p_l}$ are introduced into a trust assessment model does not affect the outcome. In practice, introducing crafted witness reports can be costly. These TRSs may allow reports from parties who made the observation, which may incur a cost (Ramchurn et al., 2004a; Fullam et al., 2005). For example, another cost can be acquiring/hiring identities for attacker or the cost of adding reports. This restricts the amount of new information entering to the system. We capture this condition by restricting the attacker by the number of witness reports it can affect, and there will be limits to the number of additional observations that it can inject into the system. We, therefore, investigate cases in which an attacker is limited by: (1) its *power*, or the number of observations that it can add through the attack ($\rho = |\mathcal{E}'|$); and (2) its *control* over the witnesses ($\mathcal{W}' \subseteq \mathcal{W}$).

We use weak compositions to distribute ρ amongst $\mathcal{W}'$. The intuition here is that within the limited power of an attack, ρ, the number of ways the attack can be distributed can be described by compositions. The compositions of a number denotes the number of different sequences of numbers that their summation where the reports need be partitioned in a way that benefits the attacker. Formally, a weak composition of any number n into k parts is a sequence of non-negative integers (zeros are allowed) where the sum of all values in this sequence is n. The elements in each sequence can be duplicates. In addition, sequences are not required to have unique elements. To illustrate, weak compositions of 5 into 2 parts are:

$$(5,0), (4,1), (3,2), (2,3), (1,4), (0,5) \tag{5.7}$$

The number of weak compositions (i.e. sequences) can be thought of in this following manner: Assume that there are $n + k - 1$ spaces marked on a paper. We would like to distribute n indistinguishable balls into these spaces. Each space can have either a single ball or a vertical bar. Assume that we place the first and last space a vertical bar. The number of ways to distribute the balls to this setting is: $\binom{n+k-1}{n}$. After this step, there will be $k - 1$ empty spaces. Assume that we place vertical bars to these empty

spaces. Within this helper representation, when bars are considered as cell boundaries, the number of balls inside each cell can be seen as each digit of the sequence. Therefore, the formula for the number of weak compositions n into k parts is:

$$J(n,k) = \binom{n+k-1}{n} \tag{5.8}$$

The space of possible attacks is then the weak compositions of ρ into the space in which the selected witnesses are controlled by the attacker to provide new reports. The number of possible attacks is $\mathcal{X}$, such that:

$$|\mathcal{X}| = \binom{\rho + k \cdot \left|\left\{O_{w_i \to p_j} \mid w_i \in \mathcal{W}', p_j \in \mathcal{P}\right\}\right| - 1}{k \cdot \left|\left\{O_{w_i \to p_j} \mid w_i \in \mathcal{W}', p_j \in \mathcal{P}\right\}\right|} \tag{5.9}$$

These weak compositions can be generated sequentially by the NEXTCOM Algorithm (Nijenhuis and Wilf, 2014). By using this algorithm, we use the generated weak compositions to create possible sets of fake/false reviews, $\mathcal{E}'$. In Algorithm 5.1, we start by generating of all the weak compositions limited by the power of attack (i.e. ρ) into the possible ways of ways that power of attack can be partitioned (i.e. $k \cdot |\mathcal{W}'| \cdot |\mathcal{P}|$). Each generated weak composition, v is:

$$v = (v_1, ..., v_{k|\mathcal{W}'||\mathcal{P}|}) \mid v_i \in \mathbb{N} \text{ for each } i \tag{5.10}$$

$$\sum_{i=1}^{k|\mathcal{W}'||\mathcal{P}|} v_i = \rho \tag{5.11}$$

The length of the sequences are $k \cdot |\mathcal{W}'| \cdot |\mathcal{P}|$, and the sum of all elements from the sequence sums to the attack power ρ.

The Algorithm 5.1 shows how we transform weak compositions (from Line 2) to the altered evidence $\mathcal{E}'$. Line 8 divides the vector into chunks of evidence. The algorithm outputs $\mathcal{X}$, which is the set of all possible $\mathcal{E}'$s. To illustrate, assume a weak composition created when $k = 2$, $|\mathcal{W}'| = 3$, $|\mathcal{P}| = 2$ and $\rho = 10$. In this case, the length of the sequence is 12.

$$v = (\overbrace{\underbrace{0}_{O_{w_1 \to p_1}}, \underbrace{1}_{O_{w_1 \to p_1}}}^{y=1 \quad y=2}, \underbrace{1,0}_{O_{w_1 \to p_2}}, \underbrace{2,1}_{O_{w_2 \to p_1}}, \underbrace{0,3}_{O_{w_2 \to p_2}}, \underbrace{0,1}_{O_{w_3 \to p_1}}, \underbrace{1,0}_{O_{w_3 \to p_2}})$$ (5.12)

As shown above, each two elements are mapped to controlled witness provider relationship pair. The indexing in Line 8 is used for accessing the elements of v given the

Algorithm 5.1 Generates all possible attacks from weak compositions

1: $\mathcal{X} \leftarrow \{\}$;
2: **for each** $v \in \text{NEXTCOM}(\rho, k \cdot |\mathcal{W}'| \cdot |\mathcal{P}|)$ **do**
3: $\quad \mathcal{E}' \leftarrow \{\}$;
4: $\quad$ **for each** $w_i \in \mathcal{W}^{'},\ p_j \in \mathcal{P}$ **do**
5: $\quad\quad y \leftarrow 1;\ x \leftarrow 1$;
6: $\quad\quad O^{t:t^{'}}_{w_i \rightarrow p_j} \leftarrow 0$;
7: $\quad\quad$ **while** $y \leq k$ **do**
8: $\quad\quad\quad counts = v_{(k((i-1)|\mathcal{P}|+(j-1))+y)}$;
9: $\quad\quad\quad$ **while** $counts > 0$ **do**
10: $\quad\quad\quad\quad O^{t:t+x}_{w_i \rightarrow p_j} \leftarrow y$;
11: $\quad\quad\quad\quad counts \leftarrow counts - 1;\ x \leftarrow x + 1$;
12: $\quad\quad\quad y \leftarrow y + 1$;
13: $\quad\quad \mathcal{E}' \leftarrow \mathcal{E}' \cup \{O^{t:t^{'}}_{w_i \rightarrow p_j}\}$;
14: $\quad \mathcal{X} \leftarrow \mathcal{X}\ \cup \{\mathcal{E}'\}$;
15: returns $\mathcal{X}$;

selections. In particular, each element of v corresponds to the count of new reports with their outcome y, that will be injected into the system given each selected witness-provider pair, $w_i \rightarrow p_j$. Finally, in Lines 9-13, according to the count, these new reports are added to form an attack.

To give a concrete example about the mapping, assume that Algorithm 5.1 is at a stage where the $w_2 \rightarrow p_1$ pair is selected. In addition, v is picked as shown in Equation 5.12. In this case, we look at the v_5 and v_6. v_5 corresponds to the number of new reports that will be injected with the opinion $y = 1$ by w_2 about p_1. The same is done for v_6, although the opinions are valued as $y = 2$. The opinions, $O^{t:t^{'}}_{w_2 \rightarrow p_1}$, that will be added to $\mathcal{E}'$ is, therefore, $(1, 1, 2)$.

The number of ways to inject new evidence is defined by Equation 5.9. We show a numerical example to demonstrate the growth of this function in Figure 5.1. When $|\mathcal{W}'|$ and $|\mathcal{P}|$ are reasonably large, therefore, it is not feasible to sample even a small percentage of this space. Thus, we explore a restriction on strategies that reduces this large attack space, while avoiding the imposition of designed-in attacks.

5.2.2.1 Restricted Space

The aim here is to retain the challenge for the attacker, where in any realistic scenario its search would be limited to the selection of witnesses to use in an attack, because using a witness may be costly in some contexts (e.g. cost of spoofing or bribing the witness.). In our previous attack space $\mathcal{X}$ definition, we considered incorporating all the controlled witness into our attack. This exhaustive search in terms of computational time and space is not feasible. For this reason, we turn our attention to a possible subset of (i.e.

$\tilde{\mathcal{X}} \subseteq \mathcal{X}$). Our considerations for reducing the space is: Instead of using all controlled witnesses, a subset of controlled witnesses can be selected. Then, each of these selected witness can provide a portion of the number of malicious reviews.

The procedure to generate each $\mathcal{E}'$ in this space is as follows:

1. We make a selection from controlled witnesses (i.e. the number of selections, s)
2. The attack power, ρ, is distributed across these selected witnesses:
   (a) ρ's all restricted partitions to s (i.e. $D = RP_s(\rho)$) and their the permutations without repetition: P_s^D
   (b) These permutations are then distributed to each (advisor, trustee) pair. The number of ways to distribute these permutations is $(|\mathcal{P}| \cdot k)^s$

Then, the number of attacks in this reduced space is:

$$|\tilde{\mathcal{X}}| = \binom{|\mathcal{W}^{'}|}{s} D \cdot P_s^D \cdot (|\mathcal{P}| \cdot k)^s \tag{5.13}$$

where the number of restricted partitions of ρ into s parts is:

$$RP_s(\rho) = RP_s(\rho - s) + RP_{s-1}(\rho - 1) \tag{5.14}$$

where $RP_0(0) = 1$ and $RP_s(\rho) = 0$ if $\rho \leq 0$ or $s \leq 0$. The number of additional reported observations from witnesses, ρ, is distributed across all partitions, restricted by the number of selected witnesses and the number of providers. By this reduction, each witness can provide a portion of the total malicious reviews (ρ) to a single selected provider.

To give a concrete example of creation of a single attack, let $k = 2$, $|\mathcal{W}^{'}| = 5$, $|\mathcal{P}| = 2$, $\rho = 5$ and $s = 2$: First, assume that we make a selection of witnesses, which are (w_1, w_3). Next, we select one of many restricted partitions, assume that the selection is $(3, 2)$. Then, we select an order of this partition selection $(2, 3)$ (intuitively in this case, there are only two: $(3, 2)$ and $(2, 3)$). We consider all the cases, in which w_1 provides 4 reports about all providers and likewise w_3 provides 6 reports. A single attack within this setting can be: $O_{w_1 \to p_3}^{t:t+2} = (1, 1)$ and $O_{w_3 \to p_5}^{t:t+3} = (2, 2, 2)$.

Algorithm 5.2 starts by selecting a subset of possible controlled witnesses (Line 2). The power of the attack then is divided these witnesses: generating restricted partitions of the power ρ into s parts. q denotes the set of the partitions for each witness as:

$$q = (q_1, ..., q_s) \mid q_i \in \mathbb{N} \text{ for each } i \tag{5.15}$$

Algorithm 5.2 Generating attacks for selected witnesses

1: **Input:** $(w_1, ..., w_s)$; the witnesses selected
2: $\tilde{\mathcal{X}} \leftarrow \{\}$;
3: **for each** $q \in$ GENERATERP(s, ρ) **do**
4: **for each** $r \in$ GENERATEPERMUTATIONS(q) **do**
5: $x \leftarrow 1$;
6: **while** $x \leq (|\mathcal{P}| \cdot k)^s$ **do**
7: $\mathcal{E}' \leftarrow \{\}$;
8: $val \leftarrow x - 1$;
9: **for each** $w_i \in (w_1, ..., w_s)$ **do**
10: $index \leftarrow val \ \% \ (|\mathcal{P}| \cdot k)$;
11: $j \leftarrow \frac{index}{|\mathcal{P}|}$; $counts \leftarrow r_i$;
12: $O_{w_i \rightarrow p_j} \leftarrow \{\}$;
13: **while** $counts > 0$ **do**
14: $O_{w_i \rightarrow p_j} \leftarrow O_{w_i \rightarrow p_j} \cup \{index \ \% \ |\mathcal{P}|\}$;
15: $counts \leftarrow counts - 1$;
16: $val \leftarrow \frac{val}{|\mathcal{P}| \cdot k}$;
17: $\mathcal{E}' \leftarrow \mathcal{E}' \cup \{O_{w_i \rightarrow p_j}\}$;
18: $x \leftarrow x + 1$;
19: $\tilde{\mathcal{X}} \leftarrow \tilde{\mathcal{X}} \ \cup \{\mathcal{E}'\}$;
20: returns $\tilde{\mathcal{X}}$;

Algorithm 5.3 Generating all attacks in $\tilde{\mathcal{X}}$

1: $\tilde{\mathcal{X}} \leftarrow \{\}$;
2: **for each** $(w_1, ..., w_s) \leftarrow$ SELECTWITNESSES$(\mathcal{W}')$ **do**
3: $\mathcal{X} \leftarrow \mathcal{X} \cup$ Algorithm 5.2 with $(w_1, ..., w_s)$;
4: returns $\tilde{\mathcal{X}}$;

The generation of these partitions can be done efficiently via the use of RuleAsc algorithm (Kelleher and O'Sullivan, 2014). This computation of each partition would be constant amortised time. Next, these partitions are permutated without repetition to cover all possible ways of distributing each integer (Line 4). According to the number of ways that each permutation q can be distributed (Line 6), we assign a proportion of ρ to each selected witness w_i (Line 9-11). The rest shows how this proportion is assigned as a new report for the each witness and inserted as an attack $\mathcal{E}'$ to the attack set $\tilde{\mathcal{X}}$. We generate all possible assignments of restricted partitions by using integer y. Assume that reports from each witness is a k by $|\mathcal{P}|$ matrix. In Line 10, we select a cell from the matrix and integer val represent the index of this cell. Given that the proportion of ρ is r_i for w_i and this proportion is going to be added to provider j (Line 11), we create a new set of reported opinions for the controlled witness about a provider in Line 12. Finally, we insert the new reports by integer $counts$ in Line 14. We follow the same steps until we finish iterating the tuple $(w_1, ..., w_s)$.

The space requirement of generation of attacks can be reduced by a truncated version of Algorithm 5.3, which stores a subset of attack space $\tilde{\mathcal{X}}$ for a single witness selection

$(w_1, ..., w_s)$. Accessing an attack (i.e. $\mathcal{E}'$), with the attack search mechanisms that we will introduce in this section, is not limited by this approach. To clarify, when the methods request to access an attack from $\tilde{\mathcal{X}}$ (i.e. a point in this space), we separate the space uniformly into $\binom{|\mathcal{W}'|}{s}$ regions. A point in this space can be accessed with Algorithm 5.4.

Algorithm 5.4 Truncated version of Algorithm 5.3

1: **Input:** $x \in [0, 1]$; a point in the space
2: $\tilde{\mathcal{W}} \leftarrow \{(w_1, ..., w_s) \mid (w_1, ..., w_s) \in \text{SELECTWITNESSES}(\mathcal{W}')\}$;
3: $\tilde{\mathcal{X}} \leftarrow$ Algorithm 5.2 with some arbitrary $(w_1, ..., w_s)$;
4: $witnessBinSize \leftarrow \frac{1.0}{\binom{|\mathcal{W}'|}{s}}$; $attackBinSize \leftarrow \frac{1.0}{|\tilde{\mathcal{X}}|}$;
5: $i \leftarrow \left\lfloor \frac{x - (x \ \% \ witnessBinSize)}{witnessBinSize} \right\rfloor$; $j \leftarrow \left\lfloor \frac{x - (x \ \% \ attackBinSize)}{attackBinSize} \right\rfloor$;
6: returns $\mathcal{E}'$ from $\tilde{\mathcal{X}}_i$ with witnesses changed to $\tilde{W}_j$;

We eliminate the number of attack through Algorithm 5.4. First, we create the combinations of selections of controlled witnesses we control (Line 2). Second, we create a subset of attacks with arbitrary witnesses by using Algorithm 5.2. Then, we calculate the sizes of bins for attacks and witnesses (Line 4). Later, the corresponding indices for both bins are calculated (Line 5). Finally, the attack that is selected $\tilde{\mathcal{X}}_i$ is returned with witnesses changed to $\tilde{W}_j$ (Line 6). In this way, the space complexity is reduced to:

$$|\tilde{\mathcal{X}}| = D \cdot P_s^D \cdot (|\mathcal{P}| \cdot k)^s \tag{5.16}$$

The restricted space is a subset of all possible ways to inject malicious reports to the system. As shown in Figure 5.1, when spaces compared numerically, including the selecting a subset of controlled witnesses stage in unrestricted space, we reduce the number of possible ways. In addition, during implementation, the restricted space size can be further reduced in Algorithm 5.4 with a factor of $1/\binom{|\mathcal{W}'|}{s}$. The constraint that we take into account in our model, where the attacker has a certain power (i.e ρ) increases the complexity of distributing the power. This requires generation of all attacks in the selected space, before approaching to the problem of finding which attack is more rewarding than others. In addition to this drawback, objective that the attacker want to achieve may or may not be achievable with the given power. We elaborate on these drawbacks and our solutions in Section 5.2.3.

Example Attack

To illustrate the kinds of attack within this space, and the potential effect of an attack on the relative trustworthiness of the agents from the perspective of the target, δ, consider Figure 5.2. Here, we have five providers, $\{p_1 \dots p_5\} \in \mathcal{P}$ and five witnesses, $\{c_1 \dots c_5\} \in \mathcal{W}$, the attacker has power, $\rho = 5$, and it has control over (and/or has chosen) witnesses

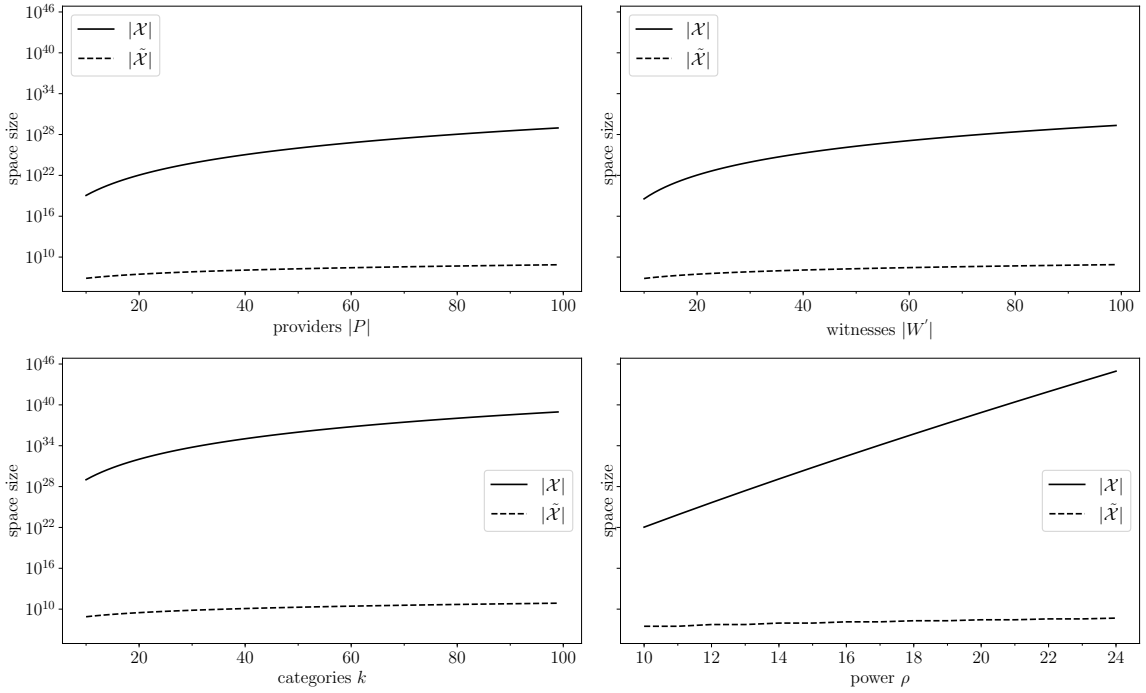

FIGURE 5.1: Log-lin plot of the comparison between attack spaces with all possible ways to attack to system with system parameters, unless varied, being $\rho = 10, k = 2, |W| = 20, |W^{'}| = 20, s = 2, |P| = 20$

c_1, c_3 and c_4 through which to target its attack. The aim is to improve the relative position of provider p_1 from the perspective of the decision maker, δ.

In Figure 5.2, we show the ranking of each provider, $r(p_i)$, before and after the attack, where this ranking is based on the trustworthiness of each provider computed using a beta distribution on the basis of positive $(+1)$ and negative (-1) observations reported by our witnesses. The detail of the attack is:

1. The attacker injects one positive rating from witness c_1 regarding p_1, increasing c_1's overall view of p_1 to $+2$.
2. It injects one negative rating from witness c_3 to p_3, reducing c_3's overall view of p_3 down by -1.
3. Finally, it injects three negative ratings from witness c_4 regarding p_5, dropping this from $+1$ to -2.

Note how the attack is distributed against all competitors to p_1's relative position.

We intentionally chose a simple trust model in this example, but it serves to highlight the kinds of attack that may be identified through our model. The questions that now remain are: what is an optimal attack, and how do we discover such attacks efficiently given the size of the search space?

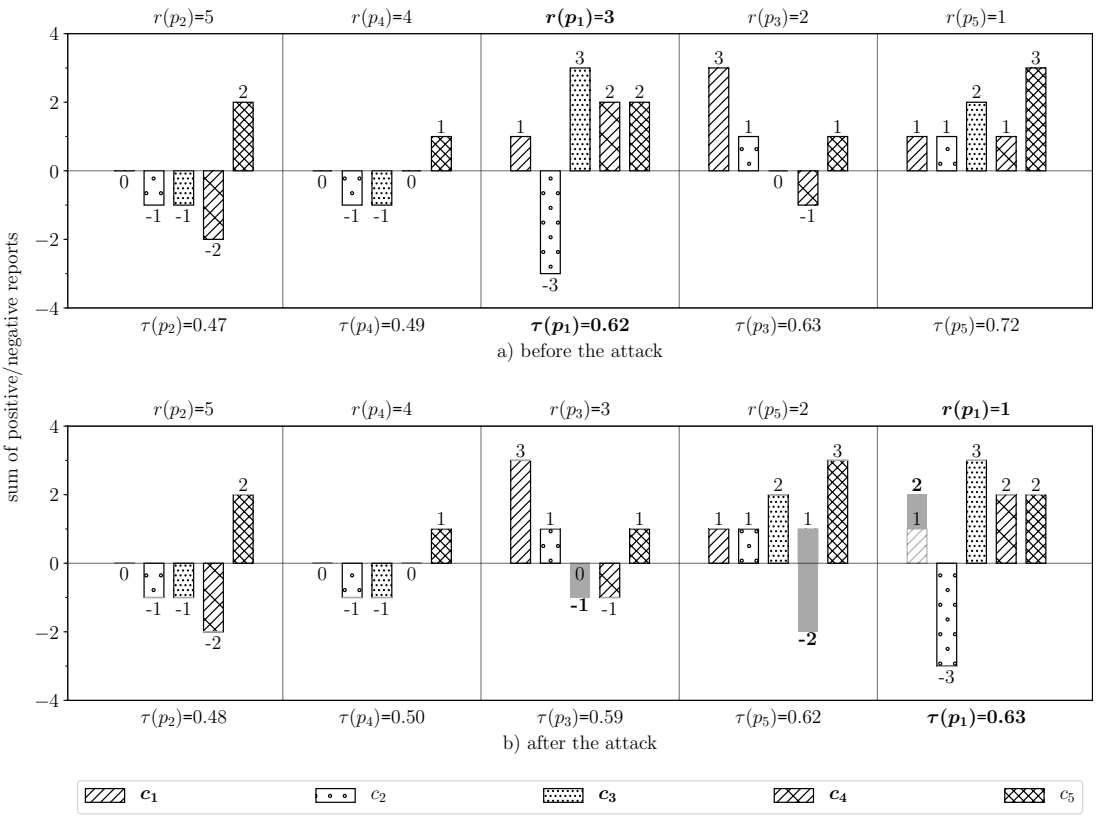

FIGURE 5.2: Agent δ's relative rankings of service providers before and after a strategic attack, where the ρ=5 and s=3. The malicious attacker, p_1, has control over witnesses $c_1, c_3, c_4 \in |\mathcal{W}'|$.

5.2.2.2 Optimal Attacks

The value gained from an attack depends on the intent of an attacker. Following our previous example, the optimal attack in this case for attacker p_a is to find the attack that gives the most trust value difference, while restricted by the power of attack, ρ. The objective can be to gain a higher trust value:

$$\begin{aligned} \mathcal{E}^* = \arg\max_{\mathcal{E}'} \ & \tau(\delta, p_a, \tilde{\mathcal{E}}^\delta) - \tau(\delta, p_a, \mathcal{E}^\delta) \\ \text{subject to } & \tilde{\mathcal{E}}^\delta = \mathcal{E}' + \mathcal{E}^\delta \end{aligned} \tag{5.17}$$

where $\tau(\delta, p_a, \mathcal{E})$ is the decision maker's assessment of the trustworthiness of the attacker p_a. However, the rank of the attacker may not increase, even in the cases where the optimal attack for this objective is found. This depends on the underlying formulation of the TRS. In addition, the decision of which party to trust are not made in isolation. A decision maker cares about how trustworthy an agent is with respect to others. As in the example in previous section, it may be more effective to badmouth others than to

inject more good reports to yourself. To this end, we change the attacker's objective to focus on its rank, in this case, it will be improving its *statistical* rank:

$$\begin{aligned} \mathcal{E}^* = \arg\max_{\mathcal{E}'} \; & r(\delta, p_a, \mathcal{E}^\delta) - r(\delta, p_a, \tilde{\mathcal{E}}^\delta) \\ \text{subject to } & \tilde{\mathcal{E}}^\delta = \mathcal{E}' + \mathcal{E}^\delta \end{aligned} \tag{5.18}$$

The optimal attack with this objective may include attacks that only improve trust value of p_a and may not change the values of other providers. This looks same as the optimization problem in Equation 5.17. However, the cases where improvement in rank is possible by decreasing trust values of others and increasing attacker's trust value are not considered. When all the cases are taken in to account, this attacker's objective is a hard (in this case, discrete) optimisation problem, which is strongly non-convex. This statement holds true for the ranking function ($r(\delta, p_a, \mathcal{E}^\delta)$), even if the trust assessment function is convex or concave. The difficulty in this problem comes from the fact that the objective function in Equation 5.18 depends on the implemented TRS, whose formulation can be impossible to access or intrinsically complex. Therefore, gradient based methods are not suitable and would fail to escape from local minima. For this reason, we propose two sampling-based optimization strategies from the literature to search for attacks.

5.2.3 Searching Attacks

By formulating attackers objective as an optimization problem, we can use well-known derivative-free methods for computing an attack. The main motivation in this approach is that searching the attack space in a brute-force manner is not feasible. To solve this, the idea is to sample the space by a set of rounds to approximate the solution of the optimization problem. In our case, the criteria to stop depends on many factors, such as computational resources of the attacker (i.e. having access to high-performance computing clusters) and/or adapting to changes in the system. By taking these into account, we introduce two sampling-based optimisation techniques:

Monte Carlo Sampling, MCS, is a Monte Carlo simulation-based approach to randomly sample the objective function to approximate the expectation of the optimal solution. In practice, until the stopping criterion is satisfied, we randomly select a set of $\mathcal{E}'$ from $\mathcal{X}$. At the end, we use best solution amongst all the candidate solutions produced. If the objective function is stochastic (i.e. trust function in Equation 5.5 being stochastic makes the objective function in Equation 5.18 stochastic.), we can further reduce the number of sampling steps by sample average approximation method (Kleywegt et al., 2002).

Hierarchical Sampling, HS, is a hierarchical optimisation technique, which assumes the objective function has a smoothness property (locally Lipschitz) (Bubeck et al.,

2011). This similarity property in our case is the assumption that similarly rewarding attacks are closely ordered in the space. This method aims to estimate the reward of a given objective accurately in the maxima (i.e. the highest reward), and loosely in the remaining partitions of the attack space. To achieve this, a binary tree is used for storing information about the attack space. Each node of this tree represents a partition of the space and holds some statistical information about estimated reward of the region (specifically, the number of times attacks are selected and the empirical average of the rewards from the region). After each selection, these estimates are updated from bottom up in the tree. At the end of sampling the space with a search strategy, the attack with the highest reward is returned.

5.3 Evaluation

Up to this point, we defined the environment the attack is going to take place in, the space of possible ways to attack the environment and the strategies to search for these attacks. Within this scope of unknown *orchestrated* strategies to attack TRSs, our aim is to characterize what these attacks are doing, the factors that influence the performance of these attacks and finally answering the question of is it applicable in practice. Hence, our hypotheses are:

- *Hypothesis 1:* Playing our (orchestrated) attack strategy is more rewarding than *simple* types of attack strategies.
- *Hypothesis 2:* Using previous agents rather than injecting new agents to the system yields a higher reward during attacks.

In our related work, we argued that the use of orchestrated strategies while attacking TRSs is plausible. However, the question of is it beneficial in terms performance for attackers remains. We capture this question by formulating our first hypothesis. We make comparisons between incorporating simple strategies from the literature. We continue by looking into the factors involved during an attack. The features of targeted environment such as connectivity (the number of interactions between agents), behavioural correlations between agents and finally use of agents that are already in the system or newly created agents are in our focus. Finally, we argue that our attack model can be used in real-world environments. To test these hypotheses, we make use of empirical experiments with simulated datasets and a real-world dataset. With simulated settings, we alter various factors of interest gracefully in the system. In addition, with the use of a real-world data set, we expand our observations capture intrinsic transformations in the dataset. This allows us to observe the differences in calculated strategies between the simulated and the real-world data set experiments. While we aim to provide maximal

coverage of possibilities with the use of simulated experiments, we further enhance our results with the use of a real-world dataset. In addition to the experiments that designed to test these hypotheses, we make an exploratory analysis on the characteristics of the attacks with respect to TRSs and show the impact of the attacks.

5.3.1 Experimental Setup

Given our hypotheses, we divide our experimental setup into two parts. First, we explain how the simulated data is generated. Second, we introduce our use of a real world data set. For this, we use Yelp's 2019 Dataset Challenge[1], which includes reviews that are written for businesses who are located in 10 metropolitan areas across United States and Canada. Four widely studied TRSs along with a simple baseline (average) function are selected for our investigation. These models (summarised below) represent a variety of commonly employed techniques for handling malicious witnesses. We implemented them based on information from respective papers, choosing reasonable values for parameters after a set of runs to ensure that performance is not hindered. We chosen these models:

BRS Jøsang and Ismail (2002) uses Bayesian updating to fuse observations from different providers and witnesses. The work by Whitby *et al.* (Whitby et al., 2004) extends the model by adding a filtering mechanism where evidence that deviates from the majority up to a degree is discarded. We selected the filtering mechanism version of the model, since we capture the original model with our baseline averaging model. *TRAVOS* Teacy et al. (2006) discounts the influence of witnesses by heuristically calculating the similarity between distributions of witness observations; in contrast, BRS discards outlier reports. In TRAVOS, similarity is calculated by tabulating the outcomes by using a particular selection of bins that denote regions of the outcome distribution. The model is selected for our evaluation with its first use of divergent reports in TRSs. *HABIT* Teacy et al. (2012) is a hierarchical Bayesian model to estimate trustworthiness by similarities between providers. Similar to TRAVOS, the decision maker calculates the similarity between the opinions of witnesses about a provider in comparison to other providers and the weighted average is calculated. Except the method eliminates the similarity heuristic by the use of the hierarchical model. *EIGEN* Kamvar et al. (2003) uses power iteration to capture transitivity of trust between parties. The outcomes of observations are normalised and stored in a global matrix. A global trust value is then calculated using the left-principal eigenvector of this matrix. We selected this popular reputation model to observe the performance of attack within a consumer-provider TRS setting. Other TRS models were not chosen, due to the known limits on their applicability to this problem and/or their similarity to the chosen models. For instance, BLADE model Regan et al. (2006b), limits our experimentation by making the assumption of *no missing data* Heckerman (2008) in each sample of available data. The given closed-form solution for

[1] www.yelp.com/dataset

computing trustworthiness of others relies on this assumption. Briefly, this means that each consumer is assumed to make the same number of interactions with each provider and receive same amount of reputational information from advisors about each provider in the system. In practice, the samples that are collected can be *incomplete*.

To get further insights about our TRSs in these different data sets, we propose a simple categorization of the attacks employed. This allows us to determine the type of attacks and the direction of primitive actions. Our categories incorporate the attack definitions given by Hoffman et al. (2009) and makes a concrete extension of their definitions. As shown in Table 5.1, these definitions are, malicious attacker uses witnesses that has control over to inject: *self-promoting* (SP) positive reports to the attacker, *self-slandering* (SS): negative reports to the attacker, *self-orchestrated* (SO): both negative and positive reports to the attacker, *slandering* (S): negative reports to other providers, *promoting* (P): positive reports to other provider, *orchestrated* (O): positive and negative reports to other providers and *complete-orchestrated* (CO): negative or positive reports to both the attacker and providers.

		Categories						
		CO	O	SP	S	SO	SS	P
Direction	Attacker	(+, -)		+		+, -	-	
	Others	(+, -)	+, -		-			+

Table 5.1: The categories of the attack strategies considered with their directions

We measure the frequency of each type of attack occurred by our attack strategy and the degree of rank gain that is achieved. This is relevant to our hypotheses, because we can then make comparisons between known types attacks versus our attack model in terms of performance. If the characteristics of the environment differs, we can measure how much it influences the attacker and the types of strategies used to benefit the attacker. Finally, we can identify the types of attacks that has more impact on the models within types of characteristics that they have using these metrics.

5.3.1.1 Simulated Dataset

Parameter	Value	Description
$\|\mathcal{P}\|$	20	The number of provider agents
$\|\mathcal{W}\|$	20	The number of witness agents
s	2	The number of witnesses under the attacker's control
t	10	The number of provider observations made by each witness

Table 5.2: Experimental Parameters

To generate simulated data for our evaluation, we follow this procedure: Before the attack, we assume a set of witnesses $\mathcal{W}$ interacted with a set of providers $\mathcal{P}$ over a number of rounds. The outcome of the observations made by witnesses about providers

are drawn from categorical distributions (i.e. $O^{t'}_{c_j \to p_j} \sim \text{Cat}(\theta^{p_j})$) with a parameter that is given to each provider. Parameters of categorical distribution, θ^{p_j}, are drawn from two different distributions. The first is a Dirichlet distribution (i.e. $\text{Dir}(\alpha)$) with all its parameters set to 1. The intuition to use this uninformative prior is to setting equal probability to all possible assignments of θ^{p_j} (i.e. the principle of indifference). The second distribution is used for looking into the effect of providers, if we assume that providers are behaving similarly. We capture this notion by using a Dirichlet distribution with the parameters set to 20. Finally, to capture connectivity in TRSs, we introduce a metric: *indirect knowledge degree* d, which denotes the probability of each witness interacting with a provider and t denotes the number of times the interaction has happened before:

$$\Pr(O^t_{p_j \to c_j} = x) = \begin{cases} 1 - d & x = 0 \\ d \cdot \theta^{p_j}_x & x \neq 0 \end{cases} \tag{5.19}$$

Witnesses transform their observations to the target by behaviour matrices Θ^{w_i}, which are right stochastic square matrices. Formally a single behaviour matrix is:

$$\Theta^{w_i} = (\theta^{w_i}_{xy} |\ \theta^{w_i}_{xy} \geq 0, \textstyle\sum_y \theta^{w_i}_{xy} = 1) \in \mathbb{R}^{k \times k} \tag{5.20}$$

where each row sums to 1. Each row represents a probability vector and reports of witnesses are categorically distributed by each row. The values for each row in Θ^{w_i} is drawn by a specified distribution where the supports of the distribution can be used as a probability vector (the sum of each row is 1.0 and each element is larger than or equal to zero). In our experiments, we used an uninformative Dirichlet distribution for each row. When a witness reports an observation $O^t_{c_j \to p_j}$ to the decision maker, then the reported observation is denoted as $O^t_{w_j \to p_j} \sim \text{Cat}(\theta^{w_i}_{x,*})$ where $\theta^{w_i}_{x,*}$ is the x^{th} row of Θ^{w_i}. To illustrate concretely how the reports are transformed consider the following example. If we assume binary observations (i.e. $k = 2$) and a witness is *completely* honest in reporting the observation (i.e. no changes to what is observed), then the behaviour matrix of the witness Θ^{w_i} is an identity matrix I_k. At the same time, if the witness is *completely* dishonest (i.e. opposite of what is observed), then the behaviour matrix is an exchange matrix J_k. In the case of a matrix of ones multiplied with 0.5 (i.e. when $k = 2$: $\Theta^{w_i} = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}$), the reports will be random, independent of the observation. Given these behaviour matrices, the number of reports that are generated by each witness is denoted with t.

Throughout our simulated dataset experiments, we consider four experimental variables: the strategy used to search for attacks (MCS or HS); the connectivity between witnesses and providers (d); the power of the attacker (ρ); and the behaviour of witnesses (θ_{c_i}). Other parameters are fixed as specified in Table 5.2. (given by the selected TRS) as the

attacker's target. The rest of the parameters remain constant as shown in Table 5.2, unless otherwise stated.

5.3.1.2 Real-World Dataset

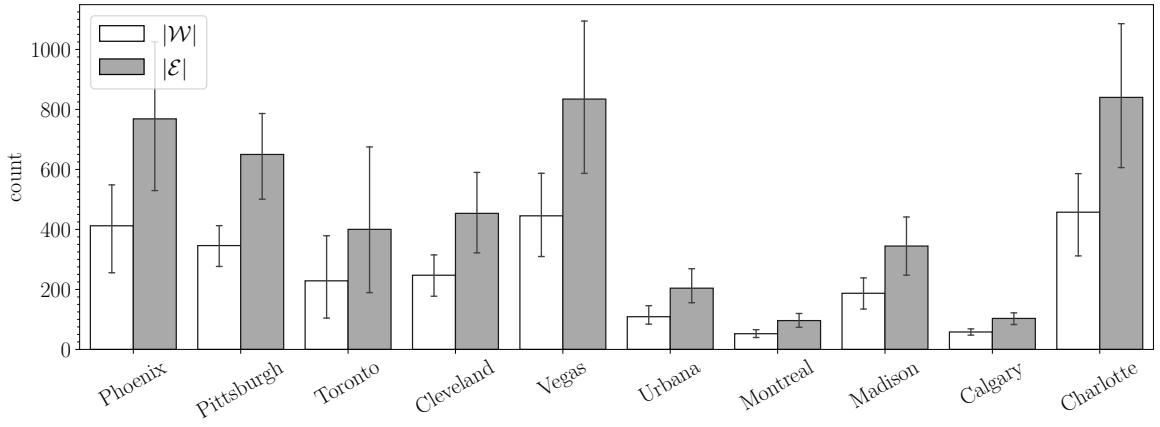

FIGURE 5.3: The average number of advisors ($|\mathcal{W}|$) and the count of all reports ($\mathcal{E}$) available in random regions of each city.

In the previous section, we explained how the synthetic data is generated and how the parameters are controlled. While, this is a common way to generate various instances of the population, how much the process accurately models the interactions in a TRS in practice is unknown. To explore real-world performance of our methods, we use a data set released from the company Yelp, as their 2019 Dataset Challenge. Dataset includes reviews that are given to restaurants, cafes and other types of businesses. In total, there are 192609 records of business and 6685900 reviews in the data set from 2014 to 2018. These businesses are located in ten different cities from the USA and Canada. In Yelp, reviewers can provide reviews with 5 star ranking. In our experiments, we select 10 random locations that are close in proximity (less than 2 kilometres) to each city centre from 2017-2018. Then, 20 restaurants which that are close to each location are selected. Since TRAVOS, BRS and EIGEN models use binary observations (positive or negative) in assessments, we preprocess the data as following: 5 star rating as 2 positive, 1 star as 2 negative, 3 star as 1 positive and 1 negative, 4 star as 1 positive and, finally 2 star as 1 negative report. Based on our observation in Figure 5.5, we tuned the exploration percentage to 0.01% to search the space for all strategies across these experimental settings.

We aim to capture various population settings in our sample from the dataset. Figure 5.3 shows the sample of the data that we gathered. In our sample, Vegas has more

[3]Brackets that are used in the equations inside the figures are *Iversion brackets* (Graham et al., 1989). Brackets convert a logical proposition to 1 if proposition is satisfied, otherwise returns 0. Formally, if P is a binary proposition , then: $[P] = \begin{cases} 1 & \text{if } P \text{ is true;} \\ 0 & \text{otherwise,} \end{cases}$

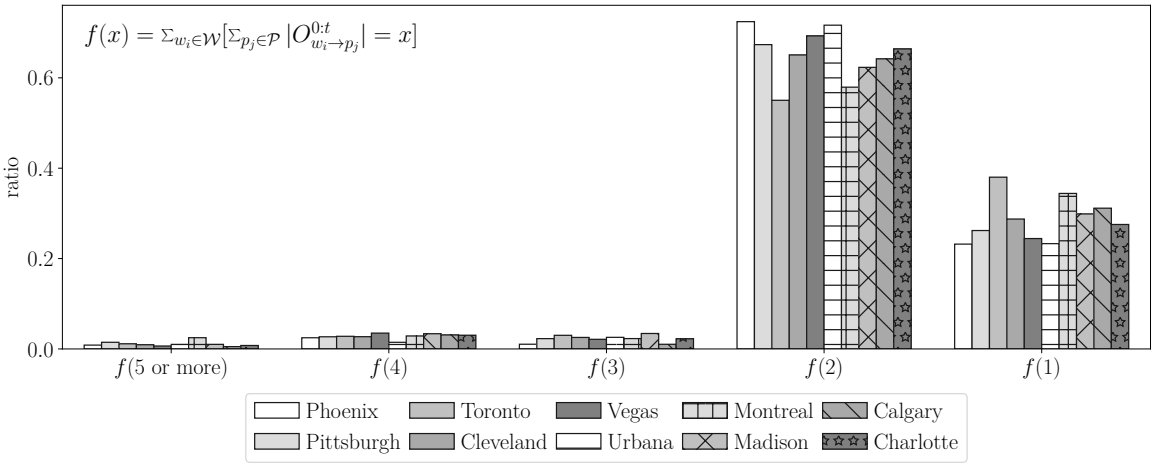

FIGURE 5.4: Comparing the ratio of witnesses from each city, categorized by the amount of reports that are provided to TRS (i.e. $f(x)$)[3].

reports and advisors than other cities. The fewest number of reports and advisors are in Montreal. The sample overall in each city includes a higher number of reports than the number of witnesses. Figure 5.4 shows the ratio of the witnesses from each city. Witnesses who provided at least 2 reports, $f(1)$ and $f(2)$, were the most common in every city. We observed that the ratio of the rest is marginal. In terms TRS selection, Yelp uses simple weighted averaging. However, we further make experiments with all TRSs mentioned before. We measure the probability of each witness interacting with a provider, *indirect knowledge degree*, from samples in Yelp by calculating:

$$\overline{d} = \frac{\sum_{w_i \in \mathcal{W}} \sum_{p_j \in \mathcal{P}} |O^{0:t^{\max}}_{w_i \to p_j}|}{|\mathcal{W}| \cdot |\mathcal{P}| \cdot t^{\max}} \tag{5.21}$$

We measured this to be $\overline{d} < 0.06$ across all cities. While the number of reports given by a witness, $t^{\max}$, in the range of $[2, 8]$. In the simulated settings, we set $d = 0.5$, but varied $t^{\max}$ in our experiments. Given in this initial analysis, when both datasets are compared, the connectivity is expected to be significantly lower in the Yelp Dataset than in our simulated experiments.

5.3.2 Results

Here, we present the results of our experiments. In each simulated experiment setting, every experiment is repeated over 3000 different instances. By doing so, we minimise the effect of different initial attacker ranks, uncertainty from the searching methods and population settings. In our results, we plot the distribution of rank gain and *mean rank gain* over these scenarios to illustrate the performance of our attacker model. To validate the statistical significance, we performed pairwise Mann-Whitney U tests with

Bonferroni correction. Our reason for choosing this test is that the resulting rank gain distributions were not normally distributed according to Shapiro-Wilk tests. As mentioned, the rank gained after each attack from the target model is treated as a black-box mechanism. This creates a new problem, where the attacker needs to make a decision on the number points to be sampled from the attack space. Since, we omit any prior knowledge regarding the TRSs used, we empirically determine a reasonable value for all experiments. In Figure 5.5, we observed that as the size of attack space increases, the proportion of the space required decreases to reach a plateau. Although, this does not necessarily mean that if we determine a proportion of space of attacks to explore from the observed plateau in every scenario, the attacker conducts an optimal attack. However, our empirical observation offers a reasonable value for our analysis to identify vulnerabilities of the models. After this observation and given the selected TRSs, we selected 1% in our simulated experiments. This value, in theory, may be derived by relaxing the *black-box assumption*. We leave the selection of this parameter given the characteristics of target model as a future work. Before going through the results that correspond to each hypothesis, we elaborate on results regarding the selection of different search strategies and optimizing attack strategies with respect to other TRSs.

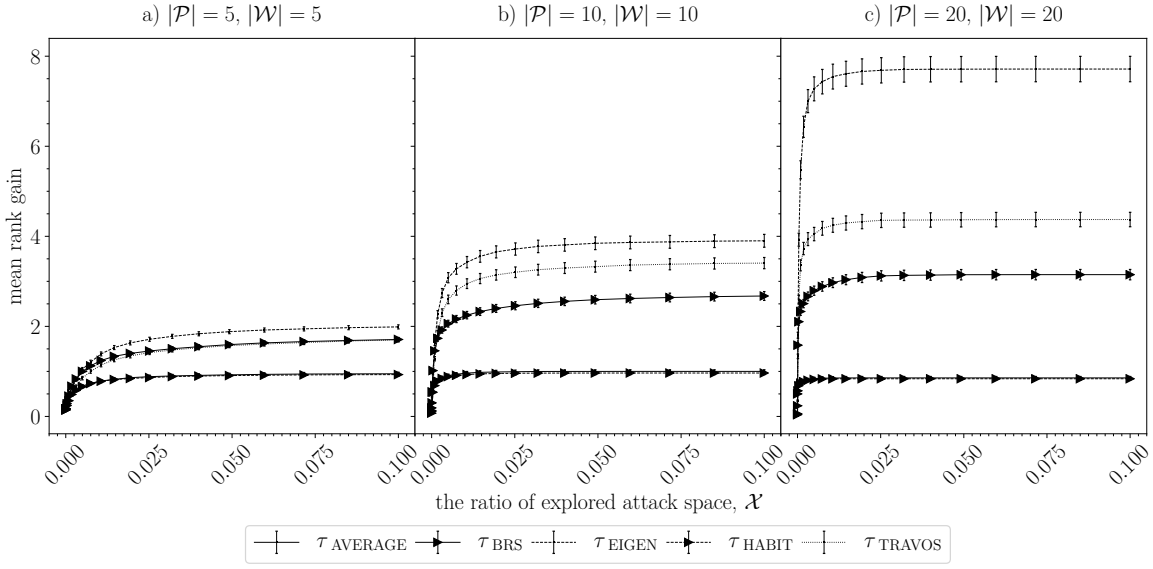

FIGURE 5.5: Mean rank gained from MCS with respect to the ratio of the attack space explored in three different environments.

Identifying an effective search strategy

Figure 5.6 shows the performance of the attacker across different TRSs given the selected search strategy and a comparison between the starting point of the attacker in our simulated experiments. The attacker achieves a minimum of 2 rank gains on average for all TRSs, but EIGEN and TRAVOS are significantly more vulnerable. With respect to

our search strategies, MCS performed at least as well as HS for all TRSs, and showed a significantly higher performance against BRS, EIGEN and TRAVOS. All three of these cases were statistically significant with $p < 0.001$. The cause of these differences might be due to the fact that the objective function lacks smoothness. Given the relative performance of MCS and HS, we selected MCS for subsequent experiments.

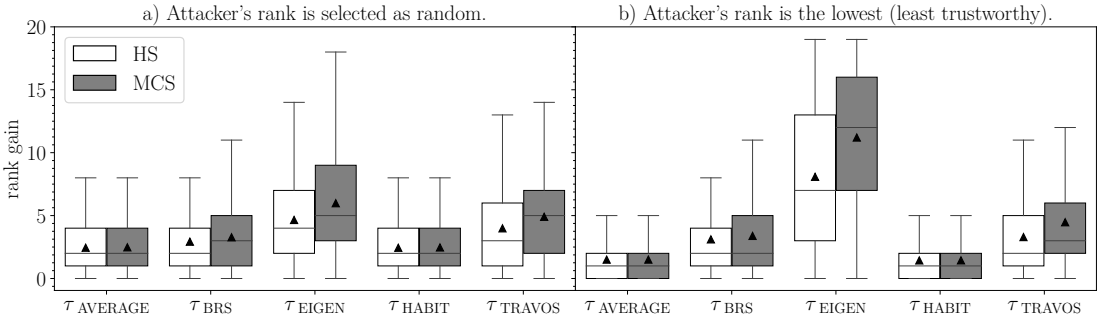

FIGURE 5.6: Comparing MCS and HS in varying TRSs. Triangles denote the mean of the corresponding distribution.

Optimizing attack strategies

Figure 5.7 shows the performance (rank gain) as power ρ, population behaviour and connectivity of available evidence (Figure 5.8) are varied. As the power of the attacker increases, this may be exploited to achieve a greater rank gain for all TRSs (Figure 5.7a-b). The rank gain is bounded by the starting rank of the attacker. The rate of increase does, however, vary across the two population profiles. When the results from Figure 5.8 are compared with Figure 5.7a, the rank gain achieved against EIGEN is significantly lower in the case when the population parameters are sampled from a concentrated Dirichlet distribution. The average rank gained against other TRSs behaves similarly as the power of the attack, ρ, is increased.

Hypothesis 1

Our hypothesis is to confirm that if our attack strategy is more rewarding than choosing simple types of strategies, given the same attack power, ρ. For this, Figure 5.9 compares the simple strategies and our attack model. The simple strategies considered are: self-promoting from a random advisor, SPO, i.e $O^{t:t+\rho}_{w_{\text{random}} \to p_1} = (1, ..., 1_\rho)$ where p_1 is the attacker, slandering a random provider (we ensure that the rank of this provider is higher than the attacker) from a random advisor, SO, $O^{t:t+\rho}_{w_{\text{random}} \to p_{\text{random}}} = (2, ..., 2_\rho)$. Our experiments partially validate our hypothesis, where *simple* types of attack strategies are less rewarding or same then our attack strategy. Particularly, on average slandering was least effective. The significant difference is observed in EIGEN, BRS and TRAVOS

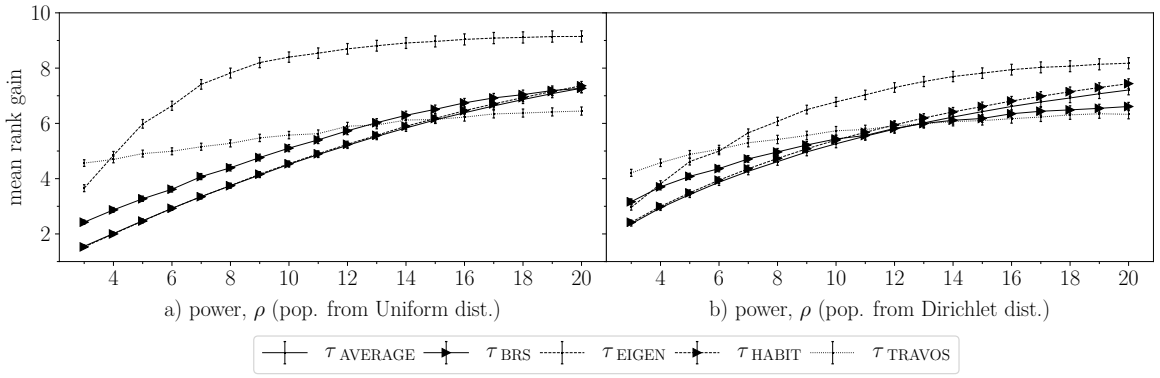

FIGURE 5.7: Comparing TRSs where power of the attacker, the evidence available and the population behaviour is varied. Error bars denote the standard error of the mean rank gain.

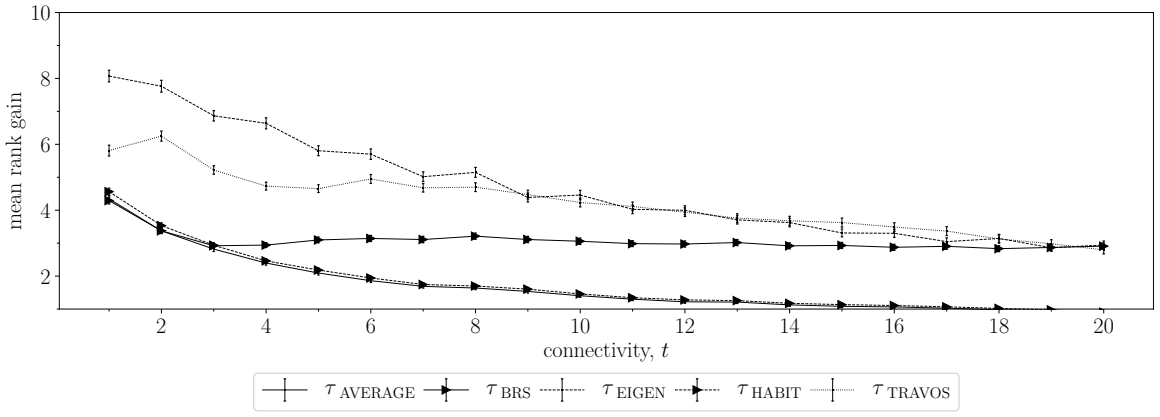

FIGURE 5.8: Comparing TRSs where power of the attacker, the evidence available and the population behaviour is varied. Error bars denote the standard error of the mean rank gain.

models, where MCS outperformed other strategies. All three of these cases were statistically significant with $p < 0.001$. The outliers show that some instances simple strategies where able achieve the maximum reward, however on average the reward gained is much lower.

Hypothesis 2

In our second hypothesis, we hypothesized that the attacker will gain higher reward if an attacker uses our strategy with agents that have already engaged with the TRS, rather than creating new agents (as is a Sybil attack). Figure 5.10 shows the difference between the mean rank gain by selecting witnesses that are already in the system versus creating new witness identities. Each point in the figure is computed by subtracting the mean rank gained from selecting *old* witnesses from selecting *new* witnesses. Our

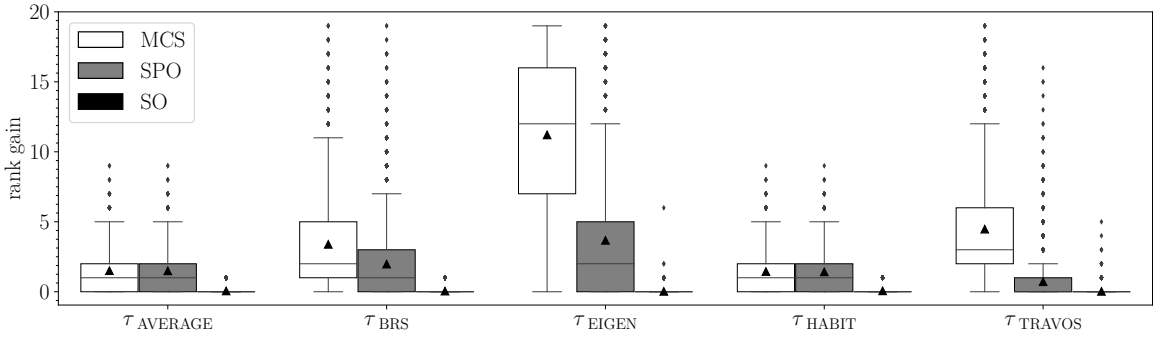

FIGURE 5.9: Comparing MCS with SPO and SO. Triangles denote the mean and circles denote the outliers of the corresponding distribution. The attacker rank is chosen to be the lowest amongst all other providers.

results show that when attacker uses our model with the use of either model, mean rank gain is identical for TRS models: Average, BRS and HABIT model. We observe that increasing the number of advisors increased the mean rank gain for the TRAVOS model. On the other hand, our results for the EIGEN model showed that creating new advisors was a better choice for the attacker, since the difference is negatively increased.

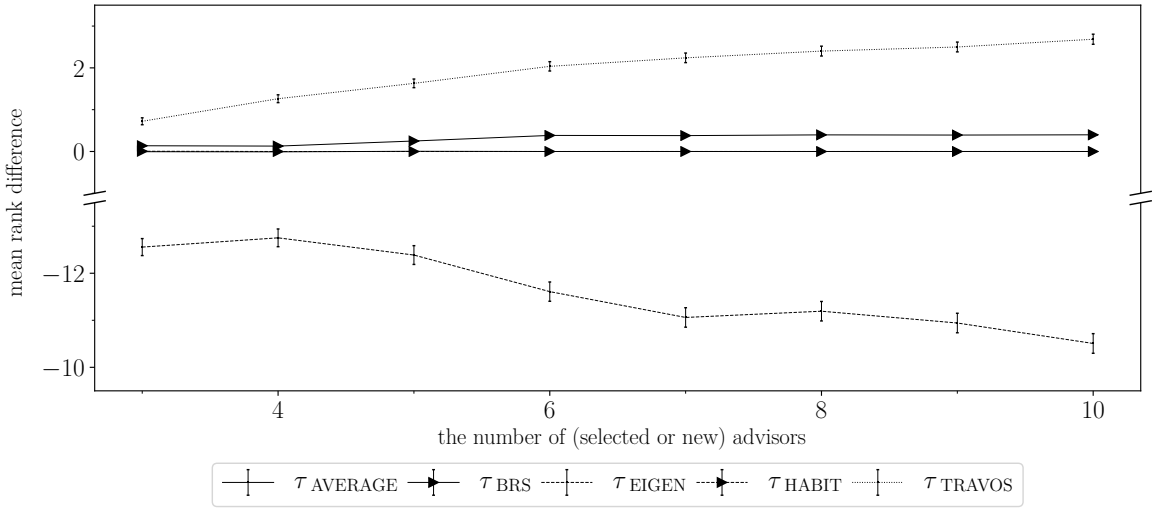

FIGURE 5.10: Mean rank difference of selecting witnesses that are already in the system versus creating new witness identities.

The effect of connectivity and attack power towards selected strategies

Figure 5.12 shows when the connectivity is set to: $t = 4$ and $t = 16$. We observe that the strategy that is selected the most is *complete orchestrated* (CO) across all settings. The second highest ratio of selected strategy is *self-promoting* (SP). As the number of reports increases, the ratio of this strategy decreases in all TRSs. Figure 5.13 shows when the attack power is set to: $\rho = 4$ and $\rho = 16$. The figure shows that in each

TRS model that we used in our experiments, *complete orchastred* (CO) was selected the most. The percentages of CO in all TRSs in $\rho = 16$ is less than in the cases where $\rho = 4$. When $\rho = 16$ in all cases, the ratio of SP is higher than in $\rho = 4$. This includes the mean rank gain of SP, which is higher than CO. With the Yelp dataset, there were more cases where a strategy was not found given where the attack power was set to $\rho = 5$. Figure 5.11 shows the ratio of best attacks in EIGEN were SP. In TRAVOS, the rank gains and the number strategies found were less than all others.

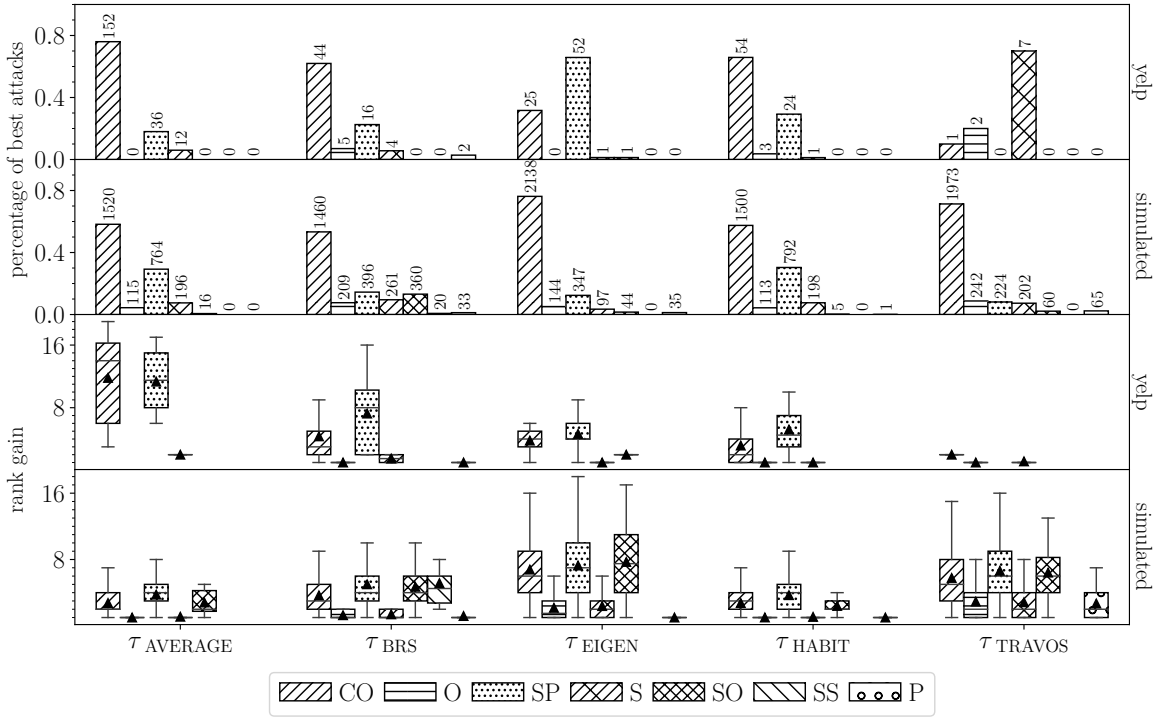

FIGURE 5.11: Distributions of rank gain achieved when an attack type is selected in varying TRSs: simulated and Yelp settings.

Summary

To summarize, the following can be stated regarding our hypothesis:

- *Hypothesis 1* was **supported** by our experiments, our strategy is more rewarding than *simple* types of attack strategies given the space we explored in our experiments.
- *Hypothesis 2* was **not supported** by our experiments: The results varied between different TRSs, when the attacker's strategies incorporate witnesses versus selecting witnesses that are already in the system.

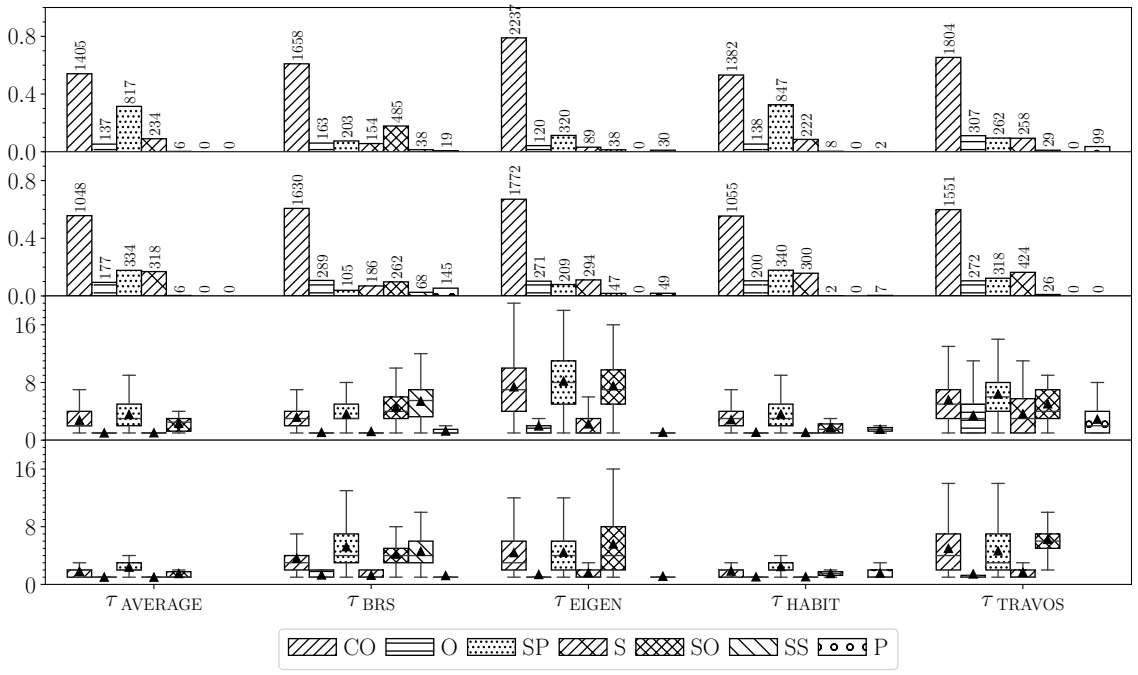

FIGURE 5.12: Distributions of rank gain achieved when an attack type is selected in varying TRSs and connectivity settings.

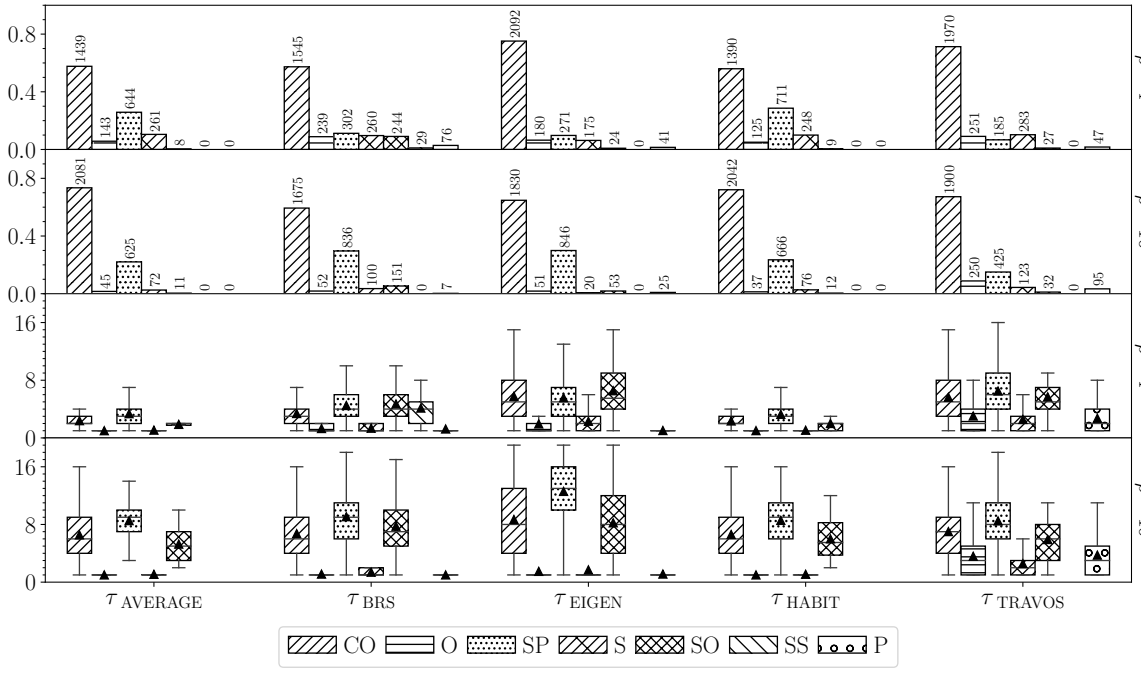

FIGURE 5.13: Distributions of rank gain achieved when an attack type is selected in varying TRSs and power settings.

In addition to these hypotheses, our key finding is that the distributions of rank gain by strategies that are identified by our model depends on connectivity, the attack power, including new or reusing previous advisors, and the TRS that is targeted.

5.4 Discussion

Our results show that devised strategies differ with respect to the environment that is targeted. In particular, when the connectivity of TRS is increased, we see a decrease in *self-promoting* being a good strategy, regardless of the attack power. Meanwhile, when the connectivity is fixed, the effect of self-promotion increases for all TRSs models with higher attack power. The amount of rank gain achieved was consistent in each TRSs, when these strategies were selected: *complete-orchastrated* (CO), *self-promoting* (SP) and *self-orchastrated* (SO), except in BRS *self-slandering* (SS) resulted a higher rank gain than others. Overall, this shows that if an attacker uses advisors to invest in injecting new reports directly to their account (i.e *self-promoting*), a better option would be to distribute a portion of reports to conduct other strategies. Our results point to this in expectation, whereas in smaller portion of the cases observed, other strategies, such as orchestrated (O), slandering (S), self-orchestrated (SO) are higher rewarding strategies.

In experiments made with the real dataset, we can not quantitatively measure the performance of models in terms of trust predictions, since the ground truth does not exist. Therefore, the parameters of models can not be selected by their empirical performance. During the experiments, we continued to use the same parameters for TRS models as the simulated data set. However, we suspect that TRAVOS' robustness to attacks (average rank gain was 0.35) may caused by the lower trust assessment performance. In terms of performance of our attack model, we showed that the performance of Hierarchical Sampling (HS) was lower than Monte Carlo Sampling (MCS). We suspect that it may be due to the fact that the method's main goal is to minimize cumulative regret on rank gain. Therefore, a proportion of sampling steps are selected in nearby regions more than distant parts of the space. As we stated earlier, our optimization problem is discrete and non-linear. Another further direction to investigate is using techniques from nonlinear integer programming. Onn (2010) shows if the problem is convex, there is a polynomial time optimization over these problems if Graver bases are given. The idea is to sample set $\mathcal{X}$ by following the direction of Graver bases vectors. If the previously selected direction does not yield an increasing reward, another basis vector is selected iteratively. However, computing Graver Bases is an exponentially hard problem. One future research direction would be to investigate sampling techniques on Graver Bases to direct MCS. This may result higher performance than MCS if our objective function is known to be or given as convex.

In our experiments, we used the direct addition of false evidence. However, this does not limit the applicability of our model to include other primitives. A selection of primitive strategies can be combined and used in our model. For instance, the space of attacks can be directly tailored for the attack that requires direct access to systems, in this way attacker can explore which sets of information to be removed from the system. However,

attacks that require an ordered sequence of causally-related primitive actions, such as *oscillation attack* (Srivatsa et al., 2005), *camouflage attack*, *reputation lag* (Muller et al., 2016), *exit attack* requires further additions to our model. Our method can be used to analyse vulnerabilities in trust models that use stereotypical information. Instead of having primitive actions that involve injection of evidence, we can explore which features are best to acquire to achieve a certain level of rank for the targeted decision maker.

Identified vulnerabilities (attack types) varied between the simulated data set and Yelp data set. We believe that the choices of strategies that our method identified was due to the intrinsic differences in structure between two datasets. We showed in Figure 5.4 the sparsity of dataset in terms of having high ratio of witnesses having provided 1 or 2 reports significantly affects the choice of a good strategy. This ratio is higher in our simulated dataset, modelled this by *indirect knowledge degree*. Therefore, our findings show signs that our method exploits the structure of the dataset and generate an appropriate strategy. In addition, the results from simulated versus Yelp data set were consistent in terms of the proportions of selected strategies. A major difference was observed in TRAVOS, where the rank gains where significantly lower than others. We suspect that this is due to not being able to select parameters required for the model empirically. Since the ground truth can not be quantitatively measured to increase performance of the models.

Our reasoning for selecting specific categories in our experimentation are two fold. First, this is to analyse how much reward can be achieved if the attack is one of those considered in prior work. A major difference here is that these categorizations given by the prior work do not give a direct plan on which advisors to select. The second reason is that when we have specified these categorized, in our analysis we are left with the cases where the attack was not in any of these categories. To capture the remaining attack types, *self slandering* and *promoting others* were added to our analysis. In our results, we found out these two strategies counter-intuitively result in rank gains, although the occurrence of these are few and the rank gains are marginal. These cases were most common in BRS. We predict that this is due to discounting/removal mechanisms that such models have for outlier providers. In rare cases, such strategies may allow the provider to be within the group by boosting or behaving similarly to others. However, these two extra strategies do not necessarily mean that these can be only attack types in the space of injecting new evidence by a set of advisors. These categories can be expanded by, for instance, is the advisor selected always provides negative feedback? This alone can be a category that can be factored in for further analysis.

It is worth mentioning the computational cost of searching for attacks. On our test system[4], we found that increasing number of witnesses and providers entails a significant

[4]We used IRIDIS 4 HPC facilities at the University of Southampton, where a single simulation is run on a node, which includes 2.6 GHz Intel Sandybridge processors with 16 GB RAM, running Red Hat Enterprise Linux Server release 6.10 (Santiago).

computational cost. This difference is shown in Table 5.3, which simulated setting had 20 witnesses, while Yelp setting had between 200 and 500 witnesses. The difference between the time taken for each attack search in different TRSs varied significantly. Figure 5.14 shows this attack search performance difference. This observation opens room for further research on approximating the targeted TRS to achieve gains in terms of time. We project that employing a set of offline TRSs and finding a similar TRS which can be selected as a representative of the actual targeted TRS is a promising future direction.

Assumptions made in this research include that the attacker can observe all available evidence, and knows the TRS being employed. The attacker can, therefore, calculate the ranks of each provider whenever the evidence changes. In practice, the attacker will have some uncertainty of the TRS being used in the target system. From the perspective of the designer, however, it is reasonable to analyse resilience of a TRS from this worst-case perspective. It is worth mentioning that our attacker model selects witnesses according to the objective function without considering the cost of using a particular witness. Costs associated with witness selection may vary; e.g. employing a witness considered trustworthy may incur higher cost. This could, however, be captured by adapting the objective function. Having said this, we have demonstrated that our TRS analysis method can assist the designers of TRSs to identify vulnerabilities against orchestrated attacks. Coordinated attack patterns identified for a specific TRS may be used as a basis for automated attack recognition mechanisms. Suspicious patterns identified can then be passed on for further investigation.

Dataset	**Average**	**Deviation**	**Total**
Yelp	4:23:31	9:41:44	83 days, 10:43:47
Simulated	0:00:20	0:00:58	786 days, 4:28:59

Table 5.3: Average time taken for each attack search and total simulation time

We view the TRS analysis method proposed as a basis for reducing vulnerabilities in future trust models. Coordinated attack patterns identified for a specific TRS may be used as a basis for automated attack recognition mechanisms to supplement the system. Suspicious patterns identified can be passed on for further investigation.

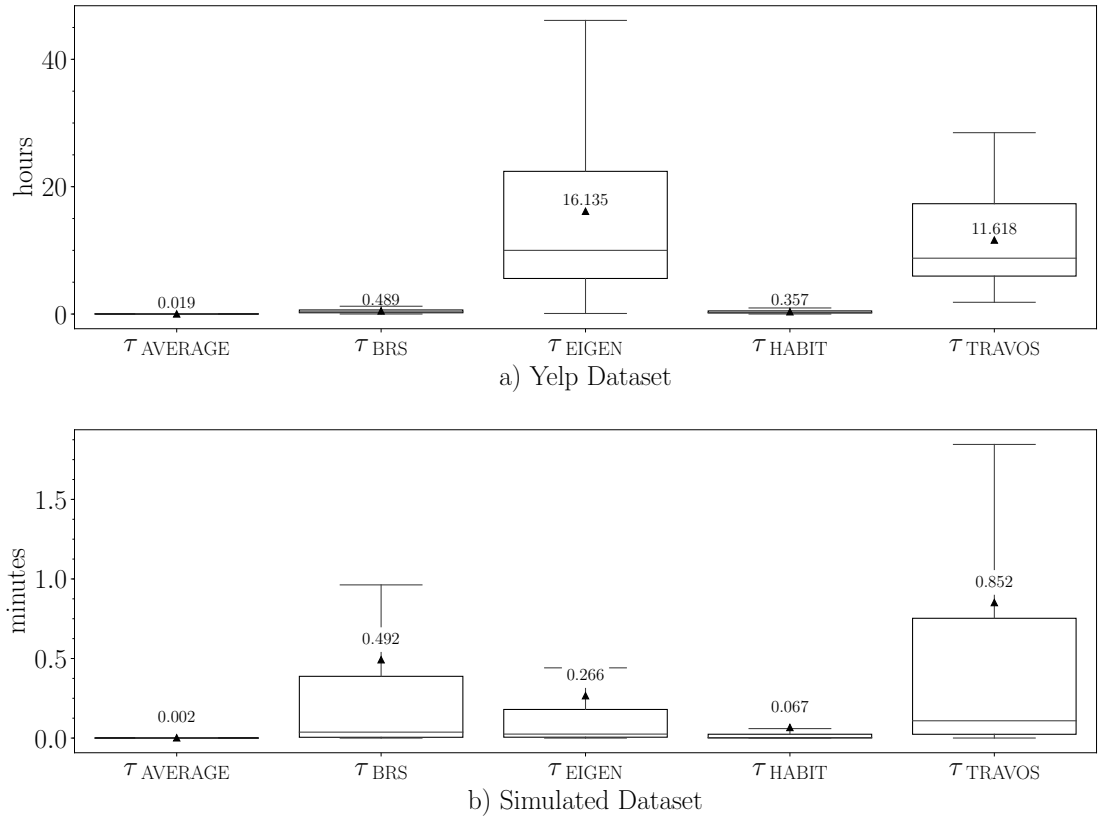

FIGURE 5.14: Distributions of time taken for each attack search in varying TRSs

5.5 Summary

We have introduced and demonstrated the practical value of a new and generic method for identifying vulnerabilities in TRSs. Given a characterisation of the space of possible attacks, we define an attacker model. Our model may then be employed to search for effective strategies through derivative-free optimisation methods. The outcome is a set of attack profiles and an estimate of the vulnerability of the TRS to an attack of this kind. In this way, we contribute to the development of future trust and reputation systems that are less vulnerable to sophisticated external threats. In the next chapter, we discuss on how the contributions from Chapter 4 and Chapter 5 can be applied in diverse scenarios.

Chapter 6

Applications

So far, we have presented trust based decision-making processes in resource constrained environments and mechanisms to search for attacks in trust models to explore vulnerabilities. Our decision processes enable decision-makers to strategically spend their resources in varying information sources by using a general trust model, meanwhile the techniques we devised for our vulnerability analysis complement these decision processes for a broad range of trust models.

In this chapter, we will provide some discussion and future research directions regarding some example applications of our decision processes and our attack search mechanisms in varying multi-agent systems, where trust is applicable. These examples will motivate how both of our contributions can be used together in diverse scenarios to tackle real-world challenges.

6.1 Automated Negotiation

Negotiation between autonomous agents is known to be a pivotal concept in multi-agent systems (Jennings et al., 2001). The concept is useful to formalize the interests of agents, interdependencies, and managing coordination-cooperation between agents at *run-time*. Example applications of this concept include conflicts between buyers and sellers in e-commerce (Kraus, 2001), energy trading in smart grids and online dispute resolution. Broadly, negotiation incorporates three topics: protocols (rules that explain how agents will negotiate), objects (i.e. preferences) and the decision-making models which agents use to achieve their objectives. While these cover more about underlying mechanisms of single negotiation, the negotiations are typically repeated over time.

In repeated negotiation scenarios (a single negotiation session illustrated in Figure 6.1), agents may choose to engage in a negotiation with previously known agents or those that have not yet been encountered. Between repetition with the knowledge gained from

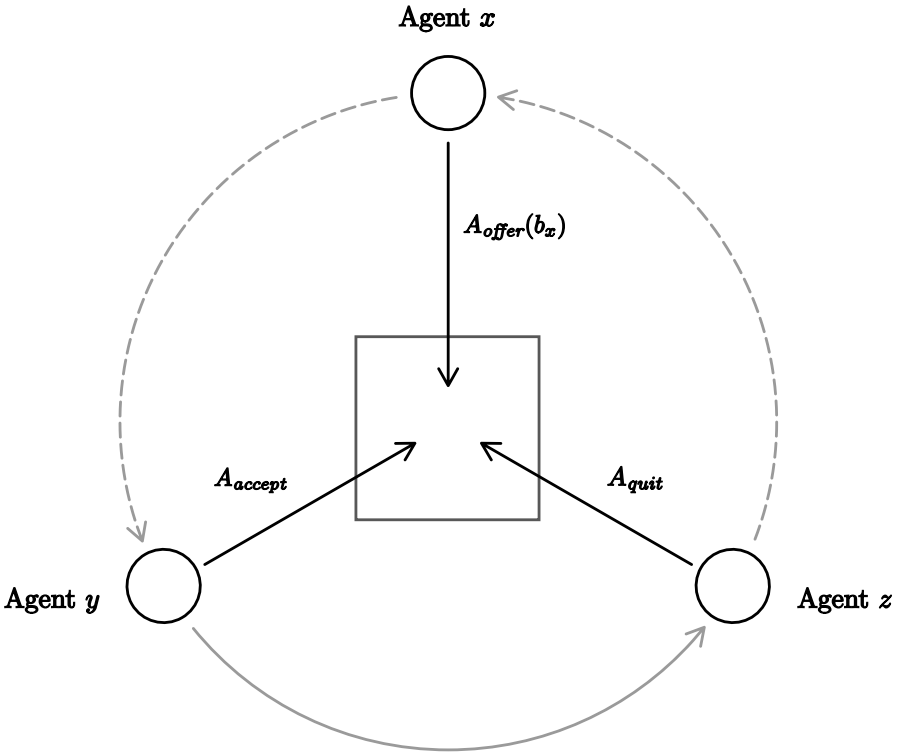

FIGURE 6.1: A multi-party negotiation session example, where x and y agrees on the offer b_x and z walks away from negotiation.

previous sessions, agents may be able to predict a degree of *trust* about the behaviour of other agents. Conceptually, *trust* in negotiation can be thought as an agent's belief about other agents being *fair* in each negotiation session. This metric can be computed by a *social welfare* function, if the utility functions of all parties are known (Fatima et al., 2014). However, agents being self-interested, makes the evaluation of a negotiation session challenging, since the utility function of the other parties (i.e. opponents) are unknown. Estimating utility functions is possible with prediction mechanisms that are known as *opponent modelling* methods (Hindriks and Tykhonov, 2008; Hindriks et al., 2009; Baarslag et al., 2013). These tend to use recordings of actions from a negotiation session.

The use of opponent modelling would enable a measure for the outcome of a negotiation session. By this, agents may build trust on other agents regarding their cooperation levels. The decision-making mechanisms that we elaborated on in Chapter 4 would apply in these cases, where an interaction can represent a single negotiation session and the outcome can be estimated (unlike the cases that we focused where the decision-maker sees the *true* outcome.) Our methods require environments where querying agents for other agents' *trust* is possible. Companies that provide analysis of other companies, for instance Gartner, can be given as example to witness information. Advisory and consultation services tend to have a cost for their analysis. The duration of negotiation and the resulting agreement can be interpreted as the costs of a direct interaction.

The degree of trust can help the choice of partners and the negotiation strategy (Ramchurn et al., 2004b). Depending on the trust level, the duration of negotiation can be lower. A hybrid model that includes our decision-making model for choosing partners and a negotiation strategy that takes into account the level of trust would increase the utility gained from a repeated set of negotiations. Other types of sociological information that we elaborated on in Chapter 2 can be incorporated by the trust model that is selected. The features of negotiating parties that can be taken into account include certificates, accreditation and degrees.

Our vulnerability analysis that we elaborated in Chapter 5 directly applies in this domain. Our methods to search for attacks can directly invoke this hybrid model to find types of attacks and the necessary strength that would be required to manipulate the decision-makers. In each negotiation session, an attacker can compute the degree of false information to inject to reduce the reputation of competitor agents. Therefore, decision-maker agents would end up preferring to negotiate with the attacker. By this, we can measure the robustness of overall negotiation strategy.

6.2 Task Delegation

Task delegation (i.e. task allocation) in multi-agent systems is the process of agents performing tasks behalf of other agents (Griffiths, 2005). This is related to *crowdsourcing* where agents tend to represent human teams. The field has been arisen by the crowdsourcing markets. In these, *requesters* would like their tasks to be completed by a set of *workers* which complete these tasks in exchange for a payment (Zheng et al., 2017). Amazon Mechnical Turk can be given as an example for a crowdsourcing market. The tasks include completing surveys and labelling images in this platform. The common challenges in the applications stem from varying worker skill levels and the difficulty of tasks (workers willing to accept the task according its price, i.e. difficulty-price ratio). Tackling these require a decision-making strategy, which needs to take these into account and make decisions on selection of *some* tasks for *some* workers with appropriate pricing. As an application of our methods, we are interested in the process of selecting/filtering workers from a pool of workers.

Trust and Reputation Systems have been employed in crowdsourcing (Slivkins and Vaughan, 2014). The general motivation is to encourage high quality tasks and reduce spams for workers. From the requester's perspective, this is to find highly skilled workers (specifically to distinguish workers in terms of performance and as a support tool for assessing performance as a prior belief.) There have been a set of decision processes that introduced with the usage of TRS. While Ho et al. (2012) and Zhang and van der Schaar (2012) use social norms to incentivize good behavior which incorporates

reputation systems, Yu et al. (2013b, 2015) proposes preliminary approaches to tackle task delegation in terms of budgetary constraints and reputation awareness.

The decision-making mechanisms that we developed can be adapted into task allocation setting. However, this requires significant changes in order to solve domain specific challenges. Direct application can be useful in a simplified version of crowdsourcing. Instead of tasks having prices associated to them, assume that workers designate the prices for their services and the tasks given to workers can be completed and verifiable *instantaneously*. While the first assumption is reasonable, the second one is unrealistic which requires further attention. *Advisors* in TRSs can be thought as different crowdsourcing markets sharing the reputational information about workers with each other. With these assumptions in place, our work in Chapter 4 can be useful in task delegation with incorporating third-party information under the budgetary constraints.

6.3 Transfer of Trust

One of the main challenges in building trust as detailed in Chapter 2 is *cold-start* (i.e. newcomer) problem. This occurs when agents join to a system (or a team) where they are unknown to other agents. Since "Robots are agents." (Kaminka, 2012), this is a true phenomenon in multi-robot settings as well. Agents leaving/joining the system and creating teams is a fundamental concern for multi-agent systems (MASs), especially for solving complex problems where formation of groups (sometimes called *ad-hoc teams*) (Jennings et al., 1995; Sycara, 1998). Complex problems include *distributed learning* tasks where agents complete tasks together and share the learned rewards via communication (Dutta et al., 2005).

To build trust throughout the formation of teams, it is known that various information sources are used (except direct evidence) in the mitigation of this problem. Different types of information can be collected by various sources (i.e. advisors) and can be fused for trust assessments. While some trust models require opinions to be shared in an *explicit* manner (for instance, Agent A reports that Agent A's interaction with Agent B at the time t is positive.), others use aggregated metrics to be *implicit* (for instance, Agent A reports that Agent B's reputation is 4.85.). There are two main advantages of using implicit approaches: preserving a level of privacy and reducing the amount of communication between agents. The shared metric in implicit approaches can be not only in the forms of a simple metric like a score, it can be, for instance a vector containing locally learned parameters of an advisor. Agents dictated by a shared trust model can share the learned parameters to bootstrap the new agent's trust models.

The process of sharing parameters learned by agents is similar to the problems tackled in Federated Learning (aka. Cooperative Learning). Agents learn from their local observations and communicate with each other (decentralized setting) or with a central

server (i.e. team leader) (centralized setting). Our attack model to analyse vulnerabilities in trust systems can be useful in decentralized setting. The specification of the space of attacks, for instance, introducing a *backdoor* (Bagdasaryan et al., 2020) for a classification task for a single client can be modelled and our methods of searching this space can be applied in this domain.

In online communities, attacks are common. These attacks can be found by so called experts, who learn the underlying formula of the system. Sometimes the system's formulation is public, sometimes it is hidden (i.e. blackbox). In these cases, our model is general enough to represent different types of actions that attacker may use. Although, this requires clear definition of which actions that can be taken by users. To reduce the space complexity, a robust representation is also necessary for practical reasons.

6.4 Multi-Armed Bandits with Informative Arms

The decision-making model that we explained in Chapter 2 is a special case of a *Budget-Limited Multi-Armed Bandit* (BL-MAB), when witness information is incorporated. We used the problem of collecting witness information as a Multi-Armed Bandit problem where the collected rewards influence direct interactions. This special case can be generalized, which would enable a broader set of applications. Our assumption was that the actions that the decision makers take does not change the *system state*. This assumes that the distribution that define agents' behaviour is constant. Consider a BL-MAB problem, where there are a set of *true* arms $k = 1, ..., K$ that return a stochastic reward when pulled and *informative* arms $l = 1, ..., L$ which does not return any reward, rather provide some information about *true* arms. Assume that the number of *true* and *informative* arms are equal and there is one-to-one mapping: for instance, information about arm $k = 1$ can be retrieved when the $l = 1$ is pulled.

The given *informal* BL-MAB model above is a generalization of Chapter 2. In this thesis, we consider the cases where these costs of true arms is higher than informative arms. In addition, pulling an informative arm represent the decision maker purchasing an opinion (or a set of opinions) from advisors, while pulling a true arm represents the decision maker interacting with a provider. The decision process that we introduced uses informative arms when there is not enough information about the true arms. There are some related work in this respect: Pandey et al. (2007) investigates MABs that have some dependency rules between arms (this is similar to a provider behaving in a similar manner to another provider); and Gupta et al. (2019, 2020) assumes that the arms are parametrized with a shared random variable (this is the assumption that the arms are correlated). As a future direction, this generalization would open more applications which include recommender systems and online advertisement selection.

Another interesting path to take is to investigate a variation MAB algorithms that has some similarity our approach is use of *Contextual Bandits* (Slivkins, 2019). In these, a feature vector is observed before each arm is pulled. The reward gained from the arm depends on the feature vector (i.e. context). Rewards are independent and identically distributed (i.e. i.i.d.) according to the feature vector and the arm. While this could be useful to model the trustworthiness of others as the context changes. For example, Agent A trusts Agent B for X type of tasks, not for Y type of tasks, contextual algorithms with a single constrained source would be a better direction for future work (Wu et al., 2015).

The model we proposed is a variation of the BL-MAB model with arms that do not yield an immediate reward. Multi-armed bandits are used in advertising, where if you have a set of arms (advertisements) to show it to the user (you select an arm). Getting information about the user to select an appropriate arm could be costly. There are contextual bandits, these can be budget-limited. Each adverting company can provide a set of features of a customer for a cost. These could be modeled as an arm. Purchasing knowledge for trust (witness information can be also sociological information) can be applied in this direction as well.

6.5 Summary

In this chapter, we explained several directions that our research can be applied alongside with some limitations of our approaches. In particular, we covered the potential applications in automation negotiation, task delegation, transfer of trust between partners and future directions in generalization of our decision-process in a larger context. We think that these are interesting avenues for future research and potentially relevant to other types of challenges in varying multi-agent systems.

Chapter 7

Conclusions

In this thesis, we focused on developing techniques to improve trust systems by the use of strategic behaviours. We investigated how agents may make strategic decisions on whom to trust under budgetary constraints, and how to create complex adversarial behaviours that can hinder this decision-making so that we can gain a level of understanding that is required to be able to establish defenses against them. Although we pointed towards the potential impacts of our research contributions throughout, we further explored their application in a number of domains.

To summarise, the main contributions of this thesis are:

- *A general trust-based decision model under budgetary constraints.* We presented a decision model that is compatible with statistical trust models. This model uses the available budget on acquiring information that varies in trustworthiness and cost. We showed that our model can significantly improve the number of trustworthy engagements when compared to relying on a single source of information. Our results show that our decision-making strategy significantly outperforms *greedy* approaches in constrained environments with third-party information.

- *A mechanism to search strategic attacks.* We have developed a general mechanism to create attackers that are capable of selecting which partners to collude with and devise attacks that are targeted. The mechanism allows for attackers to set different objectives and agnostic to the deployed TRS. In our evaluation, we show the properties that affect the performance of the generated attacks. In terms of injecting of misleading reviews, we found that that attacks found by our search mechanisms are more devastating than the ones that are considered in the literature when both have the same amount of *power* in their attacks.

- *Vulnerability analysis of trust models.* We conducted a rigorous analysis of a number of predominant trust models from the literature to explore the effects of strategic attacks that are generated by our mechanism. We found that vulnerabilities

varied in simulated and real datasets across the trust models in our experiments. The properties we looked at such as connectivity of the system, usage of colluders from inside or outside of the system and power of the attack guided the chosen strategy for the attack. Particularly, we found out that attackers that strategically distribute misleading reviews can increase their rank significantly in all trust models that we tested when compared to promoting themselves directly.

Our research contributions have a significant impact on trust systems where constraints matter. By developing mechanisms to support trust-based decision making in constrained environments, we contributed to provide *favourable* behaviours that satisfy the needs of TRSs designers. By looking into the adversarial behaviours and vulnerabilities that TRSs may have, we contributed to the question of how to devise *unfavourable* behaviours that can influence TRSs. With all these, we believe this simultaneous effort on developing strategic behaviours in TRSs was necessary to answer the next important research question, which is how we can devise a new TRS that can utilize our advancements in both fronts.

Bibliography

A. Abdul-Rahman and S. Hailes. Supporting trust in virtual communities. In *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*, pages 9–pp, 2000.

J. Ahn, D. DeAngelis, and K. S. Barber. Teammate Selection Using Multi-dimensional Trust and Attitude Models. In R. Falcone, S. K. Barber, J. Sabater-Mir, and M. P. Singh, editors, *Trust in Agent Societies*, Lecture Notes in Computer Science, pages 1–24, Berlin, Heidelberg, 2008. Springer. ISBN 978-3-540-92803-4.

E. Alpaydin. Techniques for combining multiple learners. In *Proceedings of Engineering of Intelligent Systems*, pages 6–12. ICSC, 1998.

D. Angluin. Queries and Concept Learning. *Machine Learning*, 2(4):319–342, April 1988. ISSN 1573-0565.

K. J. Åström. Optimal control of markov processes with incomplete state information I. 10:174–205, 1965. ISSN 0022-247X.

T. Baarslag, M. Hendrikx, K. Hindriks, and C. Jonker. Predicting the Performance of Opponent Models in Automated Negotiation. In *2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*, volume 2, pages 59–66, November 2013.

E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov. How To Backdoor Federated Learning. In *International Conference on Artificial Intelligence and Statistics*, pages 2938–2948. PMLR, June 2020.

N. Baracaldo, B. Chen, H. Ludwig, A. Safavi, and R. Zhang. Detecting Poisoning Attacks on Machine Learning in IoT Environments. In *2018 IEEE International Congress on Internet of Things (ICIOT)*, pages 57–64, July 2018.

N. Baracaldo, B. Chen, H. Ludwig, and J. A. Safavi. Mitigating Poisoning Attacks on Machine Learning Models: A Data Provenance Based Approach. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, AISec '17, pages 103–110, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-5202-4.

K. S. Barber and J. Kim. Soft Security: Isolating Unreliable Agents from Society. In R. Falcone, S. Barber, L. Korba, and M. Singh, editors, *Trust, Reputation, and Security: Theories and Practice*, Lecture Notes in Computer Science, pages 224–233, Berlin, Heidelberg, 2003. Springer. ISBN 978-3-540-36609-6.

M. Barreno, B. Nelson, A. D. Joseph, and J. D. Tygar. The security of machine learning. *Machine Learning*, 81(2):121–148, November 2010. ISSN 1573-0565.

M. Barreno, B. Nelson, R. Sears, A. D. Joseph, and J. D. Tygar. Can Machine Learning Be Secure? In *Proceedings of the 2006 ACM Symposium on Information, Computer and Communications Security*, ASIACCS '06, pages 16–25, New York, NY, USA, 2006. ACM. ISBN 978-1-59593-272-3.

C. Bertocco and C. Ferrari. Context-Dependent Reputation Management for Soft Security in Multi Agent Systems. In *2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, volume 3, pages 77–81, December 2008.

A. J. Bidgoly and B. T. Ladani. Modeling and quantitative verification of trust systems against malicious attackers. *The Computer Journal*, 59(7):1005–1027, 2016.

B. Biggio, B. Nelson, and P. Laskov. Poisoning Attacks against Support Vector Machines. *arXiv:1206.6389 [cs, stat]*, June 2012.

B. Biggio and F. Roli. Wild patterns: Ten years after the rise of adversarial machine learning. *Pattern Recognition*, 84:317–331, December 2018. ISSN 0031-3203.

V. Botelho, F. Enembreck, B. C. Avila, H. de Azevedo, and E. E. Scalabrin. Encrypted certified trust in multi-agent system. In *2009 13th International Conference on Computer Supported Cooperative Work in Design*, pages 227–232, April 2009.

D. Bromley. *Reputation, Image, and Impression Management.* John Wiley & Sons, 1996. ISBN 0-471-93869-6 (Hardcover).

M. Brundage, S. Avin, J. Clark, H. Toner, P. Eckersley, B. Garfinkel, A. Dafoe, P. Scharre, T. Zeitzoff, B. Filar, et al. The malicious use of artificial intelligence: Forecasting, prevention, and mitigation. *arXiv preprint arXiv:1802.07228*, 2018.

S. Bubeck, R. Munos, G. Stoltz, and C. Szepesvári. X-Armed Bandits. *Journal of Machine Learning Research*, 12(May):1655–1695, 2011. ISSN ISSN 1533-7928.

C. Burnett, T. J. Norman, and K. Sycara. Bootstrapping trust evaluations through stereotypes. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: Volume 1 - Volume 1*, AAMAS '10, pages 241–248, Richland, SC, 2010. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 978-0-9826571-1-9.

C. Burnett, T. J. Norman, and K. Sycara. Stereotypical trust and bias in dynamic multiagent systems. *ACM Transactions on Intelligent Systems and Technology*, 4(2): 1–22, 2013. ISSN 21576904.

C. Burnett. *Trust Assessment and Decision-Making in Dynamic Multi-Agent Systems.* PhD thesis, University of Aberdeen, 2011.

V. Buskens. The social structure of trust. *Social Networks*, 20(3):265–289, 1998. ISSN 03788733.

J. Carter and A. A. Ghorbani. Value centric trust in multiagent systems. In *Proceedings IEEE/WIC International Conference on Web Intelligence (WI 2003)*, pages 3–9, October 2003.

M. Celetani, D. Fudenberg, D. K. Levine, W. Pesendorfer, S. Econometrica, and N. May. Maintaining a reputation against a long-lived opponent. *Econometrica*, 64(3):691–704, May 1996. ISSN 0012-9682.

G. Chalkiadakis and C. Boutilier. Coordination in multiagent reinforcement learning: A Bayesian approach. In *Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems*, AAMAS '03, pages 709–716, New York, NY, USA, July 2003. Association for Computing Machinery. ISBN 978-1-58113-683-8.

K. Chhogyal, A. Nayak, A. Ghose, and H. K. Dam. A Value-based Trust Assessment Model for Multi-agent Systems. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, pages 194–200, Macao, China, August 2019. International Joint Conferences on Artificial Intelligence Organization. ISBN 978-0-9992411-4-1.

J.-H. Cho, K. Chan, and S. Adali. A survey on trust modeling. *ACM Computing Surveys*, 48(2):1–40, 2015. ISSN 03600300.

J. Clayton. Amazon's murky world of one-star reviews. *BBC News*, September 2020.

O. Dain and R. K. Cunningham. Fusing A Heterogeneous Alert Stream Into Scenarios. In D. Barbara and S. Jajodia, editors, *Applications of Data Mining in Computer Security*, Advances in Information Security, pages 103–122. Springer US, Boston, MA, 2002. ISBN 978-1-4615-0953-0.

W. Ding, T. Qin, X.-D. Zhang, and T.-Y. Liu. Multi-armed bandit with budget constraint and variable costs. *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence*, pages 232–238, 2013.

A. Dorri, S. S. Kanhere, and R. Jurdak. Multi-Agent Systems: A Survey. *IEEE Access*, 6:28573–28593, 2018. ISSN 2169-3536.

J. R. Douceur. The sybil attack. *Peer-to-peer Systems*, pages 1–6, 2002. ISSN 00278424.

P. S. Dutta, N. R. Jennings, and L. Moreau. Cooperative Information Sharing to Improve Distributed Learning in Multi-Agent Systems. *Journal of Artificial Intelligence Research*, 24:407–463, October 2005. ISSN 1076-9757.

J. Emont and C. Bürge. How Scammers in China Manipulate Amazon. https://www.wsj.com/articles/how-scammers-in-china-manipulate-amazon-11545044402, December 2018.

R. Falcone and C. Castelfranchi. Social Trust: A Cognitive Approach. In C. Castelfranchi and Y.-H. Tan, editors, *Trust and Deception in Virtual Societies*, pages 55–90. Springer Netherlands, Dordrecht, 2001. ISBN 978-90-481-5687-0 978-94-017-3614-5.

H. Fang, Y. Bao, and J. Zhang. Misleading opinions provided by advisors: Dishonesty or subjectivity. In *Twenty-Third International Joint Conference on Artificial Intelligence*. Citeseer, 2013.

S. Fatima, S. Kraus, and M. Wooldridge. *Principles of Automated Negotiation*. Cambridge University Press, USA, 1st edition, 2014. ISBN 978-1-107-00254-8.

K. K. Fullam, T. B. Klos, G. Muller, J. Sabater, A. Schlosser, Z. Topol, K. S. Barber, J. S. Rosenschein, L. Vercouter, and M. Voss. A Specification of the Agent Reputation and Trust (ART) Testbed: Experimentation and Competition for Trust in Agent Societies. In *Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems*, AAMAS '05, pages 512–518, New York, NY, USA, 2005. ACM. ISBN 978-1-59593-093-4.

L. Gasser. Social conceptions of knowledge and action: DAI foundations and open systems semantics. *Artificial Intelligence*, 47(1-3):107–138, January 1991. ISSN 0004-3702.

B. Gaudenzi and A. Borghesi. Managing risks in the supply chain using the AHP method. *The International Journal of Logistics Management*, 17(1):114–136, January 2006. ISSN 0957-4093.

R. Ghanea-Hercock. Dynamic trust formation in multi-agent systems. In *Tenth International Workshop on Trust in Agent Societies at the Autonomous Agents and Multi-Agent Systems Conference (AAMAS 2007), Hawaii*, 2007.

P. Godefroid, M. Y. Levin, and D. Molnar. Automated Whitebox Fuzz Testing. In *NDSS*, volume 8, pages 151–166, 2008.

N. Goel and B. Faltings. Deep Bayesian Trust: A Dominant and Fair Incentive Mechanism for Crowd. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33 (01):1996–2003, July 2019. ISSN 2374-3468.

J. Gordon and E. H. Shortliffe. The dempster-shafer theory of evidence. *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project*, 3:832–838, 1984.

R. L. Graham, D. E. Knuth, O. Patashnik, and S. Liu. Concrete mathematics: A foundation for computer science. *Computers in Physics*, 3(5):106–107, 1989.

J. Granatyr, V. Botelho, O. R. Lessing, E. E. Scalabrin, J.-P. Barthès, and F. Enembreck. Trust and Reputation Models for Multiagent Systems. *ACM Computing Surveys*, 48 (2):1–42, October 2015. ISSN 03600300.

N. Griffiths. Task delegation using experience-based multi-dimensional trust. In *Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems*, AAMAS '05, pages 489–496, New York, NY, USA, 2005. ACM. ISBN 978-1-59593-093-4.

N. Griffiths. A fuzzy approach to reasoning with trust, distrust and insufficient trust. *Cooperative Information Agents X, Lecture Notes in Computer Science*, pages 360–374, 2006. ISSN 03029743.

T. D. Gunes, E. Arditi, and R. Aydogan. Collective voice of experts in multilateral negotiation. In *Proceedings of the 20th International Conference on Principles and Practice of Multi-Agent Systems (PRIMA2017)*, 2017.

T. D. Güneş, T. J. Norman, and L. Tran-Thanh. Budget Limited Trust-Aware Decision Making. In G. Sukthankar and J. A. Rodriguez-Aguilar, editors, *Autonomous Agents and Multiagent Systems*, Lecture Notes in Computer Science, pages 101–110, Cham, 2017. Springer International Publishing. ISBN 978-3-319-71679-4.

S. Gupta, S. Chaudhari, G. Joshi, and O. Yağan. Multi-Armed Bandits with Correlated Arms. *arXiv:1911.03959 [cs, stat]*, June 2020.

S. Gupta, G. Joshi, and O. Yağan. Correlated Multi-armed Bandits with a Latent Random Source. *arXiv:1808.05904 [cs, stat]*, January 2019.

D. Heckerman. A Tutorial on Learning with Bayesian Networks. In D. E. Holmes and L. C. Jain, editors, *Innovations in Bayesian Networks: Theory and Applications*, Studies in Computational Intelligence, pages 33–82. Springer, Berlin, Heidelberg, 2008. ISBN 978-3-540-85066-3.

K. Hindriks, C. M. Jonker, and D. Tykhonov. The Benefits of Opponent Models in Negotiation. In *Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology - Volume 02*, WI-IAT '09, pages 439–444, USA, September 2009. IEEE Computer Society. ISBN 978-0-7695-3801-3.

K. Hindriks and D. Tykhonov. Opponent modelling in automated multi-issue negotiation using Bayesian learning. In *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 1*, AAMAS '08, pages 331–338, Richland, SC, May 2008. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 978-0-9817381-0-9.

C.-j. Ho, J. W. Vaughan, Y. Zhang, and M. V. D. Schaar. Towards social norm design for crowdsourcing markets. In *In AAAI Workshops*, 2012.

K. Hoffman, D. Zage, and C. Nita-Rotaru. A survey of attack and defense techniques for reputation systems. *ACM Computing Surveys*, 42(1):1–31, December 2009. ISSN 03600300.

M. Hoogendoorn, S. W. Jaffry, and J. Treur. Exploration and exploitation in adaptive trust-based decision making in dynamic environments. In *Proceedings - 2010 IEEE/WIC/ACM International Conference on Intelligent Agent Technology, IAT 2010*, volume 2, pages 256–260. IEEE, August 2010. ISBN 978-0-7695-4191-4.

D. Hume. A treatise of human nature [1739]. *British Moralists*, pages 1650–1800, 1978.

T. D. Huynh, N. R. Jennings, and N. Shadbolt. FIRE: An Integrated Trust and Reputation Model for Open Multi-Agent Systems. In *16th European Conference on Artificial Intelligence*, pages 18–22, 2004.

T. D. Huynh, N. R. Jennings, and N. R. Shadbolt. An integrated trust and reputation model for open multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, 13(2):119–154, 2006. ISSN 13872532.

L. Ilany and Y. Gal. Algorithm selection in bilateral negotiation. *Autonomous Agents and Multi-Agent Systems*, 30(4):697–723, July 2016. ISSN 1387-2532, 1573-7454.

K. E. Iverson. A programming language. In *Proceedings of the May 1-3, 1962, Spring Joint Computer Conference*, AIEE-IRE '62 (Spring), pages 345–351, New York, NY, USA, May 1962. Association for Computing Machinery. ISBN 978-1-4503-7875-8.

N. R. Jennings. Commitments and Conventions: The Foundation of Coordination in Multi-Agent Systems. *The Knowledge Engineering Review*, 8(3):223–250, 1993.

N. R. Jennings, J. Corera, and I. Laresgoiti. Developing industrial multi-agent systems (invited paper). In *1st Int. Conf. on Multi-Agent Systems (ICMAS '95) (11/06/95 - 13/06/95)*, pages 423–430, 1995.

N. R. Jennings, P. Faratin, A. R. Lomuscio, S. Parsons, C. Sierra, and M. Wooldridge. Automated negotiation: Prospects, methods and challenges. *International Journal of Group Decision and Negotiation*, 10(2):199–215, 2001.

S. Jiang, J. Zhang, and Y.-S. Ong. An evolutionary model for constructing robust trust networks. In *Proceedings of the 2013 International Conference on Autonomous Agents and Multi-Agent Systems*, AAMAS '13, pages 813–820, Richland, SC, May 2013. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 978-1-4503-1993-5.

A. Jøsang and R. Ismail. The beta reputation system. In *Proc. 15th Bled Electronic Commerce Conference*, volume 5, pages 2502–2511, 2002.

A. Jøsang. Robustness of trust and reputation systems: Does it matter? In *IFIP International Conference on Trust Management*, pages 253–262, 2012.

A. Jøsang. *Subjective Logic*. Artificial Intelligence: Foundations, Theory, and Algorithms. Springer International Publishing, Cham, 2016. ISBN 978-3-319-42335-7 978-3-319-42337-1.

A. Jøsang and J. Golbeck. Challenges for robust trust and reputation systems. In *Proceedings of the 5th International Workshop on Security and Trust Management (SMT 2009), Saint Malo, France*, page 52, 2009.

A. Josang and J. Haller. Dirichlet Reputation Systems. In *The Second International Conference on Availability, Reliability and Security (ARES'07)*, pages 112–119, April 2007.

L. P. Kaelbling, M. L. Littman, and A. W. Moore. Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285, 1996.

G. A. Kaminka. I Have a Robot, and I'm Not Afraid to Use It! *AI Magazine*, 33(3): 66–66, September 2012. ISSN 2371-9621.

S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina. The Eigentrust Algorithm for Reputation Management in P2P Networks. In *Proceedings of the 12th International Conference on World Wide Web*, WWW '03, pages 640–651, New York, NY, USA, 2003. ACM. ISBN 978-1-58113-680-7.

Z. Karnin, T. Koren, and O. Somekh. Almost optimal exploration in multi-armed bandits. In *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*, ICML'13, pages III–1238–III–1246, Atlanta, GA, USA, June 2013. JMLR.org.

J. Kelleher and B. O'Sullivan. Generating All Partitions: A Comparison Of Two Encodings. *arXiv:0909.2331 [cs, math]*, May 2014.

H. Kellerer, U. Pferschy, and D. Pisinger. Multidimensional knapsack problems. In *Knapsack Problems*, pages 235–283. Springer, 2004.

R. Kerr and R. Cohen. Modeling trust using transactional, numerical units. In *Proc. PST*, pages 1–11, 2006. ISBN 1-59593-604-1.

R. Kerr and R. Cohen. Smart cheaters do prosper: Defeating trust and reputation systems. In *Proc. AAMAS*, pages 993–1000, 2009. ISBN 978-0-9817381-7-8.

L. Klejnowski, Y. Bernard, J. Hahner, and C. Muller-Schloer. An architecture for trust-adaptive agents. In *2010 Fourth IEEE International Conference on Self-Adaptive and Self-Organizing Systems Workshop*, pages 178–183, September 2010.

A. Kleywegt, A. Shapiro, and T. Homem-de-Mello. The Sample Average Approximation Method for Stochastic Discrete Optimization. *SIAM Journal on Optimization*, 12(2): 479–502, January 2002. ISSN 1052-6234.

M. J. Kochenderfer. *Decision Making under Uncertainty: Theory and Application*. MIT press, 2015.

P. W. Koh and P. Liang. Understanding Black-box Predictions via Influence Functions. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pages 1885–1894. JMLR.org, 2017.

N. Konstantinov and C. Lampert. Robust Learning from Untrusted Sources. *arXiv:1901.10310 [cs, stat]*, January 2019.

S. Kraus. Automated Negotiation and Decision Making in Multiagent Environments. In G. Goos, J. Hartmanis, J. van Leeuwen, M. Luck, V. Mařík, O. Štěpánková, and R. Trappl, editors, *Multi-Agent Systems and Applications*, volume 2086, pages 150–172. Springer Berlin Heidelberg, Berlin, Heidelberg, 2001. ISBN 978-3-540-42312-6 978-3-540-47745-7.

J. Lazzaro, S. Ryckebusch, M. A. Mahowald, and C. A. Mead. Winner-take-all networks of O (n) complexity. In *Advances in Neural Information Processing Systems*, pages 703–711, 1989.

J. Leskovec and A. Krevl. SNAP Datasets: Stanford large network dataset collection. June 2014.

J. Lewis and A. Weigert. Trust as a social reality. *Social Forces*, 63(4):967–985, June 1985. ISSN 00377732.

K. Leyton-Brown, E. Nudelman, G. Andrew, J. McFadden, and Y. Shoham. A portfolio approach to algorithm selection. In *IJCAI*, volume 1543, page 2003, 2003.

N. Littlestone and M. K. Warmuth. The Weighted Majority Algorithm. *Information and Computation*, 108(2):212–261, February 1994. ISSN 0890-5401.

J. S. Liu. The Collapsed Gibbs Sampler in Bayesian Computations with Applications to a Gene Regulation Problem. *Journal of the American Statistical Association*, 89 (427):958–966, September 1994. ISSN 0162-1459.

X. Liu and A. Datta. Modeling context aware dynamic trust using hidden markov model. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence (AAAI'12)*, pages 1938–1944, 2012.

S. P. Marsh. *Formalising Trust as a Computational Concept*. PhD thesis, University of Stirling, 1994.

P. D. Meo, K. Musial-Gabrys, D. Rosaci, G. M. L. Sarnè, and L. Aroyo. Using Centrality Measures to Predict Helpfulness-Based Reputation in Trust Networks. *ACM Transactions on Internet Technology (TOIT)*, 17(1):8:1–8:20, February 2017. ISSN 1533-5399.

T. Muller, D. Wang, Y. Liu, and J. Zhang. How to use information theory to mitigate unfair rating attacks. In *Trust Management X*, pages 17–32. Springer, 2016. ISBN 978-3-319-41354-9.

V. Muñoz, J. Murillo, B. López, and D. Busquets. Strategies for exploiting trust models in competitive multi-agent systems. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 5774 LNAI:79–90, 2009. ISSN 03029743.

J. Navarro, A. Deruyver, and P. Parrend. A systematic survey on multi-step attack detection. *Computers & Security*, 76:214–249, July 2018. ISSN 01674048.

A. Nijenhuis and H. S. Wilf. *Combinatorial Algorithms: For Computers and Calculators*. Elsevier, May 2014. ISBN 978-1-4832-7345-7.

S. Onn. *Nonlinear Discrete Optimization: An Algorithmic Theory*. European Mathematical Society, Zürich, 2010. ISBN 978-3-03719-093-7.

S. Pandey, D. Chakrabarti, and D. Agarwal. Multi-armed bandit problems with dependent arms. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, pages 721–728, New York, NY, USA, June 2007. Association for Computing Machinery. ISBN 978-1-59593-793-3.

E. Parhizkar, M. H. Nikravan, and S. Zilles. Indirect Trust is Simple to Establish. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, pages 3216–3222, Macao, China, August 2019. International Joint Conferences on Artificial Intelligence Organization. ISBN 978-0-9992411-4-1.

I. Pinyol, J. Sabater-Mir, P. Dellunde, and M. Paolucci. Reputation-based decisions for logic-based cognitive agents. *Autonomous Agents and Multi-Agent Systems*, 24(1): 175–216, January 2012. ISSN 1573-7454.

M. Piunti, M. Venanzi, R. Falcone, and C. Castelfranchi. Multimodal trust formation with uninformed cognitive maps (uncm). In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 3*, pages 1241–1242, 2012.

A. Prasad, A. S. Suggala, S. Balakrishnan, and P. Ravikumar. Robust Estimation via Robust Gradient Estimation. *arXiv:1802.06485 [cs, stat]*, February 2018.

J. W. Pratt, H. Raiffa, R. O. Schlaifer, R. Schlaifer, et al. *Introduction to Statistical Decision Theory*. MIT press, 1995.

S. D. Ramchurn, D. Huynh, and N. R. Jennings. Trust in multi-agent systems. *The Knowledge Engineering Review*, 19(1):1–25, March 2004a. ISSN 1469-8005, 0269-8889.

S. D. Ramchurn, N. R. Jennings, C. Sierra, and L. Godo. Devising a Trust Model for Multi-Agent Interactions Using Confidence and Reputation. *Applied Artificial Intelligence*, 18(9-10):833–852, October 2004b. ISSN 0883-9514.

K. Regan, P. Poupart, and R. Cohen. Bayesian Reputation Modeling in E-marketplaces Sensitive to Subjectivity, Deception and Change. In *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 2*, AAAI'06, pages 1206–1212. AAAI Press, 2006a. ISBN 978-1-57735-281-5.

K. Regan, P. Poupart, and R. Cohen. Bayesian reputation modeling in e-marketplaces sensitive to subjectivity, deception and change. In *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 2*, AAAI'06, pages 1206–1212, Boston, Massachusetts, 2006b. AAAI Press. ISBN 978-1-57735-281-5.

J. R. Rice. The Algorithm Selection Problem. In *Advances in Computers*, volume 15, pages 65–118. Elsevier, 1976. ISBN 978-0-12-012115-1.

S. L. Robinson. Trust and Breach of the Psychological Contract. *Administrative Science Quarterly*, 41(4):574–599, 1996. ISSN 0001-8392.

Y. Ruan and A. Durresi. A survey of trust management systems for online social communities – Trust modeling, trust inference and attacks. *Knowledge-Based Systems*, 106:150–163, August 2016. ISSN 0950-7051.

J. Sabater and C. Sierra. REGRET: Reputation in gregarious societies. In *Proceedings of the Fifth International Conference on Autonomous Agents*, AGENTS '01, pages 194–195, New York, NY, USA, May 2001. Association for Computing Machinery. ISBN 978-1-58113-326-4.

J. Sabater and C. Sierra. Review on computational trust and reputation models. *Artificial Intelligence Review*, 24(1):33–60, 2005. ISSN 02692821.

S. Salah, G. Macia-Fernandez, and J. E. Diaz-Verdejo. A model-based survey of alert correlation techniques. *Computer Networks*, 57(5):1289–1317, April 2013. ISSN 1389-1286.

J. Schneider. Active optimization and self driving cars. 2017.

S. Sen. A comprehensive approach to trust management. In *Proceedings of the 2013 International Conference on Autonomous Agents and Multi-Agent Systems*, AAMAS '13, pages 797–800, Richland, SC, 2013. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 978-1-4503-1993-5.

S. Sen, A. Ridgway, and M. Ripley. Adaptive Budgeted Bandit Algorithms for Trust Development in a Supply-Chain. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '15, pages 137–144, Istanbul, Turkey, May 2015a. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 978-1-4503-3413-6.

S. Sen, A. Ridgway, and M. Ripley. Adaptive budgeted bandit algorithms for trust in a supply-chain setting. *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, pages 137–144, 2015b. ISSN 15582914.

M. Şensoy, J. Zhang, P. Yolum, and R. Cohen. POYRAZ: Context-Aware Service Selection Under Deception. *Computational Intelligence*, 25(4):335–366, November 2009. ISSN 08247935, 14678640.

S. Seuken and D. C. Parkes. Sybil-proof accounting mechanisms with transitive trust. In *Proceedings of the 2014 International Conference on Autonomous Agents and Multi-Agent Systems*, AAMAS '14, pages 205–212, Richland, SC, May 2014. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 978-1-4503-2738-1.

G. Shafer. *A Mathematical Theory of Evidence*. Princeton University Press, April 1976. ISBN 978-0-691-10042-5.

S. S. Shapiro and M. B. Wilk. An analysis of variance test for normality (complete samples). *Biometrika*, 52(3-4):591–611, December 1965. ISSN 0006-3444.

S. Sirur and T. Muller. The Reputation Lag Attack. In W. Meng, P. Cofta, C. D. Jensen, and T. Grandison, editors, *Trust Management XIII*, IFIP Advances in Information and Communication Technology, pages 39–56, Cham, 2019. Springer International Publishing. ISBN 978-3-030-33716-2.

A. Slivkins. Introduction to Multi-Armed Bandits. *arXiv:1904.07272 [cs, stat]*, September 2019.

A. Slivkins and J. W. Vaughan. Online decision making in crowdsourcing markets: Theoretical challenges. *ACM SIGecom Exchanges*, 12(2):4–23, November 2014.

M. Srivatsa, L. Xiong, and L. Liu. TrustGuard: Countering vulnerabilities in reputation management for decentralized overlay networks. In *Proceedings of the 14th International Conference on World Wide Web*, WWW '05, pages 422–431, Chiba, Japan, May 2005. Association for Computing Machinery. ISBN 978-1-59593-046-0.

J. M. Such. Attacks and Vulnerabilities of Trust and Reputation Models. In S. Ossowski, editor, *Agreement Technologies*, Law, Governance and Technology Series, pages 467–477. Springer Netherlands, Dordrecht, 2013. ISBN 978-94-007-5583-3.

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

J. Surowiecki. *The Wisdom of Crowds*. Anchor, 2005.

A. Sutcliffe and D. Wang. Computational modelling of trust and social relationships. *Journal of Artificial Societies and Social Simulation*, 15(1):3, 2010. ISSN 1460-7425.

R. S. Sutton, A. G. Barto, et al. *Introduction to Reinforcement Learning*, volume 135. MIT press Cambridge, 1998.

K. P. Sycara. Multiagent systems. *AI Magazine*, 19(2):79, June 1998.

W. T. L. Teacy, G. Chalkiadakis, A. Rogers, and N. R. Jennings. Sequential decision making with untrustworthy service providers. *Proceedings of the 7th International Conference on Autonomous Agents and Multiagent Systems*, pages 755–762, 2008.

W. T. L. Teacy, J. Patel, N. R. Jennings, and M. Luck. TRAVOS: Trust and reputation in the context of inaccurate information sources. *Autonomous Agents and Multi-Agent Systems*, 12(2):183–198, 2006. ISSN 13872532.

W. L. Teacy, M. Luck, A. Rogers, and N. R. Jennings. An efficient and versatile approach to trust and reputation using hierarchical bayesian modelling. *Artificial Intelligence*, 193:149–185, December 2012. ISSN 0004-3702.

L. Tran-Thanh. *Budget-Limited Multi-Armed Bandits*. PhD thesis, University of Southampton, 2012.

L. Tran-Thanh, A. Chapman, E. M. de Cote, A. Rogers, and N. R. Jennings. Epsilon–First Policies for Budget–Limited Multi-Armed Bandits. In *Twenty-Fourth AAAI Conference on Artificial Intelligence*, July 2010.

R. E. Walpole and R. H. Myers. *Probability & Statistics for Engineers & Scientists*. Pearson Education Limited, 2012.

D. Wang, T. Muller, Y. Liu, and J. Zhang. Towards robust and effective trust management for security: A survey. In *Proc. PST*, pages 511–518, 2014. ISBN 978-1-4799-6513-7.

D. Wang, T. Muller, J. Zhang, and Y. Liu. Quantifying robustness of trust systems against collusive unfair rating attacks using information theory. In *Proc. IJCAI*, pages 111–117, 2015a. ISBN 978-1-57735-738-4.

D. Wang, T. Muller, A. A. Irissappane, J. Zhang, and Y. Liu. Using Information Theory to Improve the Robustness of Trust Systems. page 9, 2015b.

D. Wang, T. Muller, J. Zhang, and Y. Liu. Is It Harmful When Advisors Only Pretend to Be Honest? In *Thirtieth AAAI Conference on Artificial Intelligence*, March 2016.

A. Whitby, A. Jøsang, and J. Indulska. Filtering out unfair ratings in Bayesian reputation systems. In *Proc. 7th Int. Workshop on Trust in Agent Societies*, pages 106–117, 2004.

H. Wu, R. Srikant, X. Liu, and C. Jiang. Algorithms with Logarithmic or Sublinear Regret for Constrained Contextual Bandits. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 433–441. Curran Associates, Inc., 2015.

Z. Wu and M. Pagell. Balancing priorities: Decision-making in sustainable supply chain management. *Journal of Operations Management*, 29(6):577–590, September 2011. ISSN 0272-6963.

Y. Xia, T. Qin, W. Ma, N. Yu, and T. Y. Liu. Budgeted multi-armed bandits with multiple plays. In *IJCAI'06: Proceedings of International Joint Conference on Artificial Intelligence*, volume 2016-Janua, pages 2210–2216, 2016.

H. Yu, C. Miao, Z. Shen, and C. Leung. Quality and Budget Aware Task Allocation for Spatial Crowdsourcing. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '15, pages 1689–1690, Richland, SC, May 2015. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 978-1-4503-3413-6.

H. Yu, Z. Shen, C. Leung, C. Miao, and V. R. Lesser. A Survey of Multi-Agent Trust Management Systems. *IEEE Access*, 1:35–50, 2013a. ISSN 2169-3536.

H. Yu, Z. Shen, C. Miao, and B. An. A reputation-aware decision-making approach for improving the efficiency of crowdsourcing systems. In *Proceedings of the 2013 International Conference on Autonomous Agents and Multi-Agent Systems*, AAMAS '13, pages 1315–1316, Richland, SC, May 2013b. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 978-1-4503-1993-5.

Y. Zhang and M. van der Schaar. Reputation-based incentive protocols in crowdsourcing applications. In *2012 Proceedings IEEE INFOCOM*, pages 2140–2148, March 2012.

Y. Zheng, G. Li, Y. Li, C. Shan, and R. Cheng. Truth inference in crowdsourcing: Is the problem solved? *Proceedings of the VLDB Endowment*, 10(5):541–552, January 2017. ISSN 2150-8097.