Trust-based Social Mechanism to Counter Deceptive Behaviour

by

Sarah Niukyun Lim Choi Keung

Thesis

Submitted to the University of Warwick

for the degree of

Doctor of Philosophy

Department of Computer Science

June 2011

Contents

List of Tables

List of Figures

List of Algorithms

Acknowledgments

First, I would like to thank my PhD supervisor, Dr. Nathan Griffiths for his mentorship, guidance, support, and positive criticism throughout. I am also grateful for his organisation of funding for my research through a scholarship from the Department of Computer Science at the University of Warwick - without it, this work would not have been possible.

My thanks also go to the members of the Intelligent and Adaptive Systems Group for their interest in my research and for giving me valuable feedback. Special thanks to my labmates for their companionship, interesting discussions, and endless cups of tea. I would also like to extend my appreciation to the numerous persons I have worked with and befriended in the Department of Computer Science and elsewhere at the University of Warwick. I am also thankful to my friends, in and out of university for their friendship and support. My thanks also go to my colleagues at the University of Birmingham for their support.

I express my gratitude to my family for their love and support. Special thanks to Maurice for his dedication and patience. Finally, I dedicate this work to my parents, who have always believed in me and have been my source of courage and focus.

Declarations

The work in this thesis has been undertaken by myself and has not been submitted to any other application for a degree. Parts of this thesis have been previously published. Chapter 4 describes the trust and reputation model and has been presented in workshop papers and an extended paper [69, 71, 70]. Parts of Chapter 5 and Chapter 6 related to the trust-based social mechanism, particularly the agent network model building component, have been presented in conference and workshop papers [72, 73]. Finally, parts of Chapter 2 describing the trust and reputation approaches in the literature have been published as a book chapter [74].

Abstract

The actions of an autonomous agent are driven by its individual goals and its knowledge and beliefs about its environment. As agents can be assumed to be self-interested, they strive to achieve their own interests and therefore their behaviour can sometimes be difficult to predict. However, some behaviour trends can be observed and used to predict the future behaviour of agents, based on their past behaviour. This is useful for agents to minimise the uncertainty of interactions and ensure more successful transactions. Furthermore, uncertainty can originate from malicious behaviour, in the form of collusion, for example. Agents need to be able to cope with this to maximise their benefits and reduce poor interactions with collusive agents. This thesis provides a mechanism to support countering deceptive behaviour by enabling agents to model their agent environment, as well as their trust in the agents they interact with, while using the data they already gather during routine agent interactions.

As agents interact with one another to achieve the goals they cannot achieve alone, they gather information for modelling the trust and reputation of interaction partners. The main aim of our trust and reputation model is to enable agents to select the most trustworthy partners to ensure successful transactions, while gathering a rich set of interaction and recommendation information. This rich set of information can be used for modelling the agents' social networks. Decentralised systems allow agents to control and manage their own actions, but this suffers from limiting the agents' view to only local interactions. However, the representation of the social networks helps extend an agent's view and thus extract valuable information from its environment. This thesis presents how agents can build such a model of their agent networks and use it to extract information for analysis on the issue of collusion detection.

Chapter 1

Introduction

Agent technology is a mature paradigm that allows computer systems to behave autonomously within their environment, and act on it to influence their current and future outcomes [136, 29]. Over the last 15 years, intelligent agents have become increasingly popular with applications in various domains, such as industry and manufacturing (e.g. control systems and supply chains) [63, 96], commerce (e.g. e-commerce, entertainment, telecommunications and healthcare) [27, 6, 48, 87] and simulation (e.g. military training, environmental changes and customer behaviour) [77].

Agent-based systems have enormous promise, but there are a number of challenges that must be overcome for them to fulfil their maximum potential. One important challenge is that of managing the risk, and the inherent uncertainty involved when autonomous agents interact. Trust is often considered to provide a means of managing this risk. Luhmann [80, 79] views trust as an attitude that allows for risk-taking decisions, hence the close relationship between risk and trust. Gambetta defines trust as the level of subjective probability with which an agent assesses that another agent or group of agents will perform a particular action [31]. In this thesis agents use the notion of trust for reasoning about other agents with whom they interact. Trust is especially important in the context of open distributed systems, where malicious behaviour may affect the actions of agents. The ability of agents to consider trust is also expressed as one of the requirements for developing the reasoning capabilities of agents, especially those situated in open environments [77].

1.1 The Problem

The way agents represent and assess trust in others is crucial for interactions, as decision making ultimately relies on this trust assessment, and it needs to take into account the environment in which agents evolve and the dynamic nature of behaviour. While trust-based agent systems aim to tackle the issue of uncertainty, many of the individual aspects related to trust have not been considered together. Consequently, we need trust models that integrate these aspects, in order for trust to be relied upon for accurate decision making. In this thesis, we concentrate on enhancing the ability of agents to detect malicious behaviour, to better inform their decision making for future agent interactions.

Research in agent technology, since the 1990s, has been driven by agent-related technologies, such as Internet technologies, peer-to-peer (P2P) systems, service-oriented technologies, pervasive computing, Web services and Grid computing, which provide essential infrastructure for the development of agent systems. As part of the long-term future of agent-based systems, it is projected that we will see the development of open multi-agent systems (MAS) spanning multiple application domains, and involving heterogeneous participants developed by diverse design teams [77, 78]. In the area of trust and reputation, the long-term future will see trust techniques addressing the issue of malicious agents [78, Figure 7.1], following the development of such technologies as reputation mechanisms. In our research, we aim to contribute to this aspect by considering collusion as a deceptive behaviour.

Till now, most agent systems are not completely autonomous due to the numerous barriers to the wider adoption of agent technology. For instance, in the e-commerce domain, concerns about trust, privacy, security and legal issues are perceived as barriers [27]. Users need to trust that agents will behave as expected, and provide protection of privacy and the assurances similar to traditional trading practices. Subsequently, the significant level of human involvement in agent-based systems suggests that solutions should aim at improving human users' understanding of the agent systems, besides ensuring efficient and successful agent interactions. This thesis also addresses this concern by looking at ways to assist the human designers and analysts of agent-based systems in better understanding how these systems work, in terms of the relationships that exist between agents, both to help towards minimising the uncertainty of interaction, and to further develop agent systems towards increased automation.

1.2 Research Goals

The overarching aim of the research presented in this thesis is to enable agents to detect malicious behaviour in decentralised agent-based systems. This capability helps to minimise the interaction uncertainty among agents. If an agent is able to reduce poor or sub-optimal transactions, it will benefit by achieving its goals more successfully. In open, decentralised and heterogeneous systems, agents differ in their goals, and levels of performance and honesty. A successful agent is one that is able to interact successfully with others, while being able to detect and avoid dishonest or malicious transactions, as well as recovering from poor interactions. The context of this research lies at the individual level for agents in a multi-agent system. While there may be measures in place for secure communications and identity verification, for example, we are concerned with the issue of trust in agent behaviour. We assume that agents have their own management of trustworthiness and behaviour; reputations are shared but there is no central system to reward or punish agent behaviour.

In pursuit of the overall aim of detecting malicious behaviour, we identify four specific objectives on which we focus in this thesis.

- To identify and represent a rich set of information on agent interactions and recommendations to support reasoning about agents, and to provide a mechanism for using this information in assessing trust.

- To provide a means of reasoning about the information identified above to extract further, previously unknown, information about trust, reputation, and the interactions between agents (including third parties), in order to determine the various social networks that exist between agents.

- To identify the types of collusion that exist in an agent's domain by determining the characteristics of interactions and recommendations between agents that define each type.

- To investigate the use of information on interactions, recommendations and the social networks between agents, in supporting collusion detection using existing data mining techniques. The aim is to show that by using a rich set of information to extract knowledge of the relationships between agents, along with having a clear understanding of the types of collusion, we can provide a solid base for tackling the challenging problem of collusion detection.

1.2.1 Improved Trust Assessment

In trust-based agent systems, it is important for agents to be able to assess others' trustworthiness when making interaction decisions, which is motivated by the need for achieving high rates of successful interactions. However, agents are typically situated in environments in which they have limited amounts of information. Therefore, the goal is to maximise the use of the different forms of trust information that is available, namely trust information from direct agent interactions and recommendations from third parties, both direct and indirect, when the opinions are passed along a recommendation chain. In different circumstances, different trust sources may be available and the agent would improve its trust assessment by considering all the trust information sources available and prioritising the most reliable ones. With this richer set of information on agent interactions and recommendations, the agent is better equipped to assess trustworthiness and take decisions accordingly.

1.2.2 Accurate Representation of an Agent's Social Network

An agent may not have a global view of its environment, but an accurate representation of the agents it has some form of interaction with is crucial. A better understanding of how agents are linked to one another can provide valuable insight into agent behaviour, leading to further reduction in interaction uncertainty as agents are better able to select interaction partners. The goal is to enable agents to build and maintain an accurate representation of their social networks, using the rich information gathered on interactions and recommendations. Agents can extract valuable and potentially previously unknown information about agent interactions. This will be useful for agent decision making, as well as enable human analysts to better understand how agent-based systems work, with respect to the relationships among agents.

1.2.3 Identification of Collusion Types

Agents in a multi-agent system have different goals and priorities. A further challenge for agents is to have successful interactions despite some agents behaving maliciously. Malicious behaviour such as collusion, adds uncertainty to agent interactions due to the covert nature of some forms of behaviour. This motivates the goal of enabling agents to identify the various types of collusion which may occur in their domain. The identification of collusion types is essential as a preceding step to collusion detection. The characteristics of interactions and recommendations of each type of collusion enables an agent to differentiate between agent interactions and relationships that can be beneficial or harmful to its own goals.

1.2.4 Supporting Collusion Detection

Collusion is a form of malicious behaviour, where two or more agents agree to behave in such a way as to benefit the colluding group at the expense of other agents. For an agent evaluating the trustworthiness of its potential interaction partners, it is important to take potential collusion into consideration. We aim to support collusion detection using the rich set of information gathered, as well as the representation of the agent's social network by applying known data mining techniques. Collusion detection will enable both agents and human analysts to better inform their decisions as they have information about the agents involved and the type of collusion concerned.

1.3 Contributions

This thesis contributes to the field of multi-agent systems by providing a mechanism that allows agents to make more informed decisions with the aim of minimising the uncertainty of agent interactions, while facing malicious behaviour. This mechanism is motivated by the need to correctly assess the trustworthiness of agents, and make full use of information that can be extracted from knowledge of the agent's social network to support this trust assessment. The contributions of this thesis can be summarised as follows.

- An improved trust assessment technique is presented, based on a richer set of trust-based agent interaction information and recommendations. Our trust model ensures that agents collect sufficient information from different sources about different aspects of services, as well as considering the recency and relevance of the information for trust assessment and future partner selection. We extend the use of multidimensional trust [36] and reputation, considering both direct and indirect recommendations for trust assessment, and use the richness of the recency and the relevance of interactions to achieve a greater accuracy of trust assessment (Chapter 3).

- We provide a technique for obtaining an accurate representation of an agent's social network, based on the rich set of information gathered. Using agent graphs, an agent represents the interaction information and recommendations gathered for convenient reuse and analysis. Agent relationships, including the strength of the links and the services exchanged are clearly represented in the agent graphs. We describe how agents can use trust and reputation information to build and maintain an agent network model (Chapter 4). The representation of an agent's network provides human analysts with the tools to better understand how such agent systems work.

- We present a taxonomy of collusion types, together with their individual characteristics of interaction and recommendations. For the e-commerce domain, we identify the types of collusion that exist (Section 5.4), and this forms an important first step towards collusion detection.

- We show that we can support the detection of Persistent Target-Witness collusion by applying the Cosine similarity measurement technique (Sections 5.5.3 and 5.6). The approach makes use of the agent's social network and information on agents' interactions and recommendations, together with the knowledge of the types of collusion in the domain.

Several trust and reputation models have been proposed to better inform the agent selection process, such as Marsh's trust formalism [81], Castelfranchi and Falcone's socio-cognitive view of trust [12, 25], ReGreT [106] and FIRE [46]. This thesis contributes to the area by integrating and extending the important components from existing models to improve an agent's assessment of trust, by representing interaction information and recommendations. The use of the different trust sources (direct interactions, direct recommendations and indirect recommendations) depending on their availability, and using the most recent and relevant interactions to assess agent trustworthiness, is a new technique that enables agents to accurately assess service provision in various service characteristics important for the agent.

Social networks are popular for searching information, as discussed by Milgram [84] and Kautz *et al.* [56] for example. In agent-based systems, ReGreT and FIRE use social networks to link agents to their interaction partners or to make judgements of neighbouring groups. However, these models do not detail how agents build and maintain these networks. As well as presenting how agent graphs are built and maintained, we also discuss the possible information that can be extracted from these networks (Section 4.3). Agents can use this information in their decision making to better reflect agent behaviour in the environment. Furthermore, human users and system designers can use this information to gain a better understanding of such systems.

Collusion contributes to interaction uncertainty for agents. This issue has been tackled from various perspectives, such as Jurca's incentive-compatible, collusion resistant payment mechanism [52], and TrustGuard's use of transaction proofs against fake transactions [119]. Our contribution to this area is to enable individual agents in decentralised systems to detect certain forms of collusion, from the trust information that they have available and their social network. Trust and reputation information can be used to extend the knowledge an agent has of its environment from its localised interactions, through recommendation chains and indirect information sources. The representation of the agent network aims to assist human analysts to visualise the agent relationships and interactions that may indicate collusive behaviour.

Aspects of the work presented in this thesis have resulted in the following publications. Our trust and reputation model, based on both direct and indirect recommendations, described in Chapter 3, has been published in [69, 70]. The manner in which our model also takes into consideration the recency and relevance of interactions and recommendations to assess the trustworthiness of agents was published in [71]. Our approach for using information about interactions and recommendations to extract agent social networks was published in [72, 73], as described in Chapter 4.

1.4 Overview of Solution Approach

We present an approach to modelling trust, recommendations, and agents' social networks, designed to capture the dynamic behaviours of agents and their interactions to support decision making. The approach consists of three main components: (i) data collection, (ii) network model building, and (iii) analysis of interaction data. The data collection component allows an evaluator agent to gather information about its own interactions with other agents and also from recommendations. The selection of interaction partners is performed using trust, as well as direct and indirect recommendations to better inform decision making [69]. With information gathered from interactions and recommendations, an evaluator can build a representation of its agent network, to include providers, witnesses, and the way they are linked and the strength of their relationships. The third component makes up the analysis of the agent network and interaction data to uncover knowledge of relationships and behaviours, that can be useful for informing decision making.

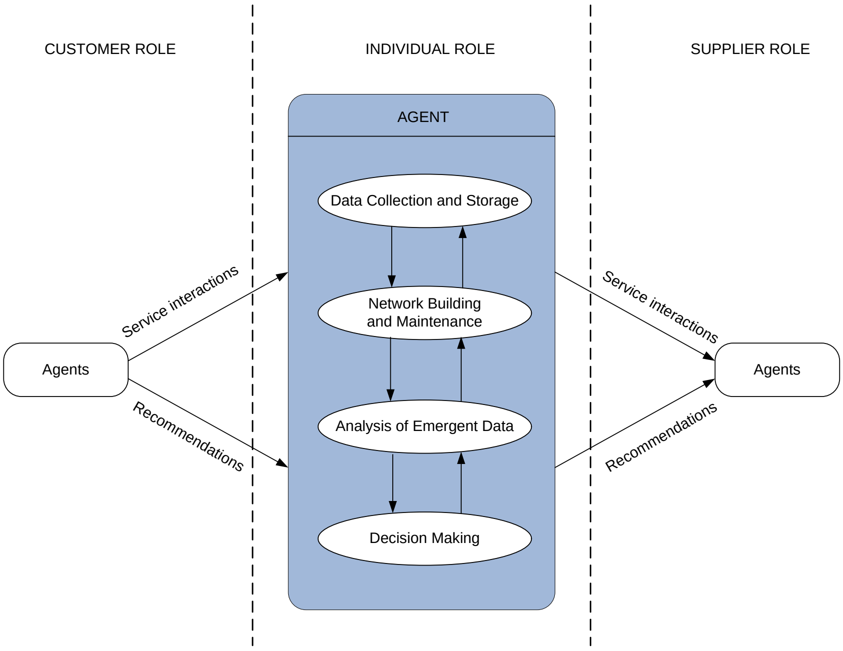


Figure 1.1: Overview of the Approach from the Perspective of a Provider Agent.

An overview of the approach from the perspective of an evaluator agent is shown in Figure 1.1. The agent, in the role of a customer, requires the services of provider agents, and the opinions of witnesses can help in this regard. In the agent environment, the agent also acts as a provider of services and recommendations, based on the analysis of its past history of interactions, recommendations and decision making on interaction choices. These activities are performed by the agent in its individual role, as it interacts with others to guide its decision making on the most appropriate future transaction. Collected data and data from the resulting analysis are transferred to and from the different processes of the individual agent. This ensures that the agent keeps an updated view of its environment with ongoing interactions with others.

1.4.1 Trust and Reputation Model for Data Collection

The evaluator stores a rich set of information to inform future interactions. This is achieved by keeping a history of past interactions with agents, for each type of service that the evaluator requires. When the evaluator needs to find a provider for a service, it searches through its past history and evaluates the trustworthiness of relevant past providers. If there is insufficient information, the evaluator requests the opinions of other agents and bases its decision on their opinion and its own judgement of the witnesses' trustworthiness in giving opinions. As shown in Figure 1.1, service interactions and recommendations are the sources of the data collected by the agent. These are stored in a suitable format for future use.

1.4.2 Agent Network Model Building

Agents are dealing with trust in other agents, similar to the concept of a web of trust. As an evaluator interacts with other agents, including providers and witnesses, it gathers information about interactions and relationships that will help it build a model of the network of agents to have a better understanding of its social network. Graph structures are well suited to represent networks and in this thesis we investigate how they can be used to represent agents' social networks. The agent networks are built and maintained as data is collected when agents interact with one another.

1.4.3 Collusion Detection

The collection of interaction data over the medium to long term enables an agent to make decisions about numerous aspects, particularly with the view to increasing the success of its interactions and maximising its benefits. Besides using trust and reputation to efficiently select interaction partners and witnesses, interaction data, together with agent network details, can bring more insight into other aspects of the agent environment. For instance, agent network information helps in reinforcing trust in the roles of witnesses to give accurate information. In our work, we explore the analysis of information that can be extracted from the agent network to further improve an agent's accuracy in making decisions with regards to interactions with other agents in the environment. We particularly focus on the issue of collusion and its detection.

1.5 Thesis Outline

The outline of the thesis is as follows. Chapter 2 discusses the related work on trust-based social mechanisms for countering deception; the use of trust and reputation for agent selection in existing multi-agent systems; how agents can use their social network; and the issue of deception in agent-based systems. The overall mechanism provides the three components that an agent uses to gather trust and reputation information, build a model of its agent network from this information, and analyse the network to extract valuable information, such as details of potentially colluding agents. These components all help towards the agent's decision making. Chapters 3, 4, and 5 detail the three components and make up the core of the thesis. They introduce the mechanisms that an agent uses to inform its decision making, and outcomes from each component feed into the next component and thus form a cycle whereby the agent's decisions affect its future actions. Chapter 3 presents our trust and reputation model, based on multidimensional trust and reputation, with extensions to provide increased accuracy of assessment, using direct and indirect recommendations, as well as considering the recency and relevance of interactions and opinions. In Chapter 4, the building and maintenance of the agent network model is described, and the various types of information that can be extracted are discussed. Chapter 5 then describes the identification and characterisation of different types of collusion, as well as the collusion detection mechanisms that can be used. Finally, Chapter 6 summarises the previous chapters and concludes with the open issues in the detection of collusion, and discusses the limitations of the mechanism presented.

Chapter 2

Related Work

2.1 Introduction

As introduced in Chapter 1, our aim is to support collusion detection with the use of a rich set of interaction and recommendation information, and knowledge of the agents' social networks. Collusion detection is further enabled by the identification of the types of collusion and their characteristics. This chapter describes important background research work and how it relates to our own research. We explore existing models of trust and reputation in agent-based systems to assess how they handle the trust assessment of other agents and how agents represent their social networks and use them in analysis and agent selection. We then discuss collusion detection in agent-based systems and in existing trust and reputation models.

This chapter is organised as follows. Section 2.2 introduces agent-based systems and the key agent and domain characteristics. It is important to highlight these characteristics as the solutions we propose to help minimise the uncertainty of agent interactions are geared towards these types of agents. We next present, in Section 2.3, the important characteristics that trust models should have to accurately assess the trustworthiness of others. Section 2.4 describes the different approaches to trust modelling in the literature (socio-cognitive, computational and reputational views). In Section 2.5 we review the key trust and reputation models with respect to these trust model characteristics, based on their widespread acceptance and their particular features. The review of these models will be used to contextualise our own trust model. We next introduce social networks, in Section 2.6, as an important element for understanding an agent's environment, especially with limited information. Our approach in Chapter 4 uses this concept to help agents build a representation of their environment to better understand how agents interact. Section 2.7 introduces malicious behaviour in agents and the different forms of deception. This section sets the scene for a particular type of deception that we will be focussing on, namely, collusion. Relevant related work on collusion detection is presented in Section 2.8. Finally, Section 2.9 identifies the weaknesses in the related work that we address as part of the aims of this thesis.

2.2 Multi-agent Systems Characteristics

Multi-agent systems exist in a broad range of domains and can be applied to many different applications, from simple agents used in information retrieval and information filtering to more complex agents used in air-traffic control. In this section, we define an agent and its behaviour by describing its key characteristics. We also define the domain characteristics, which help to shape our understanding of agents, the type of environment they exist in and the nature of their interactions.

2.2.1 Agent Characteristics

The three key agent characteristics are autonomy, heterogeneity and communication. For *autonomy*, we adopt the view of Luck and d'Inverno [76] that autonomous agents derive their autonomy from motivations. In comparison, an agent is defined as an instantiation of an object together with an associated goal or set of goals. Autonomous agents pursue their own agendas for reasoning and behaviour in accordance with their internal motivations. Based on Kunda's work in the field of psychology [60], a motivation is defined as any desire or preference that can lead to the generation and adoption of goals and which affects the outcome of the reasoning or behavioural task intended to satisfy those goals. Thus, an autonomous agent is differentiated from an agent by the goals it possesses and which are generated from its motivations, rather than adopted from other agents.

Four types of agents can be identified [122], categorised according to the agent characteristics of heterogeneity and communication: homogeneous non-communicating, homogeneous communicating, heterogeneous non-communicating, and heterogeneous communicating agents. The level of *heterogeneity* refers to how similar or different agents are with respect to their internal structure, goals, domain knowledge and actions. The other agent aspect of *communication* defines the degree to which the agents communicate with one another. We focus on heterogeneous communicating agent systems, which can be complex and powerful, and consequently have a number of domain-related issues that we are considering next. Based on the characteristics identified by Stone and Veloso [122], we outline those we believe are most representative of the e-commerce domain. Besides the autonomy, heterogeneity and communication characteristics, other relevant ones are as follows.

Deliberative Agents as compared to reactive agents are capable of adapting their behaviour according to their internal state, past history and decision making.

Local Perspective in decentralised systems involves not having a global view of the environment. From their local views, agents have a partial picture of the agent system.

Modelling Other Agents' State as even though agents are able to communicate with one another, due to reasons such as privacy, agents need to model the state, actions and knowledge of other agents. Modelling involves observation of agent behaviour and interactions and predicting future moves.

Benevolence versus Competitiveness We consider agents to be primarily selfish, as they look after their own interests and aim to achieve their individual goals. In some situations, agents may choose to be altruistic and give recommendations to others, in exchange for reciprocal behaviour.

Commitment/Decommitment Agents make commitments to one another when they communicate and decide on how they are to cooperate on a particular task. The commitments provide means for agents to trust that the committing agent will do what it initially agreed to do.

Other characteristics may be expressed by agents, however in this work, we are not focussing on them and we assume that if present, there are supporting mechanisms in place. Examples include negotiation, resource management and communication method. *Negotiation* is a process by which a group of agents communicate with one another to try and come to a mutually acceptable agreement on some matter [75]. The role of negotiation is to ensure that an agreement is reached for the tasks involved to be performed, ideally in such a way that all the parties involved benefit from the negotiation outcome. Another characteristic is *resource management*, where agents may have some interdependent actions due to limited common resources. *Communication method* is also an important characteristic of the agent domain. Since heterogeneous agents are built by different designers, there needs to be a common language and protocol for agents to interact with. We assume that the method and format of communication has been established in the later sections of this work.

2.2.2 Domain Characteristics

Population Size is the number of agents in the system, which varies according to the domain and can range from a few, to several dozens and several hundreds in electronic commerce, and electronic supply chains.

Time Dependency of Actions relates to whether the generation of actions is subjective to time pressures. The type of domains we are considering are real-time and agents' behaviours and actions are influenced by the behaviour of others in the system, as well as environmental factors. For instance, customer agents in an e-commerce system will stop buying from a supplier as it becomes increasingly unreliable.

Dynamism of Agents involves agents entering or leaving the system at will, depending on their goals at various points in the transaction period. Additionally, agents can adapt their behaviour accordingly.

Communication Cost in the domains we are considering is assumed to be almost free [127], as reciprocal behaviour benefits agents when they share information.

Failure Cost in the domains we are considering, such as e-commerce applications and e-supply chains are medium. In contrast, air traffic control is a domain with high cost of failure.

User Involvement pertains to the degree of human involvement in the multi-agent systems. We assume that humans are involved, for instance, to give user feedback on the performance of its representative agents, whose behaviour can consequently be updated.

Environmental Uncertainty can result from the domain itself, from agents not knowing the actions of other agents, and from the agents not knowing the outcome of their own actions [17].

2.3 Trust Model Characteristics

Trust and reputation are popular mechanisms used to help in the selection of the best suited interaction partners by reducing the issues related to uncertainty. Trust is an assessment of the likelihood that an agent will cooperate and fulfil its commitments [32, 81]. The reputation of an agent also contributes to its trust assessment and is derived from third party opinions. Research on trust in the agent domain has brought about many different approaches. Castelfranchi and Falcone [12, 25] view trust as being composed of representations of beliefs, such as competence, disposition, dependence and fulfilment. Marsh [81] looks at basic, general and situational trust, which considers trust with regards to the agent itself, other agents and particular contexts respectively. Griffiths [35] introduces the notion of multidimensional trust (MDT), which allows agents to model the trustworthiness of others according to various criteria. The approach decomposes the beliefs, as viewed by Castelfranchi and Falcone, according to the different dimensions of an interaction. Agents can model trust along any number of dimensions, according to their preferences and motivations. For the purposes of illustrating MDT, Griffiths uses the four dimensions of success, cost, timeliness and quality.

Reputation, a similar notion to trust, is defined as the information received by agents about the behaviour of their partners from third parties, and they use that to decide how they are going to behave themselves [11]. Due to its importance in social and commercial relations, the study and modelling of reputation has attracted a lot of interest from researchers in different fields: sociology, economics, psychology and computer science. We agree with [11] on the definition of reputation and we note that reputation includes recommendations from agents who have directly interacted with the agents we are interested in, as well as indirect recommendations, based on the propagation of reputation among agents.

A trust model should have certain key characteristics to allow an evaluator to accurately represent another agent's behaviour and assess its trustworthiness. It should enable the gathering of a rich set of information for trust assessment and reasoning about an agent's social networks. These characteristics are based on the benefits and limitations of a comprehensive set of trust models that we have studied and discuss them in more detail later in this chapter. Although the trust model characteristics are not new in themselves, and, feature in existing models individually or in combinations, the set of characteristics we describe below has not been considered together in combination in previous work.

2.3.1 Trust Information Sources

The first key characteristic concerns the gathering of trust data from third parties to represent the reputation of an agent. An agent needs to be gathering trust information from a wide range of sources, and third party recommendations should be used together with trust information from direct interactions. Service interactions with provider agents are the most reliable source of direct trust as they most closely relate to the evaluator's requirements. However, direct interactions are not always available for a number of reasons, including interacting with a new service provider, insufficient past interactions to assess a provider accurately, and interactions relating to a new service required by the evaluator. Even in these circumstances, an evaluator wants to have successful interactions and the decision making process to select interaction partners needs to include recommendations from reliable sources. An evaluator can assess the reputation of another agent from a number of recommendations obtained from third parties, either directly or indirectly.

Direct recommendations originate from agents having directly used an agent's services. Therefore, a principal witness will give its opinion of another agent only if has itself interacted with that agent for service provision. A witness's opinion can also be valuable even if it is indirect. SIR [83], TrustNet [110, 111], SPORAS [142], MDT-R [36] and TRAVOS [125] are trust models that feature direct recommendations. Indirect recommendations are provided by secondary witnesses which pass on the recommendations of a principal witness along a chain of recommendation. The shorter the chain, the closer the recommendation is likely to suit the requirements of the evaluator. This is mainly due to each subsequent witness recommending a suitable agent from its known set of agents, which may be of relevance to the requesting agent. From the literature, the following models use indirect recommendations as well as direct recommendations: Ntropi [1], ReGreT [106], Mui *et al.* [89], HISTOS [142], FIRE [46], Walter *et al.* [130], L.I.A.R. [91] and Yu and Singh [141].

2.3.2 Service-level Trust and Reputation

The second key characteristic of a trust model is to be able to assess the trustworthiness of agents at a service level, as well as at a service characteristic level. At a service level, the trust model needs to differentiate between the services provided by the agents. The trustworthiness of agents needs to be modelled per service provided. At a more granular level, an evaluator agent can assess the trustworthiness of another agent in the individual service characteristics important to them.

The assessment at service characteristic level can be performed both for direct trust and reputation. For example, the evaluator may consider timeliness as a fundamental service characteristic that a provider should have. In its assessment of the trustworthiness of that provider, the evaluator will particularly take into consideration its direct trust and reputation in the timeliness dimension. Models by Mezzetti [83] and Griffiths and Luck [39] consider the multi-dimensionality of trust, while ReGreT [106] and MDT-R [36] both also take multiple dimensions of recommendation into account.

2.3.3 Recency

Taking into account the recency of agent interactions is the third key characteristic that a trust model should have. Agents use their history of past interactions in their trust assessment of others. Recent interactions reflect the most up-to-date agent behaviours, and are most likely to indicate future behaviour. Storing long histories of past interactions provides a larger amount of data about an agent to evaluate its trustworthiness. However, in a dynamic environment where agents can change their behaviour, using older interactions may not be a good indication of future behaviour. Agents need to balance the size of the history of interactions that they store about another agent, since a small size may not be sufficient to accurately assess trustworthiness, while a large history would average out the agent behaviour over that long period of time, rather than help predict immediate future behaviour. Therefore, a trust model should be able to filter out the older agent interactions that are less useful in accurately assessing trustworthiness. Trust models in the literature that use the recency characteristic include SIR [83], ReGreT [106], Witkowski *et al.* [134], SPORAS and HISTOS [142, 143], MDT-R [36], and FIRE [46].

2.3.4 Relevance

Relevance is the fourth key factor that a trust model should take into account. It concerns the recommendations received by the evaluating agent, and how useful they are for trust assessment. Relevance is based on the recency of the recommendation interactions, the experience of the witnesses and how trustworthy the evaluator believes the witness is in giving recommendations, as well as the evaluator's confidence in that recommendation trust. These considerations ensure that the most relevant third party recommendations are used for assessing the trustworthiness of other agents. Existing trust models that take into consideration the relevance of recommendations include Ntropi [1], HISTOS [142], FIRE [46], and Yu and Singh [141].

2.4 Approaches to Trust Modelling

Interest in trust and reputation has resulted in many models being developed for the implementation and management of these notions in multi-agent systems. Researchers have adopted approaches from different disciplines to support the development of their models. The notions of trust and reputation have their roots in sociology, economics and biology, and have been applied in areas as diverse as game theory, business ethics and politics. Hence, to model them in agent-based systems, techniques from many of the above-mentioned fields have been used. The main approaches are socio-cognitive, computational and reputational.

2.4.1 Socio-cognitive View

The term *cognitive* is defined in the Cambridge Dictionaries Online as "connected with thinking or conscious mental processes" [97]. Thus, models following the socio-cognitive approach are based on underlying beliefs about a society and its members and trust is a function of the value of these beliefs [22]. Additionally, this approach involves the mental states of an agent in relying on another agent and also the consequences of the actual decision of reliance [25]. It is important to understand the mental ingredients of trust in order to explain and predict the perception and decision about an agent's risk. A cognitive analysis of trust also forms the basis for the notions of reputation, deception, and persuasion in the building of trust [13].

In the literature, only a few trust and reputation models are based on the socio-cognitive view. The main model dealing with the cognitive approach to trust is that of Castelfranchi and Falcone [12, 25], in which they define the different beliefs that an agent must hold to build up trust and expects another agent to have in order to be suitable to be relied on. Other models use the social aspect of MAS to closely represent interactions in real situations. Mezzetti [83] stresses the ideas of trust variation with time and context and the modelling of the properties that cause a reputation value to be low or high.

2.4.2 Numerical View

In this view, trust and reputation are not reflective of the mental state of an agent, but use numbers and mathematical techniques to represent and manipulate the trust value, in the form of probabilities and numerical aggregations and strategies. Within this view, models can be roughly categorised as decision-theoretical or game-theoretical.

2.4.2.1 Decision-theoretical View

Classical decision theory consists of a set of mathematical techniques for making decisions about which action to take when the outcomes of various actions are not known. *Probability theory* is a subset of these techniques, where some aspect of the current state of the environment is captured as a probability. Marsh [81] represents trust numerically between -1 and $+1$. All the three aspects of trust — basic, general and situational trust — lie within this range and he proposes a formula to calculate the situational trust. Mui *et al.* [89] also propose a mathematical model based on probability to show the link between trust, reputation and reciprocation. Models by Witkowski *et al.* [134, 135] and Sen *et al.* [113, 114] also fall into this category.

Other models place trust values and agent behaviour into categories to make them more meaningful in their utilisation. *Fuzzy set theory* is a means of specifying how well an object satisfies a vague description [103]. Zadeh [144] defines a fuzzy set to be a class of objects with a continuum of grades of membership. Many objects in the real world do not have precisely defined criteria for membership and although they are ambiguous, they are important in human thinking, pattern recognition, communication and abstraction. Fuzzy logic has emerged from fuzzy sets and is a method for reasoning with logical expressions describing membership in fuzzy sets. It allows intermediate values to be defined between conventional evaluations, such as 'yes' or 'no', 'late' or 'on time' in terms of the degree of truth. Notions like 'rather warm' or 'slightly late' can be formulated mathematically and processed by computers in an attempt to more accurately represent the way systems behave in the real world.

Wu and Sun [137] classify a seller's behaviour in a bidding environment as Random, Nice, Tit-for-Tat and Nasty, where each strategy outlines the way the seller behaves in an interaction. Abdul-Rahman and Hailes [2] also use classification in the case of trust and for the adjustment of experiences. Trustworthiness is categorised into four types, from Very Untrustworthy to Very Trustworthy, while experiences also exist in four types, from Very Bad to Very Good.

The fuzzy approach is also adopted by Falcone *et al.* [26] for an implementation of the socio-cognitive model of trust they have developed [12, 25]. Fuzzy logic has been chosen for their model because trust is a graded phenomenon that can be difficult to estimate. We consider the implementation of a socio-cognitive model to be both computational and socio-cognitive. The implementation is based on Fuzzy Cognitive Maps (FCM) [58], that allow the value of truthfulness to be computed from the belief sources. An FCM is well suited for representing a dynamic system with cause-effect relations, where nodes represent the causal concepts of belief sources, for instance, and edges represent the causal power of a node over another one. Other work using fuzzy logic includes that of Griffiths *et al.* [38], used in the context of P2P systems to select interaction partners.

2.4.2.2 Game-theoretical View

Within the computational and numerical models, there is a sub-category of models and mechanisms which are based on game theory, thus making use of utility functions and strategies similar to Tit-for-Tat in the Prisoner's Dilemma problem. Game theory has arguably originated from the work by John von Newmann and Oscar Morgenstern [128], where they define a game as any interaction between agents that is governed by a set of rules specifying the possible moves for each participant and a set of outcomes for

	Prisoner A defects	**Prisoner A cooperates**
Prisoner B defects	3, 3	0, 5
Prisoner B cooperates	5, 0	1, 1

Table 2.1: Prisoners' Dilemma Options and Payoffs

each possible combination of moves. Game theory is applicable to almost any social interaction where individuals have some understanding of how the outcome for one is affected not only by its own actions but also by the actions of others [40].

The Prisoner's Dilemma (PD) problem in game theory was described by Albert Tucker while addressing an audience of psychologists, to explain the puzzles devised by Merrill Flood and Melvin Dresher in 1950, as part of the Rand Corporation's investigations into game theory due to its possible applications to global nuclear strategy [59]. As illustrated by Tucker, two prisoners are held for the robbery of a bank. They are placed in separate cells and the prosecutor makes an offer to each of them while explaining what is likely to happen. Table 2.1 summarises the options and payoffs proposed to the prisoners, where the number pair represents the number of years in prison for prisoners A and B respectively.

There is enough evidence to convict each of a minor offence, but there is not enough evidence to convict either of them of a major crime unless one of them defects (confesses), and thus acts as an informer. If both defect, they will each be given three years in prison, due to there being no doubt over their guilt. If only one of them confesses, that prisoner will be freed and used as a witness against the other, who will spend five years in prison. If both cooperate and stay quiet, each will be convicted of the minor offence and spend one year in prison. Given that the assumption is that each prisoner cares only to avoid spending time in prison, the dominant strategy of each will be to defect. Yet, it yields a paradoxical result of making each worse off than they might have been had they each chosen to cooperate and stay quiet and so to spend only one year in prison [40].

Tit-for-Tat is an efficient strategy in game theory for the iterated Prisoner's Dilemma, where a computer tournament is conducted. The strategy is one of cooperating on the first move and then doing whatever the other agent did on the preceding move. It is thus a strategy of cooperation based on reciprocity [4].

TrustNet [110, 111] uses an extension to the Prisoner's Dilemma for the selection of interaction partners. Wu and Sun's [137] approach makes use of the Tit-for-Tat strategy for the behaviour of its seller agent.

2.4.3 Reputational View

In their evaluation of trustworthiness, many models make use of reputation, in the form of recommendations from other agents. Direct interactions with the agents of interest are not always available as sources of information, especially when there have been no previous interactions, or past interactions have occurred a long time ago. Many models take into account reputation as a complement to trust in evaluating trustworthiness. Ntropi [1, 2], ReGreT [106], TrustNet [110, 111], FIRE [46], and TRAVOS [125] all make use of reputation.

The reputation mechanism by Braynov and Sandholm [8, 9] and FIRE [46] use a form of reputation mechanism used by an agent for itself. It consists of revealing their reputation value to other agents with whom they want to interact. Most of the models mentioned use direct recommendations, that is, an agent requests the opinion of others who have interacted with the agent of interest. Indirect recommendations, that is, the opinions of other agents about an agent of interest even if they have not themselves interacted with it, are used by ReGreT [106] and FIRE [46]. The trust-based recommendation system proposed by Walter *et al.* [130] also makes use of direct and indirect recommendations from the agents' neighbours in decision making.

2.5 Review of Trust and Reputation Models

A selection of trust and reputation models from the different approaches mentioned are reviewed in this section. Table A.1 in Appendix A summarises the characteristics of the different trust models with respect to the essential trust model characteristics described previously. We have selected these models in our review based on the key trust model characteristics for the selection criteria (Section 2.3).

2.5.1 Castelfranchi and Falcone

The model proposed by Castelfranchi and Falcone [12, 25] is general and domain-independent, and is based on the mental state of trust. It suggests that an agent can trust another agent if it has an appropriate set of goals and beliefs. Trust is defined as comprising three elements: 'core trust', which is a simple evaluation of the trustee, 'reliance', the decision to rely on the trustee and 'delegation', the actual action of trusting the trustee. To build trust in another agent y, an agent x is required to have certain beliefs corresponding to the three components of trust mentioned previously. The cognitive analysis of trust is fundamental in the distinction between internal and external attribution, which predicts different strategies for building trust. Internal attribution concerns the characteristics of willingness, persistence, engagement and competence, while external attribution involves the conditions of the environment, such as opportunities, resources and interference.

Agent x must have two basic beliefs to trust agent y with core trust: competence belief and disposition belief. *Competence belief* is a positive evaluation of agent y's usefulness in producing the expected result. *Disposition belief* occurs when x believes that y will do the task that is required. In addition, for core trust and reliance to exist, agent x must have the *dependence belief*, necessary for x to rely on y to do a task, out of lack of alternatives or as the more advantageous option in comparison to not relying on y. Supported by the previous beliefs, the *fulfilment belief* arises, which drives agent x to think that the goal will be pursued and achieved.

Delegation, the last element of trust, can occur in two ways: weak and strong delegation. In weak delegation, there is no agreement and no bilateral awareness of the delegation, while in strong delegation, the trustee y is aware of the intention of the truster x to exploit its action. The following three beliefs apply in weak delegation, in addition to the other beliefs previously mentioned. The *willingness belief* models y's mind in its intention to work towards a certain goal, while the *persistence belief* occurs where x believes that y is serious in its intention of doing a task. With the *self-confidence belief*, y knows that it can do the task. Strong delegation requires an additional belief, the *motivation belief*, when x believes that y has some motives to help adopt its goal.

Castelfranchi and Falcone present the concept of *reciprocal trust* [24], which is a mutual understanding and communication between two agents that they will help each other, at different points in time. They claim that the reciprocal trust is different to bilateral trust, which occurs between two agents at the same time, but the agents are not explicitly aware of this. They argue that the opposite is also true: agent x's distrust in agent y induces distrust in y towards x. Another concept touched upon is that of the diffusion of trust. The authors suggest that the trust agent x has in agent y can influence agent z to trust y. The mechanisms suggested for this diffusion are pseudo-transitivity and conformism. Pseudo-transitivity depends on the cognitive conditions that are present and diffusion of trust will most likely occur if the agent whose trust decisions are followed is a figure of authority in the domain. Conformism, on the other hand, is not based on any special expertise and is based on copying another agent's actions or decisions. In their socio-cognitive model, the authors do not make any reference to the possibility of having dishonest agents or collusion in the system. An overall framework of trust using the various concepts introduced has also not been fully defined.

2.5.2 Marsh's Formalism

In the trust model proposed by Marsh [81], trust is viewed as three different aspects, as a result of direct interactions with other agents:

- *Basic trust* is derived from all the past experiences of an agent. It represents the trusting disposition of the agent itself. The basic trust of an agent x is denoted as T_x. This value is in the range $[-1, 1)$, that is, $-1 \leq T_x < +1$, where good experiences increase the disposition of the agent to trust. A value of $+1$ is not allowed as it implies blind trust, where an agent gives trust without hesitation, and this behaviour is not part of the trusting behaviour in the formalism [82].
- *General trust* is the trust an agent has in another agent, irrespective of the situation in which they are found. This is denoted as $T_x(y)$ and the range of general trust values is $[-1, 1)$, that is, $-1 \leq T_x(y) < +1$, where -1 is negative trust or complete distrust and $+1$ is complete trust, while 0 means no trust.
- *Situational trust* is the amount of trust an agent has in another agent in a specific situation. Thus, the notation for 'x trusts y in situation α' is $T_x(y, \alpha)$. The

importance and utility of the situation, together with the general trust value, all determine the situational trust value, which is also in the interval $[-1, 1)$.

The understanding of trust and the trust values obtained allows agents to make more informed decisions about which agents are trustworthy and who to cooperate with. Thus, the competence of the potential interaction partner is assessed based on the situation, its importance, and the risk involved. The basic formula to calculate situational trust is:

$$T_x(y,\alpha)^t = U_x(\alpha)^t \times I_x(\alpha)^t \times \widehat{T_x(y)}^t \tag{2.1}$$

where $U_x(\alpha)^t$ represents the utility x gains from the situation α; $I_x(\alpha)^t$ is the importance of the situation α for agent x and $\widehat{T_x(y)}^t$ is an estimate of the general trust after taking into account all the relevant data with respect to situational trust in past interactions. In order to calculate this estimate, the author proposes three statistical methods: the mean, the maximum and the minimum. These are translated into realism, optimism and pessimism respectively.

These notions of agent dispositions [81, 82] give an indication of how agents will act in a given situation. Along a continuum, agents can range from optimists to pessimists. Optimists are those agents who look for the best in those with whom they interact, they are forgiving and their trust in another does not decrease by much, even after being exploited by another agent. On the other extreme, pessimists see the worst in the agents they interact with and are always in doubt of the resulting situation. Even a small exploitation will result in drastic loss in trust, while continued cooperative behaviour will not greatly increase trust. In between these two extremes lie the realists, acting as a control point in studying the agent behaviours.

The formalism proposed also takes into account the notion of reciprocation, where favours are returned to those who offered them. Reciprocation is used to modify trust; if an agent y helps another agent x, x's trust in y is likely to increase, while if y defects, x's trust in y is likely to decrease.

Marsh's formalism does not model reputation and thus does not consider third party recommendations in the evaluation of an agent's trustworthiness. This may limit the amount of information for trust evaluation in cases where there is insufficient or no direct interactions with the agents of interest. The formalism does not support the trust evaluation of new entrants who have have not interacted before.

Abdul-Rahman and Hailes [2] consider the notions of risk and competence to be abstract and thus difficult to represent as numbers, especially continuous values. They also observe that Marsh's model incorporates a large number of variables, considered make the model large and complex. However, we believe that the use of environmental variables in the model allows the expression of the reasoning behind the trust computation, and helps to preserve the separate elements that make up the trust calculation.

2.5.3 Ntropi

Abdul-Rahman and Hailes [1, 2] propose a trust and reputation model, which is applicable to virtual communities. It is a numerical model with degrees of trust and is based on social characteristics and reputation. Both direct experiences and recommendations are used to form a trust opinion. Many properties of social trust are supported,

- Context dependence is similar to Marsh's use of context in situational trust.
- Positive and negative degrees of belief are supported through a four-value scale.
- Prior experiences are taken into account so that agents can identify similar experiences.
- Reputational information is exchanged among agents through recommendations.

- Non-transitivity of trust is considered and all the evaluations of recommendations take into account their source.
- Subjectivity of trust represents the varying perceptions of different observers with regard to the same agent's trustworthiness.
- Dynamism allows the level of trust in another agent to increase or decrease, according to the experiences and recommendations obtained by the trusting agent.
- Support for *Interpersonal Trust* is the direct and contextual trust an agent has for another agent.

The term 'belief' is used in a different sense to that of Castelfranchi [12]. The model deals with beliefs about trustworthiness, without considering risk, utility, and beliefs about motivation. Here, the belief that an agent is trustworthy in giving a recommendation is taken into account. Four degrees of direct trust are used: 'Very Trustworthy', 'Trustworthy', 'Untrustworthy', 'Very Untrustworthy'. A similar rating is used for experience adjustments: 'Very Good', 'Good', 'Bad', 'Very Bad'. Evaluations of direct trust, recommender trust, semantic distance and the update of experiences contribute to computing the final trust degree. An agent x may perceive its trustworthiness in another agent differently from an agent y's recommendation. Agent x can adjust y's recommendations in the future to close the distance between their respective opinions. The model is thus intended to obtain trust on the information given by witnesses. Direct experiences are used for comparison and adjustment [108].

Abdul-Rahman and Hailes recognise that the model is not recommended for agents without any prior experience nor trusted witnesses. This is due to the high level of uncertainty faced by new entrants who do not know whom to trust or distrust and they can thus become the victims of malevolent agents. With this bootstrapping limitation, the model also does not address the situations when agents lie or collude. It is also not possible to differentiate between truthful and lying agents on the basis that they have different reasoning mechanisms [108]. In addition, the authors concede that some aspects of their models, notably the trust degrees and the weightings used, are ad hoc in nature and do not represent these metrics concretely. Although the model is described as supporting context dependence of trust, this is not clearly described by the authors.

2.5.4 ReGreT

The ReGreT system proposed by Sabater and Sierra [104, 105, 106] is a trust and reputation mechanism based on three dimensions of reputation.

- The *individual dimension* models the direct interactions between two agents. It is considered to be the most reliable dimension of reputation. From an interaction between two agents, the outcome consists of an initial contract of a course of action and the result of the actions taken, and of an initial contract to fix terms and conditions of the transaction and the values of these terms. When calculating an *outcome reputation*, a weighted mean of the outcomes is used while giving more relevance to more recent outcomes.

- The *social dimension* looks at indirect interactions, especially when information from direct interactions is not available. Three types of social reputation are used in the ReGreT system.

  - *Witness reputation* is based on information gathered from other agents who have interacted with the agent of interest. There is the risk of false information being provided in this case.

  - *Neighbourhood reputation* considers links that are created through interactions, as the behaviour of neighbours can give some indication about the possible behaviour of the target agent.

  - *System reputation* makes use of common knowledge about the role played by the target agent in society.

- The *ontological dimension* models a combination of reputational aspects relevant to a particular situation. The properties give more information into the reasons why an agent's reputation is high or low. For example, the calculation of reputation using the ontological dimension can consider two dimensions: the reputation of an agent in delivering late, as well as that in over-pricing.

ReGreT also contains a credibility module to evaluate the truthfulness of information received from third party agents. It also makes use of social network analysis to improve knowledge of the surrounding society, especially in the absence of direct experiences. Social network analysis is described by Scott [112] as having emerged as a set of methods for the analysis of social structures, methods that specifically allow an investigation of the relational aspects of these structures. Moreover, the ReGreT system provides a degree of reliability for the trust, reputation and credibility values, that helps an agent to decide whether it is sensible or not to use them in its decision making process.

This model is based on the group to which an agent belongs. In looking at agent groups, the model implies that information comes from trustful agents, who would not deliberately manipulate information. However, the model does not consider agents that can belong to more than one group at a time, where there may be potential issues of conflict of group association and competition.

The authors also do not specifically mention how to bootstrap the model and how to deal with new agents who have never interacted before. The ReGreT system makes use of up to three dimensions in calculating the reputation of agents. However, the authors do not specify how the different reputation evaluations from the different dimensions can be used together.

2.5.5 Mui et al.

The model proposed by Mui *et al.* [89] has four main characteristics. Firstly, the difference between trust and reputation is explicitly made. Secondly, reputation is a quantity relative to the particular social network of the evaluating agent and its encounter history. Thus, reputation is defined as a "perception that an agent creates through past actions about its intentions and norms". The next characteristic concerns trust, defined as "a subjective expectation an agent has about another's future behaviour based on the history of their encounters", which can be inferred from the reputation of the trustee. Lastly, a probabilistic mechanism is proposed for inference among trust, reputation and the level of reciprocity, to identify a threshold for the number of encounters needed by an agent to achieve a reliable measure of another agent's trustworthiness.

Reciprocity is closely linked to trust and reputation. An increase in reputation expects an increase in trust. An increase in trust in turn expects an increase in reciprocation, and an increase in reciprocation expects an increase in reputation. The model handles the case of when two agents have no previous encounters by introducing an ignorance assumption called the Complete Stranger Prior Assumption.

The model only addresses encounters involving two agents. Other choices made in the model include the assumption that the environment in which agents evolve is static, where no new agents join or leave. Moreover, the binary actions of cooperation or defection restrict the action space of the agents.

2.5.6 SPORAS and HISTOS

Zacharia *et al.* [142, 143] believe that online communities have specific problems which must be addressed by reputation mechanisms for these domains. In online communities, it is relatively easy for agents to change their identity.

SPORAS is a reputation mechanism for loosely connected online communities. In this system the trusting agent bases its opinion of the reputation of its interaction partner on the feedback the latter gives on the trustworthiness of their latest transaction. Only the most recent ratings are stored for agents who have repeated interactions. A new user will have the minimum reputation which is gradually built up as it interacts with others. However unreliable an agent may be, its reputation value will nevertheless be higher than that of a new agent. With this strategy, a user is always worse off when it switches identities.

While SPORAS provides a global reputation value to each agent in the online community, HISTOS is a more sophisticated approach, which takes into consideration information about an agent's peers when available. Agents in this system rely more on recommendations given by agents they trust than those given by agents they have never interacted with previously. HISTOS builds a social network from the pairwise ratings it has previously obtained. This is represented as a directed graph with the nodes representing the agents and the weighted edges representing the most recent reputation rating given by one agent to another. The transitive trust relationships are thus applied where there are directed paths between two agents.

2.5.7 MDT-R

MDT-R [36] is a mechanism for multidimensional trust and recommendations. Agents model the trustworthiness of others according to various criteria, such as cost, timeliness or success, depending on which criteria the agent considers important. Agents use their own direct experience of interacting with others, as well as recommendations. Distinguishing trust and recommendations for individual characteristics is valuable in identifying the service characteristics in which the providing agents perform well, or less well. Trust information in multiple dimensions helps to maintain the original interaction data. Trust values are represented numerically in this approach due to the benefits of accuracy and the ease of comparisons and update of values. However, MDT-R stratifies trust into levels (similar to Ntropi) for ease of comparison. The sharing of information among agents often suffers from subjectivity, due to differences in interpretation. MDT-R deals with this by sharing summaries of relevant past interactions, instead of explicit values for trust.

2.5.8 FIRE

Huynh *et al.* [46] propose FIRE, a trust and reputation model that integrates many different information sources to produce a comprehensive assessment of an agent's likely performance. FIRE is designed for open multi-agent systems, where agents can be owned by several stakeholders and can join and leave the system at any time. The other characteristics of agents in open MAS include the assumption that they are potentially unreliable and self-interested. The agents also know a limited amount about their environment and there is no central authority that controls all the agents. Due to the incomplete knowledge about their environment and other agents, trust can facilitate the interactions between agents.

In order to meet the requirements of open MAS, the authors believe that a trust model should possess the following properties.

- The model should take into account a variety of sources of trust information so that the trust measure can be more precise and cater for cases when not all sources are available.
- Every agent should be able to evaluate trust for itself.
- The model should be robust against possible lying agents.

FIRE makes use of four different types of trust and reputation sources: interaction trust, role-based trust, witness reputation and certified reputation. These various sources are important in the model as they ensure a combination of available information sources and that a trust measure is obtained whenever it is needed for interaction.

Interaction trust models the trust that occurs as a result of direct interactions between two agents. The individual dimension of the ReGreT system [106] is adopted as it meets all the requirements for handling direct experiences. *Role-based trust* models the role-based relationships between two agents and rules are used to assign values to this particular type of trust. One benefit of using rules is that users can add new rules to customise their applications. The *witness reputation* of an agent x is built on the observations of its behaviour by other agents, acting as witnesses. For an agent y to evaluate the witness reputation of agent x, y must find witnesses that have interacted with x. Agents keep a list of acquaintances and query a number of them when a query needs to be made. If the acquaintances cannot answer, they will send referrals pointing to other agents they think will know the answer. The last kind of information source is *certified reputation*, where ratings are presented by the rated agent about itself which have been obtained from its partners in previous interactions. An agent is allowed to choose which ratings to show and because rational agents will always present their best ratings, it should be assumed that certified reputation information is an over-estimate of the agent's actual performance. This type of information source is valuable due to its high availability, and can hence be used even when the other three sources cannot provide a trust measure.

The four trust and reputation measures are combined to generate a single composite value, representing an overall picture of an agent's likely performance. Using the weighted mean method, a composite trust value and its reliability are calculated. Through empirical evaluation, the authors show how FIRE helps agents to select more reliable partners for interaction. In a simulated open MAS, FIRE helps agents to obtain better utility and to quickly adapt to a changing environment while maintaining a high performance.

FIRE, however assumes that agents report their trust and reputation information truthfully, thus the model does not yet deal with lying agents. This model is deemed to be ad hoc due to the hand-crafted formulae used to calculate trust [125]. Even though the model differentiates between the concepts of trust and reputation, it is unclear how the different trust and reputation measures are updated in the light of new information obtained.

2.5.9 TRAVOS

The Trust and Reputation model for Agent-based Virtual OrganisationS (TRAVOS) [125] models an agent's trust in an interaction partner. The model uses probability theory to calculate trust from information about the past interactions between agents. In addition, the model makes use of reputation information from third parties when the lack of personal experience makes direct interaction information unavailable. Dealing with third party information has the risk of inaccuracy and the model handles this aspect.

The model aims to meet the following three requirements.

- A trust metric should be provided to represent the level of trust in an agent, both in the presence or absence of personal experience. It will also be used to compare the trustworthiness of different agents.
- An agent's confidence in its level of trust in another agent should be reflected in the model.
- The model should be able to cope with inaccurate information from other agents, by discounting those opinions in the calculation of reputation.

For any two interacting agents, a history of interactions is recorded as the number of successful and unsuccessful interactions. From this, the variable $B_{a_{tr},a_{te}}$ is obtained, which is the probability that the trustee a_{te} will fulfil its obligations during an interaction with the truster a_{tr}. Thus, using the history of past interactions, the expected value of $B_{a_{tr},a_{te}}$ at a particular time t is calculated using a probability distribution, and is defined as $\tau_{a_{tr},a_{te}}$. If the truster has a low confidence level in its assessment of the trustworthiness of a partner, it can seek the opinions of third party agents. Reputation is modelled as a combination of the true and reported opinions of a source a_{op} about a trustee a_{te}. The authors claim that two conditions must hold for the trust and confidence levels from third party observations to be the same as it would be if all observations had been observed by the truster itself. The first condition states that the behaviour of the trustee must be independent of the identity of the truster with which it is interacting. Secondly, the reputation provider must report its observations accurately and truthfully. However, in a range of situations, these conditions cannot be expected to hold.

When either of the two conditions is broken, inaccurate reputation reports are obtained, due to malicious agents or inconsistent behaviour towards different agents. In the literature, endogenous and exogenous techniques [51] have been used to assess the reliability of reports. Endogenous methods attempt to identify unreliable reputation information by considering the statistical properties of the reported opinions alone. Exogenous methods rely on other information to make a judgement, for example using the reputation of the source or its relationship with the trustee. TRAVOS proposes an exogenous method to filter out inaccurate reputation, where a witness is judged on the perceived accuracy of its past opinions. In the first step, the probability that a witness will provide an accurate opinion is calculated, given its past opinions and later observed interactions with the trustees for which opinions were given. Secondly, based on this value, the distance is reduced between a witness' opinion and the prior belief that all the possible values for an agent's behaviour are equally probable. In having all the opinions adjusted in this way, the witness' influence on a truster's assessment of a trustee is reduced.

Empirical experiments demonstrate that TRAVOS allows reputation to significantly improve performance despite the negative effects of inaccurate opinions. However, the model assumes that the behaviour of agents does not change over time, but in many cases this is not a suitable assumption. The representation of the interaction ratings is considered to be oversimplified and too limited for this model to be suitable for a wide variety of applications in open MAS [47].

The model makes use of a truster a_{tr}'s estimate that a trustee a_{te} will fulfil its obligations and the confidence a_{tr} has in this value. The authors calculate the confidence metric as the proportion of the probability distribution for the trust metric that lies within the bounds of an error value estimate ϵ, that is, between $(\tau_{a_{tr},a_{te}} - \epsilon)$ and $(\tau_{a_{tr},a_{te}} + \epsilon)$. It is, however, unclear how this error ϵ is determined and what is considered to be an acceptable error margin.

Third party recommendations are obtained from those agents who have directly interacted with the agent of interest. TRAVOS does not consider indirect recommendations where an agent obtains the opinion of another agent, who has obtained it from some other agent. This source of information can be useful when not enough information is obtained from agents who have directly interacted with the target agents.

2.5.10 Walter et al.

Walter *et al.* [130] propose a recommendation system on a social network, based on trust. In their model, agents use their social network to gather information and they use trust relationships to filter the information they require. Agents get recommendations from neighbours, which are agents directly or indirectly connected in the network. Neighbours pass on queries to their own neighbours when they cannot provide a recommendation themselves. Agents use trust in their decision making, to choose the most appropriate recommendation from a set of recommendations obtained from a query.

Agents are connected in a social network and each agent is linked to a set of neighbours. For example, a group of people recommending books form such a network. Objects are the subject of recommendations and in the example, books are objects. Objects can belong to one or more categories, for instance, books can be in the categories 'Computer Science' or 'History'. Agents are also associated with a preference profile, which maps a rating to an object. Trust relationships exist among agents when they keep trust values of their neighbours. The model considers that trust is transitive and trust propagates along a path in the network, with the appropriate discounting. The trust value along a path is thus the product of the trust values of the links on that path. When an agent makes a query, it receives a set of responses back from its neighbours. The agent must then choose the best recommendation for its purposes from the set. The trust values provide a ranking of the recommendations and the selection mechanism chosen in the model is random selection among all the recommendations. The higher the trust of recommendations along a path, the higher its probability of being chosen. Once the recommendation is chosen and an interaction occurs as a result of this recommendation, the agent feeds the experience back into the trust relationship with the recommender.

The authors claim that the system self-organises in a state with performance near to optimum when the model is used. Despite the fact that agents only consider their own utility function and without explicit coordination, long paths of high trust develop in the network, allowing agents to rely on recommendations from agents with similar preferences, even when these are far away in the network.

2.5.11 L.I.A.R.

Muller and Vercouter [90, 91] present L.I.A.R., a model of social control for agents consists of several components. The model ensures the reliability of agent communications, as well as provides a framework allowing agents to detect lies and update their decentralised reputation values accordingly. A lie is defined as a wrong behaviour, whereby a query is incorrectly answered, for the benefit of one party and at the expense of another. One component models the agent interactions, while a model of norms defines which interactions are acceptable. L.I.A.R. uses a model of reputation to assess the behaviour of other agents and then enables agents to reason about trusting other agents or not, and to apply sanctions. A sanction is normally associated to the violation of such a norm in order to penalise the agents that do not respect it. However, due to the decentralised nature of the systems under consideration, there is no central institution that applies the sanctions to the violators of the rules-norms (which are norms that must be respected by every agent in the system). The sanction can nevertheless be executed by the other agents in the system through a local increase or decrease of the reputation value of the violator.

The reputation model aims to estimate the compliance of other agents' behaviour with respect to the social norms. The model identifies seven roles that agents can play and different types of reputations based on these roles. These seven roles are *target:* the agent being judged; *beneficiary:* an agent that reasons and decides based on the reputation levels; *observer:* an agent that observes a message and interprets it as a social commitment; *evaluator:* an agent that generates social policies from social commitments and norms; *punisher:* an agent that computes reputational levels from a set of social policies; *propagator:* an agent that sends recommendations — messages about observes messages, social policies or reputational levels; and *participant:* an agent that interacts with the target. The L.I.A.R. model evaluates the reputation of agents with the detection of fraud as a first step.

The L.I.A.R. model uses direct interactions, as well as direct and indirect recommendations as trust information sources. Direct interactions include messages from the target agent to the evaluator, and observations by the evaluator. Recommendations are based on observed messages, social policies and reputation levels shared among agents. Five kinds of reputations are defined, whose reputation values are ordered from most reliable to least reliable:

Direct Interaction based Reputation is based on direct experiences between the beneficiary and the target.

Indirect Interaction based Reputation is built from messages observed by the beneficiary.

Evaluation Recommendation based Reputation is built from social policies propagated to the beneficiary by a propagator.

Reputation Recommendation based Reputation is built from reputational levels propagated to the beneficiary by a propagator.

General Disposition to Trust represents the inclination of the beneficiary to trust another agent if it does not have any information about its honesty.

The separation of reputation values is maintained as they represent different points of view of agents about others and they are not all used in every situation. This differentiates from the models [46, 70, 104] where different types of trust are eventually merged into a single value for decision making. The reasoning process includes a cascading process that works with thresholding and the ordering of reputations as described previously.

2.5.12 Yu and Singh's Referral System

The proposed approach to evaluate the trustworthiness of agents is based on referral networks [139, 140]. The model represents trust as both the cognitive and the mathematical views. The cognitive view considers trust as a function of the underlying beliefs. Meanwhile, the mathematical view uses a metric to model the subjective probability with which an agent will perform a particular action, without taking the beliefs into account. Agents use their prior interactions as well as recommendations from witnesses. The reputation management model, based on the Dempster-Shafer theory [115], assigns no reputation to an agent about which no information is available. Thus, there is no causal relationship between a hypothesis and its negation, for instance, lack of belief does not imply disbelief. Agents model their acquaintances and neighbours (subset of acquaintances, with whom the agents are in contact) as part of their referral network, and this includes the agents' abilities to be trustworthy (expertise) and to recommend other trustworthy agents (sociability).

Agents propagate their direct interaction experiences with other agents, not their combined opinion from direct interactions and recommendations. Agents propagate opinions along a referral chain until a rating is obtained or the depth limit is reached. Shorter referral chains are more likely to be successful and accurate [55].

The issue of deceptive agents is presented in [141], where the approach allows agents to efficiently detect deceptive agents by using a weighted majority based technique to model the belief function (the sum of the beliefs committed to the possibilities in the subset of propositions under consideration) and their aggregation. The weighted majority algorithm (WMA) is used to improve the predictions based on a set of witnesses. Weights are first assigned to the witnesses and a prediction is made based on a weighted sum of the ratings provided. Then, the weights are tuned after an unsuccessful prediction so that the relative weight of the accurate witnesses increases while that of the inaccurate witnesses decreases. This algorithm is adapted to predict the trustworthiness of an agent based on the opinions of witnesses. A variant of WMA is WMA Continuous (WMC) that deals with predictions that are not scalar, as is the case with belief functions, which are mapped to probabilities. Agents can make accurate predictions despite the presence of lying witnesses due to the weights of different witnesses being adjusted, such that the opinions of lying witnesses have less effect on the aggregation of recommendations.

2.5.13 Other Models

We now briefly overview some of the other trust models in the literature, which however do not match our selection criteria closely enough to be discussed in detail.

Socially-Inspired Reputation (SIR) Mezzetti [83] proposes a reputation model, where an agent, having authority in a particular context or situation, can be trusted in providing reliable recommendations (direct and indirect) about other agents within that context. SIR uses attributes to express the relevant properties in that context. Only the more recent information are used by incorporating a decay rate for the trust degrees, with the rate varying depending on the level of risk associated with the context. The social reputation model updates trust and reputation values dynamically as a result of the interaction outcomes.

TrustNet Schillo *et al.* [111] present a trust evaluation mechanism that allows agents to cope in environments where both selfish and cooperative agents evolve. The approach makes use of information from direct interactions, as well as from third party observations. In relying on recommendations, there is the possibility of noise in the information obtained, due to lying and biased agents. TrustNet deals with unreliable witnesses by using an estimation of how often witnesses have lied.

Braynov and Sandholm This approach [8] targets the non-enforceable contracts between a buyer agent and a seller agent. It shows that the seller should precisely estimate the trustworthiness of the buyer in order to maximise its gains. An underestimation leads to insufficient allocation of resources and thus causes losses to both agents. To solve this problem, the buyer should reveal its actual level of trustworthiness to the seller and untrusted buyers can make advance payments to the seller.

Wu and Sun A computational approach [137] is proposed to explore the emergence of trust between agents in a multi-agent bidding setting. A seller can use four strategies (Random, Nice, Tit-for-Tat and Nasty) to reflect the adopted behaviour. Interactions in a friendly climate — where sellers use the Nice strategy — do not necessarily ensure cooperation. Cooperation is considered between self-interested parties, who are most concerned with their own utility. However, there is potential loss of utility early in the interaction period, for example, when agents need to give away some resources [99].

Witkowski et al. This approach [134, 135] focuses on the agents' direct experiences for obtaining trust information. A trading scenario is used for evaluation, and the trust calculation is simplified through measurable quantities of bandwidth allocation and use for consumer and supplier agents. Agents tend to form strong partnerships rapidly and these become more important as the demand and supply for the commodity become mismatched. As demand exceeds supply, only the more successful partnerships are sustained, whereas when the demand increases to exceed supply, supplier agents discard less trusted customers first.

Sen and Dutta A probabilistic reciprocal mechanism [113, 114] is proposed to generate cooperative behaviour among self-interested agents. Reciprocity involves a predictive mechanism, such that an agent who helps another agent will expect to get benefit from the latter in the future. The reciprocative agent balances costs and savings for cooperation decision. It can adapt to the environment and improve its individual performance in the long run, as compared to a selfish agent.

Griffiths and Luck The approach [39] considers an extension to a Belief-Desire-Intention (BDI) agent architecture, particularly to enhance plan selection. BDI agents are based around their beliefs, about themselves and others, their desires of what they want to achieve, and their intentions, made up of actions, and subgoals are represented as adopted plans. Plan selection involves choosing the plan that is most likely to succeed in terms of the least cost in time, resources and risk. As risk increases when other agents are involved in an agent's plans, the latter needs to consider the following factors to compare plans: the likely cost of a plan, the likelihood of finding the agents to execute the plan, the likelihood of their cooperation, and once committed that they will actually fulfil their commitments.

2.5.14 Synthesis on Views of Trust and Reputation Models

The trust and reputation models discussed all attempt to provide solutions to accurately represent these notions in cooperation among agents. Nevertheless, they are limited and deal with only some of the important considerations necessary when looking at open and distributed multi-agent systems.

2.5.14.1 Socio-cognitive Models

The models based on the cognitive and social nature of trust among agents detail the important aspects to consider, such as competence, willingness and motivation of the agent in trust-building. However, they do not explicitly define how these aspects are to be represented and used. Moreover, neither model reviewed [12, 83] models dishonest agents and ways to deal with lying or collusion.

2.5.14.2 Numerical Models

Numerical models allow trust and reputation to be explicitly represented as values, which can be used for further analysis and decision making. However, one concern is that some models tend to over-simplify those notions, and the important considerations in obtaining those values tend to be blurred and are no longer readily available once the trust value has been calculated. The values used in certain calculations also tend to be ad hoc in nature and there is not full justification for the choice of calculation method. Furthermore, with information being increasingly shared among agents, the trust values and their meanings can prove to be an obstacle in the efficient propagation of trust and reputation for other agents to use. While a particular number and formula can be perfectly satisfactory for an agent's sole use, their value on sharing can be very much reduced.

2.5.14.3 Reputational Models

Most reputational models have used reputation as the complement of trust. In doing so, they have reinforced the information from direct interaction with information form third-party agents. Models that do not make use of indirect recommendations lack the ability to obtain information about the trustworthiness of another agent when direct recommendations and direct interactions are rare. Moreover, many reputational models do not handle lying and dishonest agents or differentiate between mistakes in opinion and malevolent behaviour.

2.6 Social Networks

In Section 2.1, we introduced the need to investigate how agents represent their social network, with the aim to further analyse agent interactions. The search for relevant information involves finding the right sources, for example, the agents who have the desired information or expertise. The social network is important in discovering those relevant information sources. An agent is only aware of a portion of the social network to which it belongs [56]. Additionally, due to issues such as privacy, agents will not list their social relationships on a central repository. Agents can, however, gather this information through distributed searches through referrals. Referrals are important for information flow. Studies of the phenomenon of word-of-mouth found referrals to be very effective in communicating product information among consumers and influencing their purchasing choices [10]. Further evidence that referrals are effective in searching large social networks has been demonstrated, for instance, by Milgram [84, 126], leading to the concept of *Six Degrees of Separation*. Milgram examined the social connectivity among people and his study involved asking participants to send a packet to a given individual with some information about the person. The participants had to send the packet through individuals they knew by their first name, hence the participants had to choose the most likely intermediary in the chain. Milgram concluded that the individuals within the study were separated by an average of six intermediaries, or six degrees of separation. Milgram's work was one of the first in academic research on the small world theory, which suggests that any pair of entities in a seemingly vast random network can actually connect in a predictable way through relatively short paths of mutual acquaintances [34]. The classic finding of six degrees of separation has been more recently confirmed by Leskovec and Horvitz [64], who studied anonymised data capturing a month of high-level communications activities within the entire Microsoft Messenger instant-messaging system. They found that the shortest path length is 6.6.

2.6.1 Link Prediction

The high dynamism of social networks suggests the addition of new interactions and deletion of old ones in the underlying social structure representation, thus making the understanding of the mechanisms of evolution of social networks important. Liben-

Nowell and Kleinberg [68] studied *link prediction* as a basic computational problem underlying social network evolution. They described the problem as involving the accurate prediction of the edges that will be added to the network, during the interval from a time t to a given future time $t\prime$. They researched the extent to which the evolution of a social network can be modelled using features intrinsic to the network itself. The link prediction problem is also relevant to the company environment, where the company can benefit from the interactions occurring within the informal social network among its members. These interactions serve to supplement the official hierarchy imposed by the organisation [56, 98]. We view the link prediction problem to have parallels with the discovery of information about an agent's environment through the agents' local views, which can be overlapped to some extent to give a wider perspective of the other agents in the system, their transactions and social links.

2.6.2 Trust Models using Social Networks

Several of the existing trust models use the notion of an agent neighbourhood, the more relevant ones being ReGreT [104, 106] and FIRE [47]. The neighbourhood of an agent refers to the links that it creates through interactions with other agents, rather than their physical location. From neighbourhood reputation, the evaluator makes a judgement about a target agent from the behaviour of the other agents to which it is associated. In ReGreT, the neighbourhood of an agent is assumed to be a group of agents with some common knowledge and neighbourhood reputation represents the behaviour of the whole group. However, the way in which the agent builds up its neighbourhood is not specified in ReGreT. The notion of neighbourhood is used by FIRE in its witness reputation module for searching for relevant witnesses. This is based on Yu and Singh's referral system [141], described in more detail below.

2.6.2.1 Referral Systems for Multi-agents

Yu and Singh [141] present a referral system for agents that enables them to share referrals for the location of relevant information. Finding relevant information involves finding the right information sources, for instance, the people or agents to ask, who have the desired information or expertise. The social network is important in discovering those relevant sources. Due to issues such as privacy, agents will not list their social relationships on a central repository. Agents can, however, gather this information through distributed searches through referrals. Focussing on the dynamics of social networks and their effects in information flow, Yu and Singh seek to efficiently search social networks with the help of agents and their local knowledge. Each agent maintains a personal social network and queries other agents for information and these agents may respond with a reply or with referrals to others. An agent's personal social network models its acquaintances, the closest ones are known as neighbours. As it is only allowed a small number of neighbours, the agent periodically reviews its acquaintances and may promote or demote some of them. This process is based on the answers to queries and the expertise of the referring agent.

2.7 Deception

As we introduced in Section 2.1, the aim of this thesis is to support collusion detection. In this section, we investigate deception in agent-based systems, as a more general form of malicious behaviour. We look at the background work on collusion in more detail in Section 2.8.

Whaley [133] defines deception as any attempt intended to distort another person's or group's perception of reality, whether by words or actions. This differentiates it from misinformation and incomplete information. This definition also discards self-deception, as the target of deception is seen as not oneself but always another. This is depicted in a typology of perception, as shown in Figure 2.1. In particular, the difference between deception and misrepresentation is similar to our view of agents behaving maliciously compared to those who give an inaccurate opinion due to different experience or unintentionally passing on incorrect information.

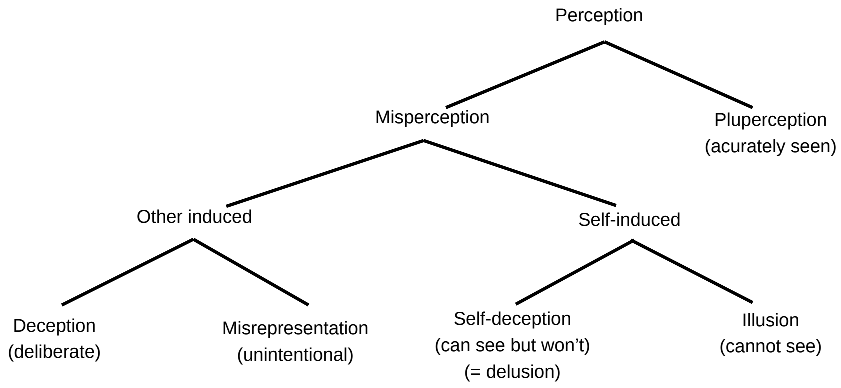


Figure 2.1: Typology of Perception [133].

Agents faced with deception are susceptible to a number of vulnerabilities. In the following sections, we discuss some of these vulnerabilities in various domains of agent-based systems. Collusion is another type of vulnerability, which we focus on later in the chapter.

2.7.1 Strategic Oscillation in Behaviour

One characteristic of P2P networks is their dynamic nature and the high rate of peer turnover when peers join and leave the application periodically. Reputation values generated are therefore from a small number of interactions, and the selection of peers based on short-term reputations is not desirable. Therefore, reliable reputation values need to be obtained, especially for unknown peers and newcomers. Malicious peers can exploit the short-term reputations to build a high reputation at the start and then periodically fail to deliver the expected level of service. A variation of this behaviour is that of the first time offender problem, or identity problem [102], where an agent only behaves maliciously once it has built up a strong trust and reputation among other agents.

Swamynathan *et al.* [123] propose to augment a reputation system with proactive reputations to make it less vulnerable to such strategic oscillation in behaviour. Moreover, proactive requests are associated with first-hand observations, which are more trustworthy and thus less vulnerable to false ratings and collusion. The approach is intended to be integrated with a traditional global reputation system. The idea of proactive reputation is to explicitly measure the reliability and trustworthiness of a target, by other means than the passive evaluation that occurs after the target executes some task for the evaluator. It aims to blend proactive requests with regular traffic to analyse the target's normal response. For this approach to be feasible, transactions must have a low cost so that they do not create significant overhead, and they must also carry a uniform value, to enable proactive transactions to have a similar priority to a typical transaction. Additionally, the transactions need to be verifiable to test whether the transaction was performed properly. Proactive reputation systems face the challenge of ensuring that the proactive requests are indistinguishable from normal application requests and that the originator of the requests remains anonymous. The authors investigated the use of several similarity metrics that would allow a target to detect proactive requests statistically, by observing and comparing the rate of messages against that of normal traffic. These metrics include conditional entropy (measures the likelihood of predicting the $(N+1)^{th}$ value given the last N values), relative entropy (a measure of dissimilarity between two probability distributions), histogram similarity (includes metrics such as weighted Euclidean distance and square distance). A distribution's entropy is a measure of its randomness. Swamynathan *et al.* found that using histogram similarity performed better than conditional entropy and relative entropy when two traffic streams as observed by a target are compared: a normal request stream without proactive bursts and the current request stream, which is possibly injected with proactive requests. Proactive reputation also includes an anonymising scheme to provide sufficient cover for anonymous proactive requests. Changing the rate of anonymous transactions periodically and randomising the number and rate of anonymous transactions appears to perform better than a constant rate of anonymous transactions. The issue of determining how much anonymity is required to evade detection remains. The way to integrate proactive reputations with global reputations while avoiding collusion vulnerability is another issue to be tackled.

2.7.2 False Ratings and Lies

We consider agents which give false ratings deliberately about other agents to behave maliciously. For instance, if an agent is asked to provide a recommendation about another, giving a higher or lower opinion than in reality would fall into this category of deceptive behaviour. Excusable failures [120] are thus not the type of issue considered here, as these result from wrongly diminishing the trust in an agent, from bad performance, rather than from deliberate malicious behaviour.

Agents can have numerous reasons to give false ratings. For example, if a provider is good, an agent might want to deter competing agents from being able to get access to this provider by deliberately decreasing the trustworthiness of the provider when sharing opinions. Another example is when an agent increases the profile of another agent, even if that agent does not perform as well as it is being claimed.

Yu and Singh [140] look at the problem of deception in the propagation of recommendations. In their model of reputation management, agents can obtain referrals to potential witnesses, from whom direct recommendations are retrieved. Direct recommendations are used to avoid the problem of double counting of evidence, which can lead to rumours in a decentralised system, where agents hold opinions about others just from having heard them from others. Moreover, the authors propose a mechanism of aggregation of recommendations to also avoid the effect of rumours. They consider three kinds of deception: complementary, exaggerated positive and exaggerated negative. These are compared to a normal rating, where the rating given by a witness is the same as the true rating of the target agent. The weighted majority algorithm (WMA) is used to assign values of importance to the witnesses and make a prediction based on the weighted sum of the ratings provided by them. After an unsuccessful prediction, the weights are tuned to increase the relative weight of the successful witnesses, while decreasing that of unsuccessful ones. The reputation management framework used by Yu and Singh is based on the Dempster-Shafer theory of evidence, which takes into account belief functions. To cater for these belief functions, a variation of the weighted majority algorithm, called WMA Continuous (WMC) is used, which also allows predictions to be within an interval, rather than binary values. The evaluating agent builds a graph of referral chains produced from the evaluator's queries, and deception is detected by applying WMC to the witnesses found.

Lies have also been studied in agent-based systems by Muller and Vercouter [90, 91]. They define a lie as a wrong behaviour, whereby a query is incorrectly answered, for the benefit of one party and at the expense of another. L.I.A.R., introduced in Section 2.5.11, is a social control approach to agent interactions, composed of a social commitment model that enables agents to represent and reason about interactions and models of social norms and social policies that allow agents to define and evaluate the acceptability of the interactions. The L.I.A.R. model includes a reputation model that enables agents to apply sanctions to their peers. The reputation model ensures the reliability of agent communications, and provides a framework that allows agents to detect lies and update their decentralised reputation values accordingly. Fraud detection consists of monitoring contract execution, whether the contracts are implicit or not. Since there is no contract established for communications between agents, a norm is introduced to define the accepted communicative behaviours, so that contradictory situations which might have been caused by lies, can be detected. The communicative behaviours of agents can be defined according to the states of their commitment stores, which are sets of commitments.

In the lie detection process of the L.I.A.R. model, an agent x observes some messages that violate the norm. In situations of contradiction where an agent y is suspected of lying, the agent that observed the contradiction, x, executes the following steps to confirm that a lie occurred:

1. x sends a message to y containing copies of the contradictory messages to state that x suspects y of lying;

2. If it can do so, y sends a message to x that cancels the contradiction;

3. If the contradiction still holds, x considers y as a liar and can update its trust model of y with this information.

Argumentation processes also apply to reach a consensus about a given fact, arising from diverging opinions. In the model, an agent has a trust model about another agent through reputation values. As described in Section 2.5.11, there are different types of reputation, depending on the roles fulfilled by agents and seven roles have been identified, relevant to the lie detection process.

The L.I.A.R. model evaluates the reputation of agents with the detection of fraud as a first step. A separation of reputation values is maintained as they represent different points of view of agents about others and they are not all used in every situation. This differentiates it from other models [46, 70, 104] where different types of trust are eventually merged into a single value for decision making. The reasoning process includes a cascading process that works with thresholding and the ordering of the five kinds of reputations (described in Section 2.5.11).

2.7.3 Shilling

In e-commerce systems, sellers aim to sell their products on the marketplace and one way to accomplish this is by producing and selling quality goods that buyers will regard highly. However, unscrupulous sellers may opt for the more deceitful route and try to influence recommender systems in such a way that their products are recommended more often to buyers, even though they might not be of high quality. One way to influence recommender systems is to arrange for a group of users to enter the system and vouch for the intended items. The users thus become *shills* whose false opinions are intended to mislead others [62]. Other related work focussed on shilling attacks in online recommender systems includes that of Chirita *et al.* [14], who propose the use of statistical metrics to detect patterns of shilling attacks.

2.7.4 Sybil Attacks

P2P systems often rely on redundancy to reduce their dependence on potentially hostile peers. A Sybil attack is the forging of multiple identities by a small number of peers in a P2P network to compromise a disproportionate share of the system, by exploiting the use of redundancy [19]. In reputation systems, a user may strategically create sybils or identities, for the purpose of boosting its own reputation. One example is the link spamming attack to PageRank [94], where a single user attempts to boost his reputation by creating a large number of duplicate identities, who all recommend him [5, 20].

P2P systems are susceptible to Sybil attacks if distinct identities cannot be established by an explicit certification authority, such as VeriSign[1] or implicitly, such as the CFS cooperative storage system [16], which identifies each node partly by a hash of its IP address. Douceur [19] claims that in the absence of an identification authority, local entities in large-scale distributed systems cannot practically validate the identities of all the other entities to ensure that they are distinct. This is due to the fact that a local entity can make this discrimination, using the assumption that an attacker has limited resources. Subsequently, entities can issue resource-demanding challenges to validate entities and entities can collectively pool the identities they have validated separately. However, this approach requires three conditions to hold, which are not justifiable or practically realisable in large-scale distributed systems: (i) all entities operate under nearly identical resource constraints, (ii) all presented identities are validated simultaneously by all entities, coordinated across the system, and (iii) when the accepting entities are not directly validated, the required number of vouching entities exceed the number of system-wide failures.

Sybil attacks may also be addressed by developing attack-resistant algorithms, such as in work by Dellarocas [18] and Kamvar *et al.* [54], or by increasing the cost of acquiring identities [30]. These techniques could also be used to defend against shilling attacks [61].

[1] `http://www.verisign.com/`

2.8 Collusion and Its Detection

Collusion is defined as a collaborative activity of a subset of users that grants its members benefits they would not be able to gain as individuals [66]. It is a complex problem that has gained interest in many domains. We look at some of the work that has been done towards solving collusion issues, including collusion detection, from both an economics and an agent-based perspective.

2.8.1 Economics Perspective

From an economics perspective, collusion occurs among firms and the main characteristics affecting collusive activity have been studied by Asch and Seneca [3]. Motta [88] has also considered collusion in industrial economics, and specifies that there are two elements which must exist for collusion to arise: (1) the participants must be able to detect in a timely way that a deviation has occurred (e.g. a firm setting a lower price or producing a higher output than the collusive levels agreed upon), and (2) there must be a credible punishment, which might take the form of rivals producing much higher quantities (or selling at much lower prices) in the periods after the deviation, thus decreasing the profit of the deviator. Collusion therefore, can only be sustained if the firms meet repeatedly in the marketplace. Otherwise, a punishment cannot take place. Collusion will not normally arise in one-shot games, therefore collusion should be modelled though dynamic (repeated) games. The identification of the main factors that facilitate collusion is important for two main practical reasons. Firstly, it allows anti-trust authorities to intervene and prevent explicit or tacit collusion whenever possible. Secondly, in cases such as merger analysis, it is crucial to evaluate whether a particular industry is prone to a collusive outcome or not. Thus, studying the industry and assessing whether there are factors that are likely to lead to collusion are important requirements.

In the following sections, we describe the factors that affect collusion, from both the work of Asch and Seneca [3], and Motta [88].

2.8.1.1 Structural Factors

With few sellers, it is easier to achieve agreement on price, and also easier to police the agreement, as cheating by one participant is revealed by a loss of sales by the others (concentration factor [88]). Another factor is the presence of many non-expert buyers, which makes it easier to escape detection as any one buyer can only deal with a few sellers. Additionally, high entry barriers make it difficult to accommodate new entrants, for instance, the prerequisite of few sellers is likely as it makes it easier to sustain a cartel. Another factor is cross-ownership and links among competitors [88], where the scope for collusion is enhanced if a firm has some form of participation in a competitor, even without controlling it. For instance, with a representative sitting on the board of directors of a competitor, it might be easier to coordinate pricing and marketing policies. Even without representation on the board, there is less incentive for the two firms to compete in the marketplace, as any profit or loss will affect both. Finally, symmetry among firms [88], in dimensions such as market shares, variety of products, and technological knowledge, is seen to facilitate collusion. This is explained by the fact that a more equal distribution of assets relaxes the incentive constraints of small and large firms and would help collusion.

2.8.1.2 Product Characteristics

For products where price is the most important aspect of competition, it is easier to organise a cartel where there is limited scope for competition in quality, service, and delivery. Moreover, intermediate products make it easier to sustain an agreement as buyers pass on increases in the cost of intermediate products in their own selling price. Another characteristic takes the form of trade buyers, who use price signalling to inform the trade in advance of price changes. The regularity and frequency of orders also give an indication of collusion, with regular and high frequency orders facilitating collusion [88]. Regular orders ensure regular collusive profits, compared to an unusually large order, while a high frequency of orders allows for a timely punishment in reaction to a deviation. Buyer power [88] is a factor that affects the sustenance of collusive prices, which depends on the concentration of buyers and their bargaining power. For instance, a strong buyer can stimulate competition among sellers and by concentrating its orders into large infrequent orders, it can also break collusion.

2.8.1.3 Demand Conditions

When the market demand price is inelastic, profits can be increased if the price can be raised to the monopoly level by agreement. This condition, where the seller's brands have close substitutes results in price reductions by one seller, and may have retaliatory effects on other competitors, eliminating any advantage. Thus, an agreement between sellers may organise and sustain a cartel. If the demand is discontinuous or volatile, suppliers are encouraged to share out business, in order to avoid bouts of severe price competition, also leading to collusion. A closely related factor, demand evolution [88], can affect collusion depending on the demand movements, especially if they are observable or not. A stable demand might help to sustain collusion as it helps increase the observability of the market.

2.8.1.4 Cost Conditions

Similar technological and input costs can facilitate agreement among firms since price differences are less explicable by differences in costs. Another factor is to incur fixed costs on a high proportion of the total costs. If the demand is discontinuous, high fixed costs increase the riskiness of price competition and hence increases the incentive to collude. The stability of costs also facilitate the making, policing and sustaining of a cartel agreement.

2.8.1.5 Organisational Conditions

Agreements are more easily reachable in a hierarchical organisation. It may also be possible to have collusion due to local operations, when it may be easier for local management to organise local cartels without the knowledge of top management. Other cooperative activities, such as the participation of industry-wide committees on standards and other technical matters, including market forecasting, may move to cartel agreements. Another factor concerns the multi-market contacts [88], where the same firms meet in more than one market, viewed as facilitating collusion. One example is the evidence of airline fares being significantly higher on routes where there exist a number of carriers that have contacts on several routes [23].

2.8.2 Agent-based Perspective

We view collusion as occurring in centralised and decentralised systems, and within each, various solutions have been proposed to address collusion issues.

2.8.2.1 Collusion in Centralised Systems

Centralised systems include centralised reputation systems, such as eBay[2] and Amazon[3], where reputation values about individual agents are collected and managed by a central system and every user in the system sees the same reputation value for another user. In these centralised systems, members have a global view of the entire system and this view is unique to all.

Jurca [52] proposes a method for designing incentive-compatible, collusion-resistant payment mechanisms, by using several reference reports. The idea behind deterring lying coalitions is to design incentive-compatible rewards that make honest reporting the unique or at least the "best" equilibrium. Meanwhile, Lian *et al.* [66] report on the analysis and measurement results of user collusion in Maze, a large-scale P2P file-sharing system. Their aim is to observe user collusion in P2P networks that use incentive policies to encourage cooperation among nodes. They search for colluding behaviour by examining complete user logs and incrementally refine a set of collusion detectors to identify common collusion patterns. They found collusion patterns that are similar to those found in Web spamming.

Wang and Chiu [131, 132] propose to use social network analysis in online auction reputation systems to analyse the underlying structure of the accumulated reputation score and its corresponding transactional network. They demonstrate that network structures formed by transactional histories can be used to expose underlying opportunistic collusive seller behaviours. Transaction logs and social relationship structures are used to reconstruct the relationship profiles to supplement the lack of demographic data in the online environment. In using social network analysis, there is a need to

[2] http://www.ebay.com

[3] http://www.amazon.com

identify ill-intended users, who leave interaction footprints when forging their credibility with additional information process resources and activities. Wang and Chiu have used real world blacklist data, consisting of suspended fraudulent accounts collected from the Yahoo Taiwan Inc. online auction site. However, they have found that the lack of cooperation from online auction hosts is a limitation to data collection and prediction capability as the hosts would have more detailed information to help verification of information.

2.8.2.2 Collusion in Decentralised Systems

In decentralised systems, such as P2P systems, trust and reputation information for members is collected and stored across the network by each individual member to help in predicting their future interactions. Moreover, individual members do not have a global view of the whole system.

TrustGuard [119] is a framework designed to provide a dependable and efficient reputation system that focuses on the vulnerabilities of the reputation system to malicious behaviour, including strategic oscillation of behaviour, shilling attacks, where malicious nodes submit dishonest feedback and collude with one another to boost their own ratings or bad-mouth non-malicious nodes, and fake transactions, which can lead to fake feedback. The main goal of TrustGuard's safeguard techniques is to maximise the cost that the malicious nodes have to pay in order to gain advantage of the trust system. The behaviour of non-malicious and malicious nodes is defined using game theory. The problem of fake transactions is tackled through having feedback bound to a transaction through the exchange of a transaction proof, such that feedback can be successfully filed only if the node filing the feedback can show the proof of the transaction. To deal with the problem of dishonest feedback, a credibility factor is proposed that acts as a filter in estimating reputation-based trust value of a node in the presence of dishonest feedback.

In the domain of grid computing, Staab and Engel [121] propose a collusion detection algorithm based on correlated outcomes in votes. In this context, a master assigns computational tasks to resources, known as workers, which are expected to execute those tasks and return the results. The approach uses information about the frequencies of pairs of workers appearing together in the majority/minority of votes and how often they appear in opposite groups. The concept of majority voting views the majority of a vote to be the strictly largest group of workers that returned identical results. They found that correlation can be used to differentiate between honest and malicious workers. They also propose an algorithm based on graph clustering to discover the division between honest and malicious workers. From their experimental study, the performance of the Markov Cluster Algorithm is good for unconditional colluders, which are those workers who always try to collude, every time they are involved in a vote.

2.8.3 Synthesis

Open issues, such as collusion, still need to be resolved in decentralised multi-agent systems. The main strategy to detect collusive behaviour, as used in centralised systems, is to have a global view of the system in order to identify the possible colluding agents. However, such a global view is not available to individual agents in a decentralised MAS, as there is no central management of agent information. Despite the limitations of an agent's local view of its environment, we believe that the local view can be complemented by recommendation information about other agents to form an extended view, so that individual agents can have access to a relevant set of information concerning their own transactions. Trust and reputation information, together with the agent network, can build and maintain the extended localised view of the agent environment.

2.9 Summary and Conclusions

In this chapter, we have discussed the relevant work related to our aim in this thesis of supporting collusion detection using rich interaction and recommendation information to analyse agents' social networks. The most relevant trust models in the literature have some limitations with respect to the thesis aims. Ntropi [1, 2] gathers interaction information from a range of trust sources, including indirect recommendations. However, it does not gather information within the multiple dimensions of trust and reputation at a service-level, nor does it consider the recency of interactions for trust assessment. Ntropi does not represent agents' social networks from the interaction information and thus does not use that information in further reasoning about issues such as collusion. MDT-R [36] considers trust information from direct interactions and recommendations, except for indirect recommendations. It also fulfils the other trust model characteristics, despite the relevance characteristic being only partially considered through the use of a trust threshold by witnesses. Like Ntropi, MDT-R does not represent the agents' social networks to tackle the issue of deception. The next model relevant to our work is ReGreT [106, 104], which fulfils most of the trust model characteristics except for the relevance characteristic. Although ReGreT represents the social networks of agents through agent neighbourhoods, the building and maintenance of those networks are not specified and are not used for collusion detection. None of these models use the trust model characteristics together, to reason about their social networks for detect collusion. We address this issue by proposing a trust and reputation model that takes into account all the trust model characteristics to form a solid base for reasoning and analysing the agents' social networks.

Agents in multi-agent systems evolve in a social environment where they interact with others for the execution of tasks and exchange of information. Some trust models have tapped into the agents' social network to help understand their interactions and relationships [46, 106, 142]. We improve the use of these networks by describing how agents can build and maintain a network model of the agents in their environment from the interaction and recommendation data they gather. Additionally, we explore how agents can use this agent network model to help them in their decision making for selecting interaction partners, and information sources.

Deception is a complex issue that agents face as they compete in an environment where individuals seek their own benefits. Various forms of deception have been the subject of research in various domains, including lies in agent communication [91], collusion detection [119, 121] and prevention [52]. In our research, we seek to address some of the limitations of these solutions, which are application-specific and often require more information to be available to the agents in order to counter deception. Thus, we aim to enable agents to counter deception, specifically collusion, by using information they already gather for deciding on their interactions and applying it to discover collusive agents in their environment.

The next three chapters look in detail at each of the aims of this thesis, with respect to the limitations of the trust models and existing approaches in resolving the issues we are focusing on. Chapter 3 introduces our trust and reputation model, aimed to improve trust assessment as well as to enable individual agents to enhance their decision making by using trust, reputation, and social relationship information to extract useful information, for collusion detection, for example. Our model is based on the trust model characteristics described in Section 2.3.

Chapter 3

Trust and Reputation Model

3.1 Introduction

In Chapter 2 we discussed existing trust models and their limitations with respect to our aims of supporting collusion detection using routine interaction and recommendation information, represented by agents' social networks. In response to this need, we present a model of trust and reputation that aims to improve the accuracy of an evaluator's assessment of the trustworthiness of other agents for each of the specific services they provide. The model also ensures that agents gather a rich set of interaction and recommendation information for further reasoning of agent relationships. We make two key contributions that build on and extend existing approaches to trust and reputation. The first contribution is a trust model that evaluates trustworthiness for individual services in different service characteristics, and utilises a broad range of trust information sources in a flexible manner. Independently assessing agent services allows evaluator agents to accurately predict the future behaviour of potential interaction partners with respect to the specific service being considered. Additionally, while agent-level trust indicates how trustworthy an agent is as a whole, trust at the level of service characteristics conveys the reliability of an agent with respect to characteristics such as timeliness, cost and quality for each service type offered by the agent. Using a wide spectrum of trust information sources, such as direct interactions, direct and indirect recommendations, in a flexible manner allows an agent to assess the trustworthiness of another using all available trust information sources, without causing the assessment to fail if some sources are unavailable.

Our second contribution is to combine the recency of interactions and the relevance of recommendations in a single model for trust assessment. The more recent agent interactions ensure that the most up-to-date agent behaviour is modelled so that future behaviour can be accurately predicted. The recency of recommendations is one of the elements that makes third party opinions relevant. The other elements that determine the usefulness of recommendations are the experience of the witnesses and how trustworthy the witness is in giving recommendations. Recency and relevance further improve the accuracy of trust evaluation, when used together with trust assessment in individual agent services and service characteristics that is flexibly based on a wide spectrum of trust information sources.

We have taken into account the key trust model characteristics, as discussed in Section 2.3 while developing our model. The aim of the model is also to allow agents to gather a richer set of interaction and recommendation information for future analysis. Our trust model does not consider collusion in the agent environment as part of its decision making. However, it supports collusion detection by providing sufficient information to help in the reasoning process. In the remainder of this chapter, we will first give an overview of our proposed trust and reputation model and the related trust models. We then describe the features of our model and how they can be used. In

Section 3.5, we present the implementation and the evaluation of our model. Finally, in Section 3.6.2 we conclude the chapter.

3.2 Overview

The data collection component of our trust-based social mechanism consists of our trust and reputation model, which builds on and extends several existing trust models. We now present the features of our model and describe the base models that have influenced their inclusion in our model. As first presented in Section 2.3, the four main characteristics of a trust model are the use of a wide range of trust information sources, the assessment of trust in individual service characteristics, the consideration of the recency of interactions and taking the relevance of recommendations into account.

3.2.1 Model Features

These essential characteristics of a trust model allow agents to accurately assess the trustworthiness of potential interaction partners. Following the survey of relevant trust models presented in Section 2.5, we describe the trust model characteristics with reference to the most relevant trust models.

3.2.1.1 Wide Range of Trust Information Sources

Trust models, including FIRE [46], Ntropi [1], ReGreT [106], HISTOS [142], and L.I.A.R. [91] support the use of a wide range of trust information sources to help assess trustworthiness. Trust information is gathered from direct interactions with the service providers. When this type of information is insufficient, recommendations are requested from third party agents, whether they have directly interacted (direct recommendation) or indirectly interacted (indirect recommendation) with the agent under consideration.

With a wide range of information sources, the evaluating agent can make a more accurate evaluation of the trustworthiness of the target agent. For this reason, our trust model also allows agents to use their history of past interactions and direct, as well as indirect recommendations from trusted parties to assess trustworthiness.

3.2.1.2 Trust Assessment at Service Level

Being able to assess trustworthiness not only at agent level, but also at service level gives an agent additional information to base its decisions on. Provider agents do not necessarily perform tasks to the same standard and for a particular task, their performance in different dimensions may also vary. Among the trust models reviewed, only a handful consider trust assessment at service level, namely ReGreT [106], MDT-R [36], and SIR [83]. We believe that agents will be able to react more appropriately to changes in their environment, due to agents changing their behaviour and they way they provide services.

3.2.1.3 Recency of Interactions

Past interactions are valuable for informing future ones as they indicate how agents have behaved in the past and how they are likely to behave in the future. However, agents can also change behaviour and older interactions do not reflect this. For this reason, taking into account the recency of interaction is important to ensure that agents get the most up-to-date information about the dynamic behaviour of their interaction partners. ReGreT [106], SIR [83], SPORAS and HISTOS [142], MDT-R [36], and FIRE [46] include recency, while L.I.A.R. [91] and Mui *et al.* [89] only partly consider it in their solutions.

3.2.1.4 Relevance of Recommendations

Recommendations are a valuable source of information about the trustworthiness of other agents in the environment. However, not all recommendations received by an agent will be as useful for better assessing other agents. Relevance is dependant on other aspects, such as recency and recommender trust. Ntropi [1] and the referral system by Yu and Singh [140] make use of relevance, while MDT-R and L.I.A.R. [91] partially consider it.

3.2.2 Base Trust Models

Our trust and reputation model is broadly based on MDT-R [36], introduced in Section 2.5.7, as it already features many of the characteristics that we require in a trust model. Multidimensional trust, direct recommendations and recency are used, together with the partial use of relevance. The next base model is Ntropi [1, 2], introduced in Section 2.5.3. Ntropi also uses trust and reputation, including indirect recommendations. The relevance characteristic is applied to recommendations and although the technique used is different to ours, the similarity resides in taking into account the context of the recommendation and the reliability of the witnesses along the recommendation chain.

3.2.2.1 MDT-R

MDT-R [36] represents multiple dimensions of trust and reputation, the nature and number of these dimensions depending on the priorities and motivations of the agents using that model. The trust in a particular dimension d for the agent a_t, denoted as $T^d_{a_t}$, is defined as the likelihood in dimension d of a_t executing the task such that expectations are met. Information is shared among agents in form of interaction summaries which retain a maximum amount of original information to minimise the subjectivity of interpretation. Our model adopts the multidimensionality of trust and recommendations, as well as the sharing of interaction summaries from MDT-R. The trust value in each dimension is updated after an interaction, and even when interactions do not occur, over time, the value is reduced through the use of a decay function that brings the trust value to its original value, when the agent was first interacted with. The recency of interaction information is reflected by the decay of trust over time. We use the recency of interactions in a similar way to give more weight to more recent interactions. We extend MDT-R by including information on the experience of witnesses when sharing interaction summaries. This allows an evaluator to more accurately select witnesses, and thereby providers, as it further reduces the subjectivity of interpretation.

While Ntropi uses discrete trust levels, MDT-R supports trust in a continuous level, represented by real numbers as the subjective probability values. While continuous values of trust allows updates to be more easily calculated, MDT-R also uses stratified trust for simplified trust comparisons, which are especially useful to reduce the risk of misinterpreting insignificant numerical differences as important. The use of a number of strata classifies trust values into groups for easier trust comparisons but with the flexibility of adjusting the precision of comparisons by increasing the number of strata to represent trust values. We stratify trust in our model in a similar way. Our model also adopts the way the parameters involved in trust assessment are combined in MDT-R. We extend this approach to indirect recommendations, which are not used in MDT-R.

Recommendations are requested from trusted agents and each witness a_w provides its experience of interacting with the target a_t via the number of interactions that have met expectations ($I^{d+}_{a_w a_t}$) and the number that have not ($I^{d-}_{a_w a_t}$). The performance of the target agent is then calculated by combining the direct trust and recommendations using a number of weights for the factors that influence the interaction. These performance values for the potential interaction partners are then compared to select the most trustworthy agent.

3.2.2.2 Ntropi

As introduced in Section 2.5.3, Ntropi [1] models trust through discrete levels, which have loosely defined semantics. Agents refine these semantics with experience and time since they are influenced by the nature of the agent relationships. Trust relationships exist in four phases: unfamiliar, fragile, stable and untrusted. Each phase underpins a number of policies whereby agents provide services or not depending on the trust level. The transition between the levels is based on thresholds, that is, the number of interactions at a phase before moving to another phase.

Ntropi uses the semantic distance between an evaluator's opinion and that of a witness to adjust future recommendations from that witness if needed. Thus, recommendations are always translated to match the recommendations with the experience the evaluator has with regards to the witness. Our model considers the relevance of recommendations, in a similar way to improve the selection of witnesses and to assign them appropriate weights when calculating reputation. Recommendations in Ntropi include indirect recommendations along a recommendation chain. We also consider indirect recommendations as a source of trust information in our model, to gather a richer set of information about the agents involved in the evaluator's interactions.

3.2.3 Application Example: Supply Chain Management

In the ideal case, a supply chain facilitates the availability of the right amount of the right product at the right place with the minimal amount of inventory across the network [21]. Most supply chains are associated with several firms and these can be viewed as being represented by software agents in an e-supply chain. Each agent has its own goals and objectives and makes decisions based on the available information in its environment. A supply chain configuration consists of the selection of services based on their characteristics, such as cost, profit, and timeliness, in order to achieve a certain objective, for instance, that of delivering a product and minimising cost. For each order, there are a number of possible supply chain configurations that can deliver the product, the number of configurations depending on many factors, including the number of products, suppliers, and supply chain stages. In most real world situations, it is not possible to evaluate every single configuration, due to limits on the resources available.

We have taken the example of a supply chain to illustrate the use of our mechanism in partner selection. For example, in some environments, agents need to rely more on reputation information, and this can be reflected in the weights each source of trust information is allocated for trust evaluation. Another example is the decay rate for trust values. In peer-to-peer systems where agent behaviours can change very quickly, it might be beneficial for an evaluator's trust in another to decay quickly towards its initial trust value.

3.2.3.1 Computer Hardware Supply Chain Scenario

The personal computer (PC) industry is a dynamic industry that faces short product life cycles [65]. PC manufacturers design, manufacture and market a range of computer systems, including desktops and notebooks. They source their components from a number of suppliers and can also outsource the assembly of components to other companies. We will consider the case of an computer hardware e-supply chain, where the component suppliers provide products to customers, which include computer systems manufacturers, computer shops and computer parts resellers.

In a two-stage supply chain, a customer obtains components directly from the supplier, for instance the RAM card and hard disk. A customer typically needs to purchase different types of components and there are several suppliers that can do the job. In an e-supply environment, many computer manufacturers and resellers need to interact with various suppliers to source the necessary components to build or sell their systems. Customers can also act as suppliers for partly-assembled components, for example, a computer shop sells partly-built computers, to which components, such as hard disks and memory chips, need to be added on. In this competitive industry, there are many stakeholders and they each try to get the most benefits and attain their individual goals and objectives.

For illustration purposes, we consider the case of a small e-supply chain, consisting of 30 computer parts customers and 20 component suppliers. Some customers can also be suppliers for part-assembled components and together, they form a population of 50 agents in the e-supply chain environment. Suppliers and customers are assumed to be of variable reliability and performance, for example, a supplier for monitors can produce defective monitors half of the time. Similarly, a customer may be late in paying the supplier. Agents exchange goods as well as information. Information exchange includes order specifications and opinions about products and suppliers. In the agent population of 50, not all agents interact with one another, as agents only need to deal with those agents whose services they need and vice versa.

In an environment where suppliers have variable performance and reliability, a customer needs to ensure that it interacts with the most trustworthy supplier for the required product to minimise costs and production times. Suppose that a computer systems manufacturer, denoted as customer C_1, needs to purchase computer monitors and there are 3 suppliers, S_1, S_2 and S_3, with different offers. The cheapest supplier is not necessarily the best choice as it might also be the one providing the worse quality products. Using trust and reputation, C_1 can make the decision on which supplier to use, based on previous interactions and recommendations from other agents.

3.2.3.2 Terminology

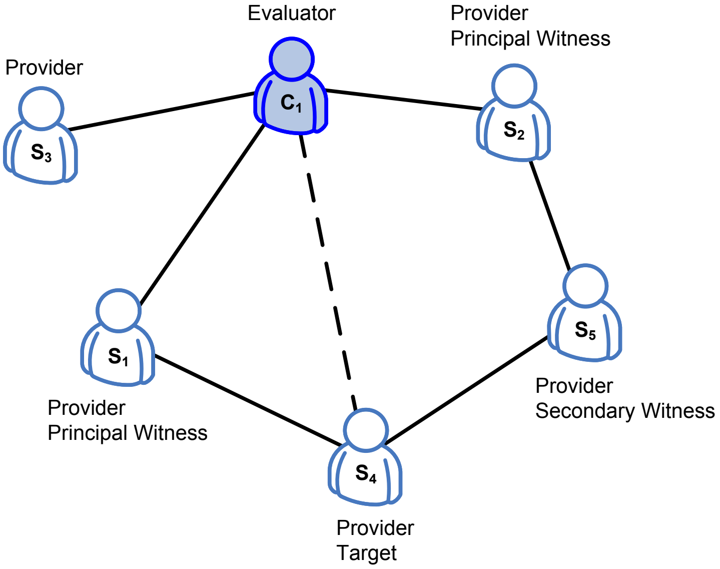


Figure 3.1: Representation of Agent Terminology.

We now define the terminology that we will use throughout the thesis to describe the different agents and their roles with reference to the supply chain scenario in Section 3.2.3.1 and Figure 3.1. An *evaluator* is the agent assessing the trustworthiness of potential interaction partners. A *provider* agent is one which offers some services and could potentially become one of the evaluator's interaction partners. The *target* is a particular provider agent that the evaluator is interested in assessing. In gathering trust information from agents, the evaluator can request for the opinions of *witnesses* who have interacted with the target, or know someone who has. In Figure 3.1 the evaluator,

C_1, has directly interacted with three providers: S_1, S_2, and S_3, as shown by the solid lines linking the agents. It is currently assessing the target agent S_4, the dashed line indicating that there has been no interaction yet. To help with this assessment, C_1 requests the opinions of S_1, who is a direct witness, having interacted with S_4, and of S_2, who is the indirect witness linked to the witness S_5 who has interacted with the target. In this recommendation chain, S_2 is known as the *principal witness*, as the evaluator has made the request to S_2. Agent S_5 is the *secondary witness* as it is a witness but is not the agent that has been approached by the evaluator. If the recommendation chain is longer than two, the witnesses along the chain who are not the principal witness are referred to as *intermediate witnesses*.

3.3 Model Description

We propose a trust and reputation model that has four main properties. The first property is the combination of direct past interactions with third party recommendations for trust assessment, with the distinction between direct and indirect recommendations. This allows the evaluator agent to accurately assess trustworthiness in different situations, using as many trust information sources as are available. The second property is that the trustworthiness of agents is assessed in their individual service characteristics. This level of granularity allows the accurate trust assessment in the service characteristics that the evaluator deems important. Next, the third property concerns the recency of interactions. Recency plays an important part in the accuracy of trust assessment as more recent interactions tend to predict more accurately the future behaviour of those agents. Finally, the fourth property is the relevance of interactions, which ensures that the evaluator takes into account all the factors that cause the trust information source to be reliable. As our model is based on MDT-R, many of the algorithms for calculating trust and reputation are similar. Differences appear where our model uses indirect recommendations and the relevance of recommendations in the assessment. Indirect recommendations result in the recommendation trust of the principal witness being updated and this changes only minimally from MDT-R. We now describe how the trust model properties are used in the assessment of agent trustworthiness.

3.3.1 Service-Level Assessment

Past service interactions give a good indication of the reliability of a provider. Past interactions with the provider for a particular service allow more accurate assessment for that particular service. However, taking into account all the services interactions enables the evaluator to assess the agent as a whole.

3.3.1.1 Direct Service Interactions

Direct service interactions occur between a provider and a consumer. Referring to the application example introduced in Section 3.2.3, let us suppose customer C_1 is the evaluator and wants to assess which of the three suppliers is the most trustworthy for future transactions. C_1 has interacted with two of the suppliers previously, S_1 and S_2. From its history of past interactions, C_1 can assess how trustworthy each supplier has been, based on service characteristics, such as successful delivery, timeliness and cost. For a similar number of interactions, suppose that supplier S_1 has been trustworthy in all the important service characteristics 90% of the time, compared to 50% for supplier S_2. From this comparison, C_1 can decide to use supplier S_1 for its next order of computer monitors.

Trust information is captured in multiple dimensions, as in MDT-R [35, 36]. The separation into several dimensions enables information about specific service characteristics to be preserved. The subjectivity of trust, especially from recommendations, is an obstacle to making full use of the information obtained from witnesses. Sharing multi-dimensional trust information within interaction summaries [36], instead of calculated trust values decreases subjectivity. The dimensions correspond to the necessary characteristics that define a service. Multiple criteria decision analysis (MCDA) describes a collection of formal approaches which seek to take explicit account of multiple criteria in helping individuals or groups explore decisions that matter. In our model, agents do not use MCDA to choose the service dimensions to represent. Human analysts may use MCDA to decide on the dimensions the agent should use to define service characteristics. The trust an evaluator a_e has in a provider a_p about a particular service s in a dimension d, is denoted as $T^{sd}_{a_e a_p}$. Any number of dimensions can be used, but for the purpose of illustration, we consider that an evaluator a_e models trust in provider a_p along four dimensions [36]:

- success ($T^{sS}_{a_e a_p}$): the likelihood that a_p will successfully execute the task,
- timeliness ($T^{sT}_{a_e a_p}$): the likelihood that the task will be performed no later than expected,
- cost ($T^{sC}_{a_e a_p}$): the likelihood that the cost of performing the task will not be more than expected, and
- quality ($T^{sQ}_{a_e a_p}$): the likelihood that the quality requirements of the task performed by a_p will be met.

These trust values are derived from the past interactions of a_e and a_p. The evaluator stores information about each interaction in which a_p has performed a task on its behalf. Information about each interaction includes the service characteristics offered by a_p, as well as the actual values obtained on completion. The derived trust values refer to a specific task and so this is a type of situational trust. A successful interaction is one where a_p delivers results, irrespective of whether the other three characteristics are met. Meanwhile, a positive interaction with respect to the dimensions of timeliness, cost and quality refers to a_p performing as expected or better, and conversely for negative interactions. Trust values are calculated when the evaluator needs to make a decision about whom to interact with. The range of the trust values in each dimension is $[-1, +1]$, where -1 means complete distrust and $+1$ means complete trust. The evaluator stores a history of past interactions with each provider for each task type. We denote the set of interactions in the history for the service type s as H_{i_s}. The size of the history corresponds to the number of interactions that the evaluator stores for future reference. The history size is determined by the system architect depending on the amount of resources available, or by the number of interactions that are useful for analysis. In future work, evaluators should be able to change the size of the history on a per-target basis to enable agents to store only the required information to assess trust.

In our model, we consider three types of trust, as in Marsh's formalism [81]. Firstly, an agent has an *initial trust*, $initialT$, in another agent when it first starts interacting and has no previous interactions. It is based on the agent's disposition to successfully interact with another agent and is denoted as $initialT = disposition_{pass}$. The term $disposition_{pass}$ represents the success disposition of the evaluator, which is an indication of its behaviour as a result of a successful interaction.

Another type of trust is *situational trust*, ST, is the trust in the target about a particular service. The situational trust value $ST^{sd}_{a_e a_p}$ is a function of the history of interactions of evaluator a_e with target a_p: Trust in another agent for a specific service type is referred to as *situational trust* (ST), and it is the proportion of successful interactions compared to the negative interactions the consumer has experienced with the provider. When the assessment is done in individual service characteristics, this is denoted by the letter following the service type s, for example sQ refers to the quality characteristic of the service.

$$\begin{aligned} f(Interactions) &= ST^{sd}_{a_e a_p} \\ &= \frac{count^{sd+}_{a_e a_p} - count^{sd-}_{a_e a_p}}{count^{sd+}_{a_e a_p} + count^{sd-}_{a_e a_p}} \end{aligned} \tag{3.1}$$

where $count^{sd+}_{a_e a_p}$ is the number of positive interactions the evaluator a_e has experienced with provider a_p, of service s, in dimension d, and $count^{sd-}_{a_e a_p}$ is the number of negative interactions. Equation 3.1, adapted from [37], is used to calculate trust, and interaction summaries are shared with other agents to preserve the components of the equation, which are the number of positive and negative interactions in each service dimension.

When there are insufficient or no past interactions for a particular service, it is useful to assess the trustworthiness of the provider as a whole. This type of trust, *general trust* (GT) is the average of all interaction experiences in the different service types. The following equation is used in MDT-R and originates from Marsh's notion of general trust [81].

$$\begin{aligned} f(Interactions) &= GT_{a_e a_p} \\ &= \frac{\sum_{s=1}^{count^s} ST^{sd_s}_{a_e a_p}}{count^s} \end{aligned} \tag{3.2}$$

where $count^s$ is the count of all the service types. We use only the success dimension to simplify calculation, since completing a interaction successfully has overriding priority when obtaining an agent's overall trustworthiness, in the cases where past experience in specific service types are not available.

Coming back to our example scenario presented at the beginning of this section, suppose that customer C_1 wants to assess the trustworthiness of the computer monitor suppliers before selecting the most trustworthy. Since C_1 has previously purchased monitors from both suppliers S_1 and S_2, the customer can calculate their situational trust concerning the service of selling monitors. Assuming that C_1 has had 20 interactions with supplier S_1 about monitors before, of which 90% were successful, the situational trust of S_1 in say, the success dimension is calculated as $ST^{sS}_{C_1 S_1} = \frac{18-2}{18+2} = 0.8$. Similarly, if C_1 has had 6 interactions in the past with supplier S_2 concerning monitors, the situational trust in the success dimension based on 50% success rate is $ST^{sS}_{C_1 S_2} = \frac{3-3}{3+3} = 0$. C_1 has however, never interacted with supplier S_3 for monitors, but it has previously purchased two other products, optical mice and SD memory cards. Let's suppose that $ST_{C_1 S_3}$ for optical mice is 0.2 and that for SD memory cards is 0.3. For the supplier S_3, C_1 can calculate its general trust as $GT_{C_1 S_3} = \frac{\sum_1^2 ST^{sS}_{C_1 a_p}}{2} = \frac{0.2+0.3}{2} = 0.25$.

3.3.1.2 Direct Recommendations

The reputation of a provider is calculated from a number of recommendations. Direct recommendations originate from witnesses which have directly interacted with the provider for service provision. These agents, referred to as principal witnesses, share their history of interactions concerning that provider, with the evaluator. For each recommender, the function of the interactions it shares is presented as:

$$f(Interactions) = \frac{I^{sd+}_{ia_p} - I^{sd-}_{ia_p}}{I^{sd+}_{ia_p} + I^{sd-}_{ia_p}} \tag{3.3}$$

where $I^{sd+}_{ia_p}$ is the number of interactions of the witness a_r with the target a_p for service type s, for which a_p has met expectations for the dimension d, and $I^{sd-}_{ia_p}$ is the number where expectations are not met. The Equation 3.3 is similar to the one for situational trust.

3.3.1.3 Indirect Recommendations

The model allows agents to provide indirect recommendations to requesting agents. The evaluator does not distinguish between direct and indirect recommendations for the evaluation of recommendation trust. This is due to the evaluator assessing the trustworthiness of the principal witness in giving recommendations, and not the secondary witnesses. Therefore, it is assumed that the principal witness will give recommendations from further along the recommendation chain with care, as its reputation as a witness depends on it.

3.3.2 Recency

The recent interactions and recommendations indicate the most likely future behaviour of agents. This is especially the case in dynamic environments, where agents tend to change behaviour and the recent interactions will be the most relevant. For this reason, the trust model allows evaluator agents to take the more recent interactions into consideration by using a trust decay function that gives higher importance to the more recent interactions.

3.3.2.1 Direct Service Interactions

Direct service interactions occur between a provider and a consumer and the recency of the interactions is highlighted in the decay of the trust value towards the initial trust value $initialT$. The Equation 3.4 is from the MDT-R model. In our model, we expand on this to give Equation 3.5, to demonstrate that the trust decay rate is a function of several parameters.

$$\begin{aligned}
f(Recency) &= decay(ST^{sd}_{a_e a_p}) \\
&= ST^{sd}_{a_e a_p} - (ST^{sd}_{a_e a_p} - initialT) \times \omega_{td} \qquad (3.4) \\
&= ST^{sd}_{a_e a_p} - (ST^{sd}_{a_e a_p} - initialT) \times \\
&\quad f(disposition_{pass}, t_{now}, t_{last}, frequency(i_s), \omega_{H_{i_s}}) \qquad (3.5)
\end{aligned}$$

where t_{now} is the current time, t_{last} is the time at the last interaction, $frequency(i_s)$ is the average frequency of interactions of that service type. The weight $\omega_{H_{i_s}}$ is assigned to an interaction according to recency; the more recent the interaction, the more weight it has, since more recent interactions give a more accurate reflection. In Equations 3.4 and 3.5, the general trust can be used if the situational trust is not available. Trust in a_p decays towards the initial trust value of a_e, rather than the actual agent behaviour because the lack of recent interactions does not allow the evaluator a_e to have an accurate picture of the agent a_p. It therefore relies more on its own disposition to trust another agent. The lack of recent interactions may have several reasons, including the provider being unavailable, the evaluator not requiring the types of services offered by the provider, or the evaluator having the opinion that the trustworthiness of the provider is too low for interaction. When the evaluator does not interact for a period of time with that provider, it might be missing out on the benefits of interacting with it, especially if the provider's behaviour has changed for the better. The decay of trust towards the initial trust allows the evaluator to attempt to interact again with the provider.

3.3.2.2 Direct and Indirect Recommendations

The recency of recommendations can be used in the evaluation of reputation if available. The recency will be a function of the weight of the recommendations relating to their recency.

$$f(Recency) = \omega_{H_{ir}} \tag{3.6}$$

3.3.3 Relevance

The relevance property of the trust model concerns the relevance of service interactions and recommendations for the assessment of trustworthiness. For accurate evaluation of trustworthiness, only the most relevant interactions should be taken into account.

3.3.3.1 Direct Service Interactions

For direct service interactions for the evaluation of direct trust, the recency of service interactions ensures that the most recent interactions are used to predict the future behaviour of agents. Evaluator agents also use interactions in the service type required first to assess the trustworthiness of a potential provider (situational trust). If these are not available or insufficient, the evaluator can use other service interactions it had with the provider for decision making (general trust).

3.3.3.2 Direct and Indirect Recommendations

The relevance of recommendations from third party agents is an important factor to take into consideration. Witnesses share their experiences with the evaluator, but they do not guarantee that if the evaluator chooses to use the services from that particular provider, it will have a similar experience as the witness. For this reason, the evaluator needs to assess which recommendations are most appropriate for its purposes.

Recommendation trust estimates the accuracy of recommendations and the trustworthiness of witnesses in giving recommendations. The relevance of recommendations is a function of the recency of recommendations, the experience of the witness and the trustworthiness of the witness in giving recommendations. Relevance is a novel aspect of our model and this is not used in MDT-R.

$$\begin{aligned} f(Relevance) &= WRR \\ &= \left(\frac{t_{curr} - t_{median(H_{i_s})}}{t_{curr}}\right) \\ &\quad + \frac{max_{WI}}{total_{WI}} + RT_{a_e a_r} + \omega_{RT_c} \end{aligned} \tag{3.7}$$

where WRR stands for witness reputation relevance, t_{curr} denotes the current time and $t_{median}(H_{i_s})$ is the recorded time of the median interaction as provided by the witness a_r for interaction with target a_p. The inclusion of time in the calculation indicates the recency of the interactions on which the recommendation is based. The maximum number of interactions that the witnesses have used when giving recommendations is max_{WI}, and $total_{WI}$ is the total number of interactions actually used in that recommendation. The confidence of the evaluator a_e in its recommendation trust $RT_{a_e a_r}$ in the witness a_r is denoted as RT_c and the confidence weight ω_{RT_c} shows the amount of influence of this recommendation confidence. This equation is used in our model as a way of handling the relevance of reputation.

3.3.4 Aggregation

This section describes how the different sources of trust information are aggregated to evaluate the trustworthiness of agents.

3.3.4.1 Direct Trust

Direct trust from direct interactions is calculated by the evaluator a_e as a function of its direct interactions with the target a_t, and is a function of the freshness of these interactions.

$$\begin{aligned} f(\mathit{Trust}) &= f(\mathit{Interactions}) \times f(\mathit{Recency}) \times f(\mathit{Relevance}) \\ &= ST - (ST - \mathit{initialT}) \times \mathit{disposition}_{pass} \\ &\quad \times \left(\frac{t_{now} - t_{last}}{\mathit{frequency}(i_s) \times (\omega_{H_{i_s}} \times 10)} \right) \end{aligned} \tag{3.8}$$

As proposed in MDT-R [37], trust values in our model are stratified at the time of comparison. When using numerical values, there is a risk of considering even insignificant differences in values to be important, and stratifying trust reduces this risk. Stratified trust is only used for comparisons and is not communicated to others. In our model, the number of strata used can be specified to allow for different levels of sensitivity. For example, if the number of strata is 10, then trust values in the range $[0.8, 1]$ are taken to be the same. Thus, if two agents a_β and a_γ are being compared by situational trust in the success dimension, then if $ST^s_{a_\alpha a_\beta} = 0.85$ and $ST^s_{a_\alpha a_\gamma} = 0.95$ both agents are taken to have similar trust values. A larger number of strata ensures a smoother transition between different strata, especially at the boundary between positive and negative trust [37].

3.3.4.2 Reputation

Reputation is the trust of a target as communicated by third parties and can be built from either direct or indirect recommendations. The reputation of a target is sought when the evaluator has insufficient information from its own past experience to make a decision about whether to cooperate. A lack of information may occur for several reasons. For example, consider an evaluator a_e who wants to consider agent a_p for interaction, to perform a service of type s_1. In the first case, suppose a_e has never interacted with a_p before and thus has no experience of a_p's behaviour. Alternatively, suppose a_e has previously interacted with a_p but for a different service type, such as s_2. Another case is when a_e has had too few interactions with a_p, or they are too outdated. In all these cases, a_e can ask the opinions of others who have interacted with a_p, in order to get a more accurate assessment of a_p's trustworthiness. Direct and indirect recommendations can provide useful information about the trustworthiness of the target in meeting its commitments.

In our running example, suppose customer C_1 also requires supplies of hard disks, a recent addition to the component parts it needs. Furthermore, suppose that there are 2 suppliers for this component, namely S_3 and S_4, such that C_1 has purchased from S_3 once before and has not interacted with S_4 previously. With insufficient past interactions to reliably assess the trustworthiness of either supplier, C_1 can complement information from direct trust with recommendations from agents that have previously interacted with S_3 and S_4. In our example, suppose that C_1 has a regular customer C_2, a computer shop, which resells computers and computer parts. Since C_2 stocks hard disks for resale from both suppliers, C_1 can obtain its opinion about these suppliers.

Considering our scenario, suppose that C_1 wants to assess the trustworthiness of suppliers S_3 and S_4, but has insufficient direct interactions with them to make an informed decision about whom to approach for the next order. This time, customer C_2 has not interacted with either supplier, but it knows another agent C_3, which has interacted with both S_3 and S_4. C_2 therefore gives an indirect recommendation about the suppliers to C_1, based on C_3's experience.

When an evaluator requires recommendations for an agent, it must decide which agents to ask. Such agents might have different kinds of experience with the target, and their opinions might not be useful to the evaluator. To decide whom to ask, the evaluator can use *recommendation trust*, which estimates the accuracy and relevance of a witness' recommendation for the evaluator's purposes. Accuracy measures the similarity between the evaluator's own experience and the opinion given by the witness. Meanwhile, relevance relates to how useful the recommendation is based on the recency of the interactions, the experience of the witness, and how trustworthy the witness is in giving recommendations.

FIRE [46] considers whether the witness has sufficient information about the target to give an opinion. An extension to FIRE [45] considers the credibility of the witness in providing opinions about other agents. This enables the evaluator to identify the accuracy of the recommendation by comparing it with its own experience, after an interaction occurs. However, the model does not consider the relevance of a witness' trust information for the evaluator's purposes. In MDT-R, an agent selects witnesses by considering its most trusted interaction partners. However, it does not select witnesses based on the relevance of recommendations and there is no validation of whether the witness has given accurate information. The uncertainty lies in the possible difference in behaviour of the target towards different evaluators. Ntropi [1] considers two factors when dealing with recommendations: (i) the closeness of the witness' recommendation and the evaluator's own judgement about the target, and (ii) the reliability of the witness in giving accurate opinions over time.

Our approach to reputation is influenced by Ntropi's consideration of accuracy and relevance when selecting witnesses. The relevance of recommendations is calculated by taking into account their recency, the experience of the witness, as well as the evaluator's recommendation trust and confidence in the witness. The same mechanism applies to direct and indirect recommendations as the evaluator does not differentiate between the two sources of recommendation. The evaluator's recommendation trust in the principal witness relies on how reliable it is in providing accurate and relevant opinions. As for the accuracy of opinions, this is done for interactions that have taken place following positive recommendations. The evaluator compares the outcome of the interaction with the recommendation previously obtained to assess how accurate it was. The evaluator does not distinguish between direct and indirect recommendation trust and therefore the recommendation trust value represents the trustworthiness of the witness in providing any type of recommendation. Recommendation trust is updated for each agent that has given recommendations. Initially, witnesses have a recommendation trust value equal to their general trust. This is later updated if the evaluator interacts with the recommended provider. The update functions are outlined in Equations (3.9) to (3.12). The evaluator keeps a record of all the witnesss for a task and it updates its recommendation trust in each of them after the interaction with the target.

Equation 3.9 shows the evaluator a_e's update of its recommendation trust RT in witness a_r when $STdiff < \tau$. The difference between the new situational trust value resulting from the interaction and the value recommended by witness a_r is referred to as $STdiff$. For small differences, for instance, $\tau = 0.2$, the recommendation trust increases as it suggests that the recommendation is accurate and reliable enough.

$$update(RT_{a_e a_r}) = RT_{a_e a_r} + increment^{+} \quad \text{if } STdiff < \tau \tag{3.9}$$

$$
\begin{aligned}
increment^{+} &= \left(\frac{RT_{max} - STdiff}{\mid STdiff_{max} \mid}\right) \times \omega_{H_{i_r}} \\
&\quad \times disposition_{pass} \times (RT_{max} - RT_{a_e a_r})
\end{aligned}
\tag{3.10}
$$

where RT_{max} is the maximum recommendation trust, and $STdiff_{max}$ is the maximum difference in value between the resulting situational trust and the recommended value. The threshold τ can be varied according to how close the actual trustworthiness is to the recommended trust for the recommendation to be considered as accurate enough for the evaluator's purposes.

The next two Equations (3.11) and (3.12) show how the recommendation trust is updated if the recommendation is further from the actual interaction.

$$
update(RT_{a_e a_r}) = RT_{a_e a_r} - increment^{-} \quad \text{if } STdiff >= \tau \tag{3.11}
$$

$$
\begin{aligned}
increment^{-} &= \left(\frac{STdiff}{\mid STdiff_{max} \mid}\right) \times \omega_{H_{i_r}} \\
&\quad \times disposition_{fail} \times (RT_{min} - RT_{a_e a_r})
\end{aligned}
\tag{3.12}
$$

where $disposition_{fail}$ is the failure disposition of the evaluator, which is an indication of its behaviour as a result of a failed interaction.

Witnesses provide the evaluator with interaction summaries for a specific task type where available. The summaries contain information such as the number of interactions the recommendation is based on, the recency of these interactions, and the proportion of positive and negative interactions in each trust dimension. If the witness does not have situational trust information, it provides its general trust in the target. The use of interaction summaries is similar to that in MDT-R with the additional sharing of information about recency and experience, which can improve the evaluator's adaptation to changes in the behaviour of target agents. The evaluator combines the different recommendations by applying weights according to how relevant the witness' experience is, compared to the evaluator's. The weight ω_{WRR} is the weight of the witness reputation relevance WRR of witness a_r in providing a recommendation for target a_p.

Thus, the witness reputation WR of target a_p for a service type s in the dimension d, as viewed by evaluator a_e, is a function of the opinions received from witnesses and their respective weights:

$$\begin{aligned} f(Reputation) &= f(Interactions) \times f(Recency) \times f(Relevance) \\ &= WR^{sd}_{a_e a_p} \\ &= \sum_{i=\gamma}^{\epsilon} \left(\frac{I^{sd+}_{ia_p} - I^{sd-}_{ia_p}}{I^{sd+}_{ia_p} + I^{sd-}_{ia_p}} \times \omega_{WRR} \right) \end{aligned} \tag{3.13}$$

where γ to ϵ are the set of selected witnesses for target a_p. The term $I^{sd+}_{ia_p}$ is the number of interactions of the witness a_r with the target a_p for service type s, for which a_p has met expectations for the dimension d, and $I^{sd-}_{ia_p}$ is the number where expectations are not met. The weight ascribed to a witness recommendation is dependent on its experience and its relevance. Thus, the evaluator can include the recommendations in each trust dimension of success, timeliness, cost and quality.

3.3.4.3 Performance Evaluation

The performance value for each potential provider a_p is calculated in a similar way as in MDT-R, with the exception that our model uses the relevance of recommendations WRR, which has been added to the equations below.

$$PV(a_p) = \prod_{i=1}^{n} (f_{a_{p_i}})^{\mu_i} \tag{3.14}$$

where there are n factors and $f_{a_{p_i}}$ is the value for agent a_e in terms of the i'th factor and μ_i is the weighting given to the i'th factor in the selection of the agent's preferences.

To assess trust using only direct trust, the values are stratified and the performance value is given below. The values are stratified or are placed into ranges of values such that small differences in values are not mistaken for being significant.

$$\begin{aligned} PV(a_p) &= (max_C + 1 - a_p^C)^{\mu_C} \times (a_p^Q)^{\mu_Q} \\ &\quad \times \mathit{stratify}(ST^{sS}_{a_e a_p})^{\mu_{tS}} \times \mathit{stratify}(ST^{sT}_{a_e a_p})^{\mu_{tT}} \\ &\quad \times \mathit{stratify}(ST^{sC}_{a_e a_p})^{\mu_{tC}} \times \mathit{stratify}(ST^{sQ}_{a_e a_p})^{\mu_{tQ}} \end{aligned} \tag{3.15}$$

where a_p^C and a_p^Q are a_p's advertised cost and quality respectively, max_C is the maximum advertised cost of the agents being considered, μ_C and μ_Q are the weightings given to the advertised cost and quality, and μ_{tS}, μ_{tT}, μ_{tC}, μ_{tQ} are the weightings for the trust dimensions of success, timeliness, cost and quality respectively. The general trust is used if the situational trust is not available.

The calculation of the performance value, considering both direct trust and witness reputation is as follows:

$$PV(a_p) = (max_C + 1 - a_p^C)^{\mu_C} \times ({a_p}^Q)^{\mu_Q}$$

$$\begin{aligned}
&\times \mathit{stratify}(ST^{sS}_{a_e a_p})^{\mu_{tS}} \times \mathit{stratify}(ST^{sC}_{a_e a_p})^{\mu_{tC}} \\
&\times \mathit{stratify}(ST^{sT}_{a_e a_p})^{\mu_{tT}} \times \mathit{stratify}(ST^{sQ}_{a_e a_p})^{\mu_{tQ}} \\
&\times \mathit{stratify}(WR^{sS}_{a_e a_p})^{\mu_{rS}} \times \mathit{stratify}(WR^{sC}_{a_e a_p})^{\mu_{rC}} \\
&\times \mathit{stratify}(WR^{sT}_{a_e a_p})^{\mu_{rT}} \times \mathit{stratify}(WR^{sQ}_{a_e a_p})^{\mu_{rQ}} \qquad (3.16)
\end{aligned}$$

where $WR^{sd}_{a_e a_p}$ is the evaluator a_e's witness reputation for target a_p for service type s in the dimension d, and μ_{rS}, μ_{rC}, μ_{rT}, μ_{rQ} are the weightings for the witness reputation in the dimensions of success, timeliness, cost and quality respectively (note that the weights μ_i must sum to 1). The performance values are what an evaluator uses to select among a number of potential interaction agents. The highest performance value suggest the most trustworthy agent.

3.4 Recommender's Perspective

The previous sections have described our model from the point of view of an evaluator. The evaluator builds the reputation of a target agent in the same way, whether the recommendations provided are direct or indirect. It assesses the principal witness on its reliability and accuracy of providing recommendations, using recommendation trust, irrespective of the source. In future work, we will consider using two separate recommendation trust values for direct and indirect recommendations from the principal witness.

The principal witness is the agent from whom the evaluator requests information about a target and it is selected from the evaluator's trusted witnesss or providers. It first searches for any direct task interactions with the target in its interaction history. Past experience with the target is shared with the evaluator in the form of an interaction summary. If there are insufficient or no direct task interactions, the principal witness requests the opinion of its most trusted witness. In this version of our model, we consider one level of indirection as this reduces the possibility of inaccuracies. Future work will look into how to apply an efficient way of obtaining indirect opinions along a recommendation chain, whilst maintaining accuracy and relevance.

The secondary witness returns direct task interaction information with the target to the principal witness as an interaction summary. If it has had interactions about different task types, the secondary witness shares its overall agent recommendation about the target. If the principal witness has interacted with the target in a different task type as requested by the evaluator, it will return its own agent recommendation, rather than the indirect agent recommendation from the secondary witness. The principal witness does not update its recommendation trust in the secondary witness as it is only passing on the indirect opinion and there has been no effect on its own tasks.

We have described our trust and reputation model and seen how agents use this model to assess the trustworthiness of other agents. In the next section, we set out to evaluate our model to show how it performs compared to other agent assessment methods.

3.5 Evaluation of Trust and Reputation Model

In order to assess the performance of our trust and reputation model, we have built a simulation environment and conducted a number of experiments. In all the experiments, the evaluation is done from the point of view of an evaluator agent. The implementation was written in Java using the NetBeans IDE [1]. The aim of our evaluation is to compare the performance of the evaluator in different types of agent populations when it uses a number of interaction partner assessment approaches for selecting the most

[1] http://netbeans.org/

reliable provider agent. In our experiments, we will compare the performance of the evaluator when using our trust and reputation model and when it uses other agent assessment methods, such as using service characteristics, trust only and trust with direct recommendations only.

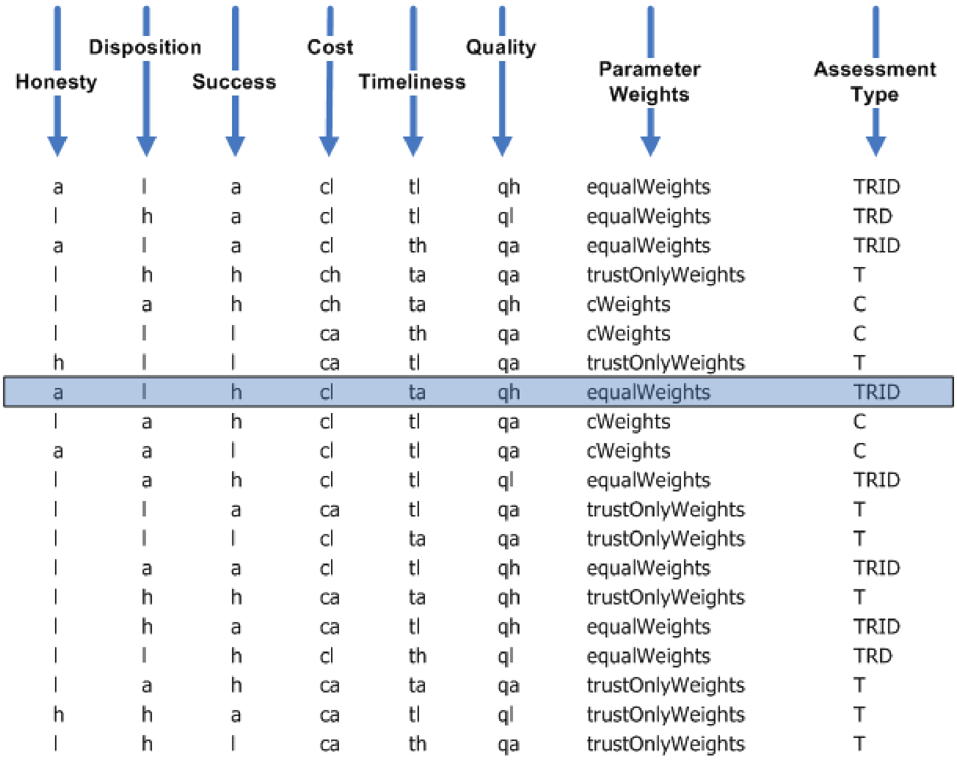
3.5.1 Experimental Setup

Firstly, we define the agent population parameters, which determine the characteristics of agents' behaviour. The range of population configuration parameters reflect the heterogeneity of the agent populations being represented. Secondly, we describe the population configurations we use in our experiments. Next, we specify the changes in agent behaviour in the population throughout the experiments. Agents may exhibit dynamic behaviours and we want to assess how our trust model copes with such changes in behaviour. Finally, we specify the metrics for the trust model evaluation and experimental results.

3.5.1.1 Agent Population Parameters

The agent population parameters describe the behavioural aspects of agents, in terms of their honesty, disposition, success and other service dimensions, as well as the weight of those parameters in their assessment of other agents. An example of an agent population file is shown in Figure 3.2. This generated agent population file is then used by our simulation program to build the agent objects and to start transactions. For each population configuration, 50 different populations are generated and we perform 5 runs per population. The population parameters are further detailed below.

1. Population size (*PopulationSize*) defines the total number of agents in the population. In our experiments, the population size ranges from 10 to 100. The range



Honesty	Disposition	Success	Cost	Timeliness	Quality	Parameter Weights	Assessment Type
a	l	a	cl	tl	qh	equalWeights	TRID
l	h	a	cl	tl	ql	equalWeights	TRD
a	l	a	cl	th	qa	equalWeights	TRID
l	h	h	ch	ta	qa	trustOnlyWeights	T
l	a	h	ch	ta	qh	cWeights	C
l	l	l	ca	th	qa	cWeights	C
h	l	l	ca	tl	qa	trustOnlyWeights	T
a	l	h	cl	ta	qh	equalWeights	TRID
l	a	h	cl	tl	qa	cWeights	C
a	a	l	cl	tl	qa	cWeights	C
l	a	h	cl	tl	ql	equalWeights	TRID
l	l	a	ca	tl	qa	trustOnlyWeights	T
l	l	l	cl	ta	qa	trustOnlyWeights	T
l	a	a	cl	tl	qh	equalWeights	TRID
l	h	h	ca	ta	qh	trustOnlyWeights	T
l	h	a	ca	tl	qh	equalWeights	TRID
l	l	h	cl	th	ql	equalWeights	TRD
l	a	h	ca	ta	qa	trustOnlyWeights	T
h	h	a	ca	tl	ql	trustOnlyWeights	T
l	h	l	ca	th	qa	trustOnlyWeights	T

Figure 3.2: Sample Agent Population File.

and incremental steps will be specified for each set of experiments later in the chapter.

2. Behaviour configuration describes a set of parameters that characterise how agents behave when interacting with other agents and their service provision. These parameters are grouped in three categories (high, average, low), represented as Xh, Xa, Xl in Figure 3.2, where X denotes the service characteristic. Agents with a high service quality can have different quality rates within that category. For example, on a scale from 1 to 50, high quality range *qh* is (41,50), while low quality *ql* has range (1,10) and average quality *qa* has range (11,40). We considered the following parameters.

- Honesty determines how accurately an agent executes a service (in terms of the service dimensions) compared to the advertised service characteristics.
- Disposition is the general willingness an agent has to trust another agent, especially when it has not interacted with it before.
- Success measures the ability of an agent to complete a task.
- Cost of a service.
- Timeliness of execution or delivery of the service.
- Quality of the service.

3. The assessment approach indicates how agents assess the reliability of other agents in their environment.

   (a) The possible assessment types are:

   - Service characteristics, illustrated by Cost (C) in our experiments.
   - Trust only (T).
   - Trust with Direct Recommendations (TRD).
   - Our trust assessment method using Trust with Direct and Indirect Recommendations, as well as the recency of interactions and the relevance of recommendations (TRID).

   (b) Parameter weights are the importance of the parameters for the evaluating agent when assessing other agents. The weights used in our experiments consider service characteristics only, i.e. cost in our experiments (*cWeights*), trust-related parameters (*trustOnlyWeight*) and equal weights for service dimensions, trust and reputation (*equalWeights*).

Figure 3.2 illustrates an example agent population file, which specifies the population parameters. Each row corresponds to the specification for one agent in the population. This example file is for a population of size 20, where the evaluator agent (first row) is using our trust and reputation model for assessing other agents in the population. Let us consider the shaded row in the diagram. The agent in that row has average honesty, low disposition to trust, high success, low cost, average timeliness and high quality aspects of service. It assesses other agents using the TRID approach and considers the service, trust and reputation parameters equally.

In our implementation, there are several population parameters that we have kept constant as we wanted to evaluate the effects of the parameters presented above. Among those parameters that we keep constant is the trust threshold used by the trust models, namely the value below which an evaluator considers that another agent is too untrustworthy to interact with. We have also assumed that the agent behaviours and the service characteristics offered apply to all the services offered by an agent.

3.5.1.2 Experimental Population Configurations

We consider four different agent population configurations to highlight the performance of our trust and reputation model in different circumstances. These configurations are indicative examples of the characteristics of the agent population, as the sample space is too large to be considered in its entirety. For each population, we specify the proportion of agents with each behaviour category (high, average, low). As an example, consider a population composed of 80% high honesty agents and 100% high cost agents. With 80% of agents having high honesty, this population has a very high proportion of agents that deliver their services according to expectations. The remaining 20% of agents have either low or average honesty. Moreover, all the agents in the population have services with high cost. In this example, two service characteristics have been specified, honesty and cost. For the remaining service characteristics, the agents have behaviours which are randomly selected from the three behaviour categories (high, average or low).

1. **Population 1** consists of 70% low honesty agents, 50% low cost agents and 50% low timeliness agents.
2. **Population 2** is made up of 70% high honesty agents, 70% high success agents and 40% high cost agents.
3. **Population 3** is composed of agents with each of the honesty, disposition, success, cost, timeliness and quality parameters in the proportions of 20% low, 60% average and 20% high categories.
4. **Population 4** consists of 70% high success agents and 100% average cost agents.

For Population 1, while 70% are agents with low honesty, the remaining 30% of agents are of either high or average honesty. The sum of the proportions of agents in the three categories amount to 100%. In our experiments, each population configuration is used to generate 50 different populations and each population will be run 5 times for each of the four assessment types C, T, TRD and TRID. The agent population file shown in Figure 3.2 is from Population 1. As 70% of the population have low honesty, there are 14 out of the 20 agents with low honesty. Of the remaining 6 agents, 4 have average honesty while 2 agents have high honesty. This follows from the random allocation of the remaining 6 agents to the remaining categories of reliability (average and high). Similar calculations of population proportions have been applied for the other parameters.

3.5.1.3 Experimental Agent Behaviour Profiles

Agents in a population may have dynamic behaviour, whereby they can have different honesty and service characteristics at different times. Changes in agent behaviour are reflected in changes in the corresponding properties in the agent population parameters. We consider four types of agent behaviour profiles in our experiments.

1. **Behaviour Profile A** involves all agents with static behaviours. Agents do not change their behaviour during the simulation period.

2. **Behaviour Profile B** includes all agents with high honesty, which change to low honesty half way into the simulation period.

3. **Behaviour Profile C** involves all agents with high success changing to low success a quarter of the way into the simulation period.

4. **Behaviour Profile D** causes all agents with high honesty and high success to change to low honesty and low success a quarter of the way into the simulation period.

The runs for each generated population are repeated for each of the 4 behaviour profiles, such that we can compare the results when the evaluator uses different types of assessments to decide on agent selection for future interactions. Therefore, for each of the behaviour profiles (A to D), the experiment is run 1000 times. The experiments comparing these different behaviour profiles have been performed for population configurations Population 1, Population 2, Population 3 and Population 4. The population sizes used in each run are 10, 30, and 50.

3.5.1.4 Metrics for Trust Model Evaluation

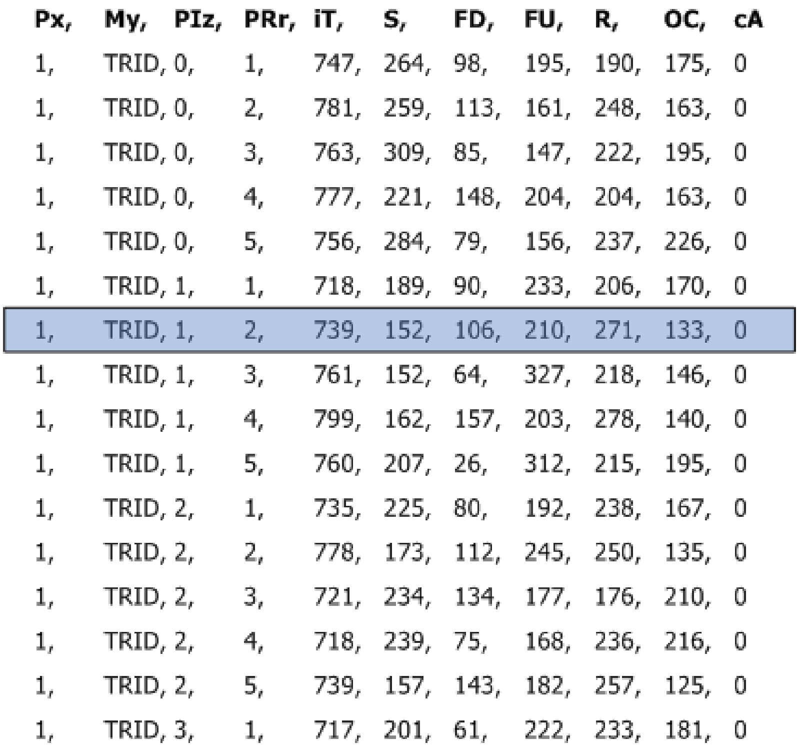
The experiments aim to assess the performance of the evaluator in selecting agents, when faced with heterogeneous agents of different capabilities as well as changes to agent behaviours. Performance assessment is measured by failure ratio and the ratio of services that were over the advertised price. A number of information elements about the population, the services requested and executed are gathered as explained below.

1. Population configuration number (*Px*), with values in the set 1, 2, 3, 4 due to the four types of agent populations we are considering in our evaluation.
2. Model type (*My*), the type of assessment used by the agent C, T, TRD, TRID.
3. Population index (*PIz*), the index of the population generated for a population configuration 0-49.
4. Population run number (*PRr*), which specifies the run for a particular population 1-5.
5. Number of tasks requested (*iT*) by the evaluator during the simulation period.
6. Number of successfully completed tasks (*S*) among the tasks requested by the evaluator.
7. Number of tasks failed due to providers declining (*FD*). The evaluator considers a task has failed in this way when providers have declined to execute the task three times.
8. Number of failed tasks due to providers not successfully completing tasks (*FU*) among the tasks accepted to be executed by providers.

9. Number of remaining uncompleted tasks (R) after the end of the allocated simulation period.

10. Number of tasks whose cost was higher than advertised cost (OC).

11. Number of agents whose behaviour changed in the simulation (cA), this is based on Section 3.5.1.3 and is from a system perspective, not from that of the evaluator.



```
Px,  My,   PIz, PRr, iT,  S,    FD,   FU,   R,    OC,   cA
1,   TRID, 0,   1,   747, 264,  98,   195,  190,  175,  0
1,   TRID, 0,   2,   781, 259,  113,  161,  248,  163,  0
1,   TRID, 0,   3,   763, 309,  85,   147,  222,  195,  0
1,   TRID, 0,   4,   777, 221,  148,  204,  204,  163,  0
1,   TRID, 0,   5,   756, 284,  79,   156,  237,  226,  0
1,   TRID, 1,   1,   718, 189,  90,   233,  206,  170,  0
1,   TRID, 1,   2,   739, 152,  106,  210,  271,  133,  0
1,   TRID, 1,   3,   761, 152,  64,   327,  218,  146,  0
1,   TRID, 1,   4,   799, 162,  157,  203,  278,  140,  0
1,   TRID, 1,   5,   760, 207,  26,   312,  215,  195,  0
1,   TRID, 2,   1,   735, 225,  80,   192,  238,  167,  0
1,   TRID, 2,   2,   778, 173,  112,  245,  250,  135,  0
1,   TRID, 2,   3,   721, 234,  134,  177,  176,  210,  0
1,   TRID, 2,   4,   718, 239,  75,   168,  236,  216,  0
1,   TRID, 2,   5,   739, 157,  143,  182,  257,  125,  0
1,   TRID, 3,   1,   717, 201,  61,   222,  233,  181,  0
```

Figure 3.3: Extract from Results File for Population 1, Behaviour Profile A, using assessment model TRID.

An example extract of a results file is shown in Figure 3.3. It is a comma-separated file for Population 1 and Behaviour Profile A, when the evaluator is using the TRID assessment model. Considering the shaded row, this is the simulated interaction results for the evaluator, based on population configuration Population 1 where the evaluator is using the TRID assessment model. This row entry concerns the second population generated (of the 50) and is on run number 2. During the interaction period, the evaluator requested 739 service tasks to be executed on its behalf. Among those tasks, 152 were successfully completed, 106 failed due to being declined for execution, 210 failed due to the providers unsuccessfully completing them and 271 remained uncompleted at the end of the period. Among the tasks that completed, 133 of them resulted in an overspend. As this example concerns Behaviour Profile A, no agent changed its behaviour during the interaction period.

Following the gathering of data, we analyse the performance of the evaluator using the different assessment models under various conditions (different population configuration and behaviour change). We calculate two further values for analysis.

- The failure ratio due to unsuccessful completion of tasks by the providers (*FU_ratio*). This is given by:

$$FU\_ratio = \frac{FU}{iT - R} \tag{3.17}$$

  where $iT - R$ is the number of tasks that were completed.

- The overspend ratio (*OC_ratio*) of completed tasks that incurred an overspend (the actual cost being higher than the advertised cost). This is given by the equation below.

$$OC\_ratio = \frac{OC}{iT - R} \tag{3.18}$$

The total number of tasks requested to be executed consists of the following elements:

$$iT = S + FD + FU + R \tag{3.19}$$

where S is the number of successfully completed tasks, FD is the number of tasks that have failed due to having found no providers, FU is the number of failed tasks due to unsuccessful completion by the providers, and R is the number of remaining tasks not completed at the end of the simulation period. We focus on FU as it is a better indication of whether agents are able to avoid interacting with untrustworthy agents, as we assume that the loss of utility from not interacting is less than that of a failed interaction.

3.5.2 Experimental Results

The evaluation of our trust and reputation model involves comparing the performance of the evaluator when using different agent assessment models under the same environmental conditions. We run experiments for each of the population dynamics, Behaviour Profile A, Behaviour Profile B, Behaviour Profile C and Behaviour Profile D, described in Section 3.5.1.3. For each behaviour profile, we run four experiments according to the assessment model used by the evaluator, namely service dimension (C), trust (T), trust with direct recommendations (TRD), or trust with direct and indirect recommendations (TRID). This is repeated for each of the four population configurations Population 1, Population 2, Population 3 and Population 4 (described in Section 3.5.1.2). Since we are comparing the different assessment models, we have grouped the results per behaviour profile (described in Section 3.5.1.3). Each of the four behaviours consists of 3000 result entries per model type.

We present two sets of results. Firstly, the ratio of failed tasks is compared for each of the four assessment models (C, T, TRD and TRID) in all four behaviour profiles (A, B, C and D). A larger ratio of failed tasks indicates that the assessment model copes less well with changes in agent behaviour that are caused by unsuccessful service provision. Therefore, the assessment models with a smaller failed task ratio are better at coping with changes in agent behaviour throughout the interaction period.

Secondly, the overspend ratio is compared for the assessment models in all four behaviour profiles. Each service has an expected cost, and dishonest agents or poor performing agents can cause the actual cost to be higher than expected. Agents that use trust and reputation as well as service characteristics for assessing partner agents should be better able to manage the uncertainty of interactions, compared to agents that only use service characteristic, such as cost. An agent that has a large overspend ratio cannot cope appropriately with changes in agent behaviour, when it does not concern only the service characteristic.

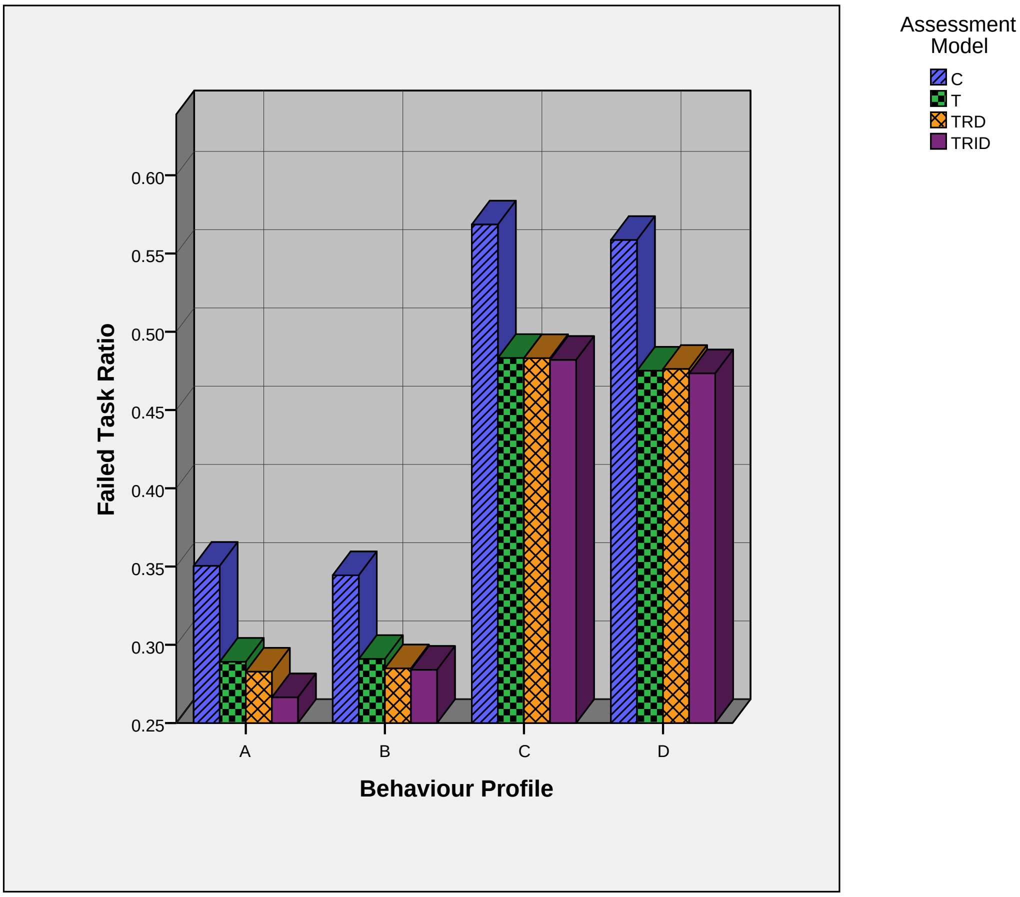


Figure 3.4: Mean Failed Task Ratios Per Behaviour Profile for Each Assessment Model.

Figure 3.4 shows the evaluator's mean failed task ratio for each of the four behaviour profiles when using each of the four assessment models. We can observe that the ratio of failed tasks is significantly larger for the Cost assessment model (C), compared to the three trust models. The differences in ratio for the three trust models are small. However, we can observe that our TRID model (unshaded fourth bar from the left in each cluster) performs slightly better than the other two trust models, and that the failed task ratio is smaller, especially for behaviours profiles A and B.

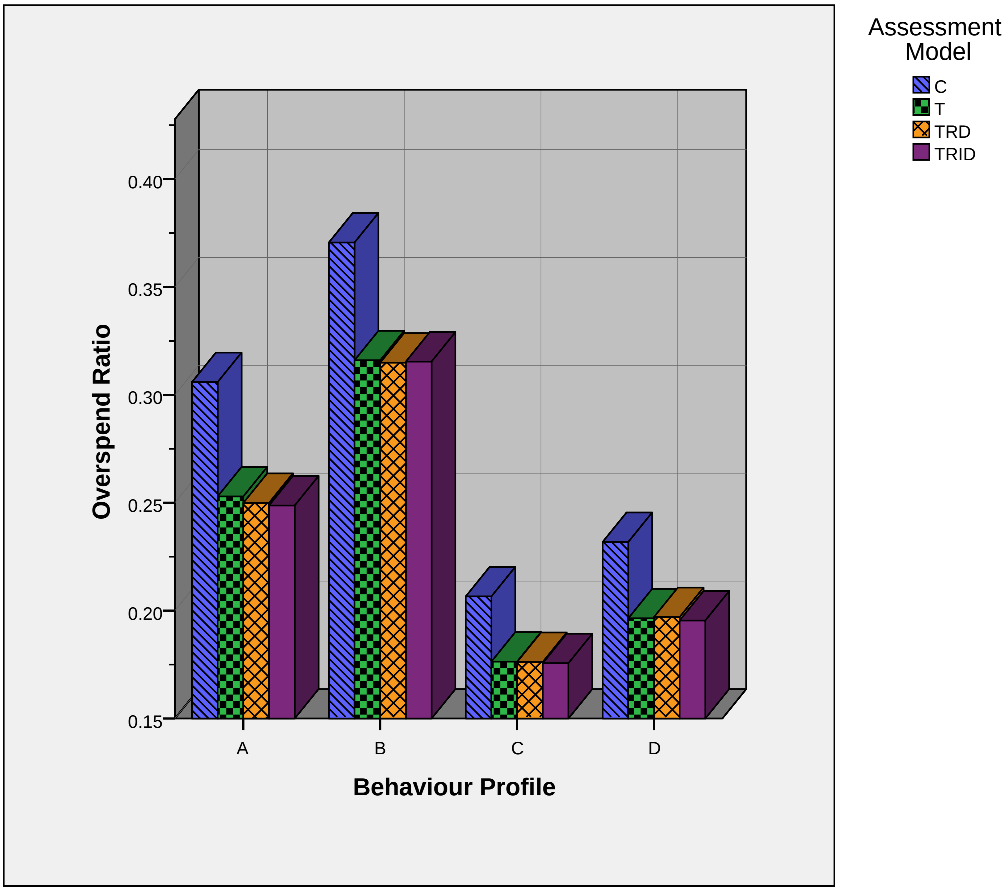


Figure 3.5: Mean Overspend Ratios Per Behaviour Profile for Each Assessment Model.

Figure 3.5 shows the evaluator's mean overspend ratio for each of the four behaviour profiles when using each of the four assessment models. Again, we can observe that the evaluator incurs the most overspend when it relies on the cost dimension of the service to select partner agents (assessment model C). There are only small differences in overspend ratio among the three trust models, but in behaviour profiles A, C and D using the TRID model results in a smaller overspend ratio, suggesting that the TRID assessment model can better cope with changes in the agent environment, including the decrease in honesty of a number of agents.

3.5.3 Approach to Statistical Significance Testing

The main aim of the evaluation is to assess the differences when using different assessment models for agent selection. From the results in Section 3.5.2, we have seen that there are some differences in performance of the different assessment models. In order to assess whether we can generalise these results, we test them for statistical significance. We apply paired *t*-tests on pairs of assessment models we want to analyse. Paired *t*-tests are appropriate since we adhere to the following assumptions.

- The value pairs are independent and the experiments have been set up to run with the same configurations, with only the assessment model used by the evaluator changing. The two sets of experiments have also been run separately.

- The sample data is drawn from a normal population of agents according to the selected population configurations. The *t*-test would also perform well if this assumption is violated [44].

The test procedure involves the analysis of the differences between the failed task ratio (*FU_ratio*) when two different assessment models are used. Similarly, we also analyse the differences between the overspend ratio (*OC_ratio*) between two assessment models as used by the evaluator. The mean of the differences should be 0 if there is no difference between the respective ratios.

Let (X_{11}, X_{21}), (X_{12}, X_{22}), ..., (X_{1n}, X_{2n}) be a set of n pairs where we assume that the mean and variance of the population X_1 are μ_1 and σ_1^2, and the mean and and variance of the population X_2 are μ_2 and σ_2^2. The difference between each pair of ratios is defined as $D_j = X_{1j} - X_{2j}$, where $j = 1, 2, \ldots, n$. Hypotheses for the failed task ratio F and overspend ratio O take the following forms.

$H_{F\alpha z}$: $\mu_D = \mu_{F_1} - \mu_{F_2} = 0$

$H_{O\alpha z}$: $\mu_D = \mu_{O_1} - \mu_{O_2} = 0$

where α is the hypothesis identifier and z is 0 or 1 for null or alternative hypothesis. F and O are the failed task ratio and the overspend ratio respectively. Each population X_{AM_1} represents the set of failed task ratios for assessment model AM_1, and population X_{AM_2} represents the set failed task ratios for assessment model AM_2. Our list of null hypotheses for precision are as follows.

H_{Fa0}: $\mu_D = \mu_C - \mu_{TRID} = 0$

H_{Fb0}: $\mu_D = \mu_T - \mu_{TRID} = 0$

H_{Fc0}: $\mu_D = \mu_{TRD} - \mu_{TRID} = 0$

H_{Fd0}: $\mu_D = \mu_C - \mu_T = 0$

H_{Fe0}: $\mu_D = \mu_C - \mu_{TRD} = 0$

H_{Ff0}: $\mu_D = \mu_T - \mu_{TRD} = 0$

Similarly, our null hypotheses for overspend ratios are listed below.

H_{Oa0}: $\mu_D = \mu_C - \mu_{TRID} = 0$

H_{Ob0}: $\mu_D = \mu_T - \mu_{TRID} = 0$

H_{Oc0}: $\mu_D = \mu_{TRD} - \mu_{TRID} = 0$

H_{Od0}: $\mu_D = \mu_C - \mu_T = 0$

H_{Oe0}: $\mu_D = \mu_C - \mu_{TRD} = 0$

H_{Of0}: $\mu_D = \mu_T - \mu_{TRD} = 0$

The alternative hypotheses for failed task ratio and overspend ratio are now described and these indicate that there is a difference between the ratios when different assessment models are used. This is the set of hypotheses that we will be investigating. The alternative hypotheses for the failed task ratios are as follows:

H_{Fa1}: $\mu_D = \mu_C - \mu_{TRID} \neq 0$

H_{Fb1}: $\mu_D = \mu_T - \mu_{TRID} \neq 0$

H_{Fc1}: $\mu_D = \mu_{TRD} - \mu_{TRID} \neq 0$

H_{Fd1}: $\mu_D = \mu_C - \mu_T \neq 0$

H_{Fe1}: $\mu_D = \mu_C - \mu_{TRD} \neq 0$

H_{Ff1}: $\mu_D = \mu_T - \mu_{TRD} \neq 0$

Similarly, the alternative hypotheses for the overspend ratios are:

H_{Oa1}: $\mu_D = \mu_C - \mu_{TRID} \neq 0$

H_{Ob1}: $\mu_D = \mu_T - \mu_{TRID} \neq 0$

H_{Oc1}: $\mu_D = \mu_{TRD} - \mu_{TRID} \neq 0$

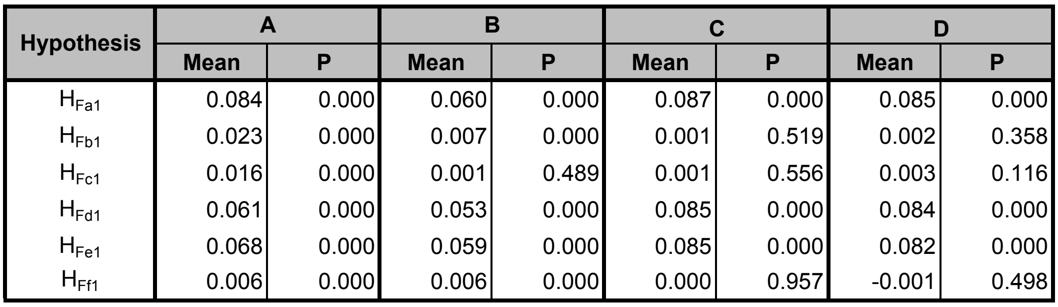
H_{Od1}: $\mu_D = \mu_C - \mu_T \neq 0$

H_{Oe1}: $\mu_D = \mu_C - \mu_{TRD} \neq 0$

H_{Of1}: $\mu_D = \mu_T - \mu_{TRD} \neq 0$

3.5.4 Discussion of Statistical Significance Tests

The statistical software package PASW Statistics (SPSS Statistics) 18[2] has been used to calculate the paired *t*-tests. Tables 3.1 and 3.2 show the summary of hypotheses results for the four behaviour profiles A, B, C, and D. The full details of the results are presented in Appendix B.1.



Hypothesis	A		B		C		D	
	Mean	P	Mean	P	Mean	P	Mean	P
H_{Fa1}	0.084	0.000	0.060	0.000	0.087	0.000	0.085	0.000
H_{Fb1}	0.023	0.000	0.007	0.000	0.001	0.519	0.002	0.358
H_{Fc1}	0.016	0.000	0.001	0.489	0.001	0.556	0.003	0.116
H_{Fd1}	0.061	0.000	0.053	0.000	0.085	0.000	0.084	0.000
H_{Fe1}	0.068	0.000	0.059	0.000	0.085	0.000	0.082	0.000
H_{Ff1}	0.006	0.000	0.006	0.000	0.000	0.957	-0.001	0.498

Table 3.1: Summary of Hypotheses Results for Failed Task Ratio for 4 Behaviour Profiles

Table 3.1 shows that in most cases, the assessment models that use a richer set of data perform better. This is shown by the positive mean differences in the failed task ratio. Taking into account the four assessment models (C, T, TRD, TRID), these are ordered in increasing richness of data used for agent assessment. For the hypotheses where $P < 0.05$, we can reject the null hypotheses and can conclude that

[2] `http://www.spss.com/software/statistics/`

there is a significant difference between the failure ratios between the corresponding assessment models. Some of the mean differences are not statistically significant, but we can nevertheless observe the performance trend. We note that for the hypothesis H_{Fc1} there is no significant difference between the failed task ratio when using TRD and our model TRID, except for Behaviour Profile A, which represents a static environment. Although not statistically significant, TRID results in fewer failed tasks than TRD.

Hypothesis	**A**		**B**		**C**		**D**	
	Mean	**P**	**Mean**	**P**	**Mean**	**P**	**Mean**	**P**
H_{Oa1}	0.057	0.000	0.055	0.000	0.031	0.000	0.036	0.000
H_{Ob1}	0.004	0.001	0.001	0.663	0.001	0.390	0.001	0.269
H_{Oc1}	0.001	0.324	0.000	0.749	0.001	0.509	0.002	0.090
H_{Od1}	0.053	0.000	0.055	0.000	0.030	0.000	0.035	0.000
H_{Oe1}	0.056	0.000	0.056	0.000	0.030	0.000	0.035	0.000
H_{Of1}	0.003	0.020	0.001	0.451	0.000	0.850	-0.001	0.539

Table 3.2: Summary of Hypotheses Results for Overspend Ratio for 4 Behaviour Profiles

The next set of statistical tests concerns the difference in the ratio of tasks that resulted in overspending when the evaluator uses different assessment models. Overspending occurs when the cost of a service is higher than what the provider initially advertised it for. It relates to the honesty of the provider in giving the correct information about its services. From Table 3.2, we can observe that the differences in overspend ratios are significant, mainly for comparisons involving the Cost assessment model. For all the statistically significant differences in ratio, the evaluator has overspent for a larger proportion of tasks when it used the Cost model compared to using either of the three trust models. The alternative hypothesis H_{Oc1} states that there is no difference in the mean overspend ratio between using the TRD and the TRID models. As $P \geq 0.05$ in all four cases, we cannot reject the null hypothesis and cannot conclude that there is a significant difference between the two ratios. Despite the differences not being significant, we observe that the mean differences are positive, suggesting that the failed task ratio when using TRID is less than when using the TRD model.

3.5.5 Summary of Results

The results in Section 3.5.2 show that agents using assessment models based on trust perform better than those that only consider service characteristics (assessment model C compared to models T, TRD and TRID). Agents using our assessment model TRID had fewer failed tasks compared to those using the other two trust assessment models. We tested the differences in performance between pairs of models for statistical significance. The test results show that with our TRID model, agents have fewer failed tasks compared to models C and T, and these results are significant. Although the difference between models TRID and TRD were not significant, agents using TRID tended to have fewer failed tasks than those using TRD.

In terms of the overspend ratio, from the graph in Figure 3.5, agents using the TRID assessment model perform at least as well as agents using trust and reputation in their assessment. The statistical test results show that in the large majority of tests conducted, TRID enables agents to manage their costs better than the other assessment models. This trend is also reflected for agents that use trust and reputation information in increasing richness. Despite only part of the result set being statistically significant, the differences between models is mainly positive, indicating that agents perform better when they use a rich set of information about interactions and recommendations.

In Section 2.9, we summarised the limitations of certain existing trust and reputation models (namely MDT-R, Ntropi and ReGreT) with respect to the trust model characteristics that we believe they should possess. Our TRID model is designed to use a wide range of trust information sources, including direct interactions and direct and indirect recommendations, along with information at the level of service characteristics to ensure availability and quality of the information used for reasoning. Existing approaches do not include all of these features. For example, MDT-R [36] does not use indirect recommendations to reason about the trustworthiness of agents. Similarly, Ntropi [1] does not enable agents to assess agent trustworthiness in as many service characteristics as they deem necessary. In dynamic environments, agents tend to change behaviour rapidly and the freshness of interaction information is important for an evaluator to rapidly identify and respond to such changes. This is achieved by using recency as a characteristic in our TRID trust model, while other models, such as Ntropi, do not include this feature. Recommendations are an important source of information for trust assessment. However, not only should witnesses be known to the evaluator, but their opinions should be relevant, such that only reliable witnesses are approached for useful recommendations. Again, other models, such as ReGreT [104], do not use this approach to witness selection. These benefits ensure that our model can reliably support agents in assessing the trustworthiness of agents, as well as enabling the gathering of a rich set of interaction and recommendation information for further reasoning.

3.6 Discussion and Conclusions

3.6.1 Related Work

Using trust together with reputation taps into various sources of trust information to assess agents' trustworthiness. Our model uses both sources of information, as do existing models, including ReGreT [106], FIRE [46], Ntropi [1], and MDT-R [36]. The trust information from direct interactions is the most reliable and least prone to subjectivity, as it it the evaluating agent's own experience. In certain circumstances when direct interactions are few or outdated, the evaluator supplements direct trust information with reputation information obtained from third parties. This allows the evaluator to still make an accurate assessment of the target agent's trustworthiness.

The multiple dimensions of trust and reputation based on MDT-R [36] aim to further improve the accuracy of trust assessment and reduce the subjectivity of trust information via recommendations. Separating trust and reputation information into the important service characteristics allows decisions to be made based on the priorities of the evaluator.

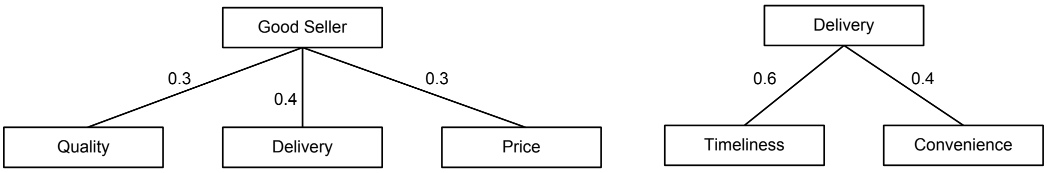


Figure 3.6: Ontological Structure in ReGreT.

In our model, agents have weights associated with the relevance of each service dimension, trust and reputation in the computation of the final performance value that is used to compare several potential interaction agents. ReGreT [104, 107] also assumes the use of weights by agents when considering the various behaviours in combining the different sources of trust and reputation in their ontological dimension. Figure 3.6 illustrates an ontological structure of the reputation of a good seller, as used in ReGreT. The reputation of a good seller is related to the reputation of its delivery, the price and quality of its product. As illustrated in the diagram, the delivery aspect can be further broken down into the aspects of timeliness and convenience. In ReGreT, the calculation of the reputation of each aspect involves calculating the reputation of the related aspects which can be in subgraphs [106]. Although the model handles complex behaviours of agents, changes in the weights in any of the subgraphs would involve a recalculation of the reputation in the related aspects, in order to obtain the highest reputation value that reflects the agent's current behaviour.

The mechanism used to apply different weights to witnesses, as used by Yu and Singh [140], is similar to the use of recommendation trust in our model. In both cases, agents are able to detect those witnesses which are not giving accurate ratings. However, neither is able to detect whether the differing ratings are a result of errors, collusion, or other type of deception. Our model is also different from Yu and Singh in that we allow indirect recommendations.

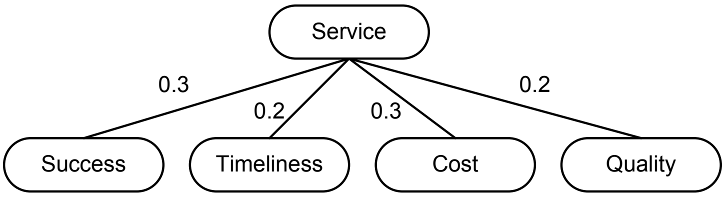


Figure 3.7: Model Overview From Provider's Perspective.

In our approach, agents agree on a fixed set of dimensions that characterise the services in the domain. For example, as illustrated in Figure 3.7, a service can be characterised in four dimensions, each of which has a weight associated with it to represent its importance. Even if the agents update the weights of the different dimensions to reflect their preferences, this does not affect how the trust and reputation values are calculated. Furthermore, the agreed set of dimensions makes the sharing of information more flexible among agents as all agents use dimensions within the set. The values for each dimension are still subjective, but the sharing of information about the aspects of a service is easier, as compared to ReGreT, where the ontology used can vary among agents and for different aspects that they represent. The different way of expressing these aspects in ReGreT makes the translation of the meanings among different agents more difficult and is more prone to the subjectivity problem.

3.6.2 Conclusions

In this chapter, we have presented our model of trust and reputation, which is has four main characteristics. Firstly, it is based on a number of trust information sources, including direct interactions, as well as direct and indirect recommendations. Secondly, trust assessment is done at a service level, enabling services to be individually assessed as well as at the level of their separate characteristics. Thirdly, the model considers recency for an accurate trust evaluation. Finally, the relevance of recommendations is taken into account for assessing recommendations. These trust model characteristics have been identified from a review of trust models in the literature and having analysed their contributions and limitations.

Combining these sources of trust information aims to ensure that the evaluator can more accurately assess the trustworthiness of a potential interaction partner in different situations. Insufficient direct interactions and direct recommendations can be complemented by including indirect recommendations from trusted agents. Our approach also represents trust and reputation in multiple dimensions to maintain the richness of the original information so as to make more accurate decisions based on the various service characteristics and agent behaviour. From our review of trust models in Section 2.5, ReGreT is the only model to use a number of trust sources and considers the service characteristics in its ontological dimension of reputation. However, ReGreT does not look at trust and the other dimensions of reputation (individual and social) as pertinent to individual service characteristics that are important to the evaluator agent. Some existing trust models consider either recency or relevance, but not both. Our model therefore uniquely brings together these four key characteristics to improve trust assessment and to allow for a richer set of data for future analysis.

In our evaluations (Section 3.5), we have shown that our trust and reputation model (TRID) has a good performance compared to the other trust models. From the results, we observed that although the difference between failed task ratios using our model and a comparison model was not always statistically significant, the sign in the difference value showed that our model performed slightly better than the TRD model, for instance. The results also showed the trend of the failure ratio decreasing as the assessment model used richer data for its agent assessment. One important issue to consider in future work is how agents can dynamically adapt to changes in agent behaviour to maintain their performance rates. In the present model, recommendation trust is used to assess the trustworthiness of a witness in giving recommendations for interactions similar to those of the evaluator. One limitation of using only one recommendation trust value is the non-differentiation of the trustworthiness of direct and indirect witnesses. The recommendation trust could be further subdivided into direct and indirect recommendation trust if it is important for the evaluator to specifically choose among witnesses based on the type of recommendation they give. Additional open issues that would need to be considered are the optimal length of recommendation chain to use and the confidence in the indirect recommendation at different points in the chain.

Chapter 4

Agent Network Model Building

4.1 Introduction

The main motivation for individual agents to build a model of their agent network is to obtain a better picture of their environment, through their own interactions and from the recommendations of others. The aim is to use the agent network model to support their future decision making. Past interaction histories and recommendations throughout the interaction period are made up of different types of information which, if properly extracted and analysed, can be valuable for agents. Different types of information can be extracted from the agent network model, regarding the relationships between agents. The information in the form of agent graphs is also useful for system architects and human analysts who are responsible for tuning the agent system parameters to maintain effectiveness. Agent systems can be complex, with many nodes and many more edges among them. Visualisation tools to easily view the how agents are linked and the data shared are increasingly becoming necessary to support human users.

The aim of this work is to easily represent and view agent networks and the rich set of data that have been gathered with our trust and reputation model, discussed in Chapter 3. Human analysts can get a better understanding of how the agent network is organised and can subsequently extract other useful information that was previously unknown. The remainder of this chapter is organised as follows. The following subsections give an overview of the types of information discovery that can be performed on the agent graphs. Next, we present the different types of agent graphs, built and maintained to represent the agent network model. We then study a number of agent graphs that are built by agents and view the data they hold with the help of visualisation tools. Finally, we present an example of an evaluator uses interaction and recommendation information to build agent graphs.

4.1.1 Rediscovery of Service Providers

Agents are assumed to have access to a service directory that provides an up-to-date listing of service providers and the services they provide. When an evaluator agent wants to acquire a particular service, it usually looks up potential providers from the service directory. The selection of the most appropriate provider is based on past experience, and also recommendations. For example, an evaluator a_e may have had a number of recent unsuccessful interactions with a particular provider a_p. As a result, the evaluator's trust in a_p decreases and if the trust value goes lower than its trust threshold, a_e is unlikely to interact with a_p in the future. In the case where the evaluator has a long memory of past interactions, it might take longer to notice positive changes in a_p. With the help of an agent network model, a_e can keep track of how agents in its environment are interlinked despite not interacting with a_p as a result of its untrustworthiness. Recommendations about different services and agents may reveal that agents are interacting with a_p, possibly arising from a change in behaviour. The evaluator may then use this information to resume transactions with a_p sooner than it would otherwise have done.

4.1.2 Agent Interaction Discovery

Agents have direct knowledge about their own interactions. However, they can only learn about those of other agents in the environment through communication, in particular from recommendations. By sharing its opinion, a witness a_r is effectively giving details about its own interactions with the target agent a_t, or is passing on the experiences of an intermediate witness $a_{r\prime}$. From recommendations, both direct and indirect, the evaluator can learn how the agents in its neighbourhood are interlinked. For instance, an indirect recommendation from a_r to $a_{r\prime}$, which is then passed on to the evaluator, informs of a_r and $a_{r\prime}$ being possible interaction partners, due to the short recommendation chain.

4.1.3 Collusion Detection

Collusion among a set of agents can affect an evaluator agent in many ways. For example, it may be the victim of demotion by agents acting as witnesses, or it may be paying high prices as a result of price collusion. It is therefore important for an agent to be able to detect such collusive behaviour and act upon it to ensure successful future interactions. Information about how agents are interconnected, and on which level, such as service provision or recommendation links, are useful to enable the observing agent to analyse any particular interaction patterns or irregularities that lead to the detection of possible collusive activity.

4.2 Agent Graphs

The conceptual graphs (provider, witness, and combined graphs) are modelled and implemented as four types of graph, two in each of the provider and witness categories: service-oriented and agent-oriented provider graphs, as well as service-oriented and agent-oriented witness graphs. We now describe these graphs in more detail, as well as highlight some useful graph elements.

4.2.1 Service-oriented Provider Graph

A service-oriented provider graph is built and maintained by an agent to keep a record of the providers it interacts with for particular services. An evaluator holds one service-oriented provider graph for each of the service types it is concerned with. When the agent is new to the environment and has no previous interactions, it uses its initial trust to interact with other agents. The graph of agent providers initially depicts only the direct service providers of the evaluating agent. The nodes represent agents and the edges show the interactions between agents, including the strength of each link, in terms of experience (number of interactions). The direct providers of the evaluator form a star-shaped structure, with the evaluator in the centre. Figure 4.1 shows an example of such a graph. The evaluator keeps a record of interaction information for each type of service.

Each edge is directed from the evaluator to the provider, and the weight includes the number of positive and negative interactions between the two agents for a particular service, denoted as:

$$edge : a_e \rightarrow a_p, count^+, count^- \tag{4.1}$$

where there is an edge from each evaluator a_e to each of the providers a_p, the edge also has values relating to the interaction, $count^+$ being the count of positive interactions, while $count^-$ is the count of negative interactions. The algorithm for building this graph is presented in Algorithm 4.1, where the service-oriented provider graph is updated every time an interaction is completed between the evaluator and a service provider.

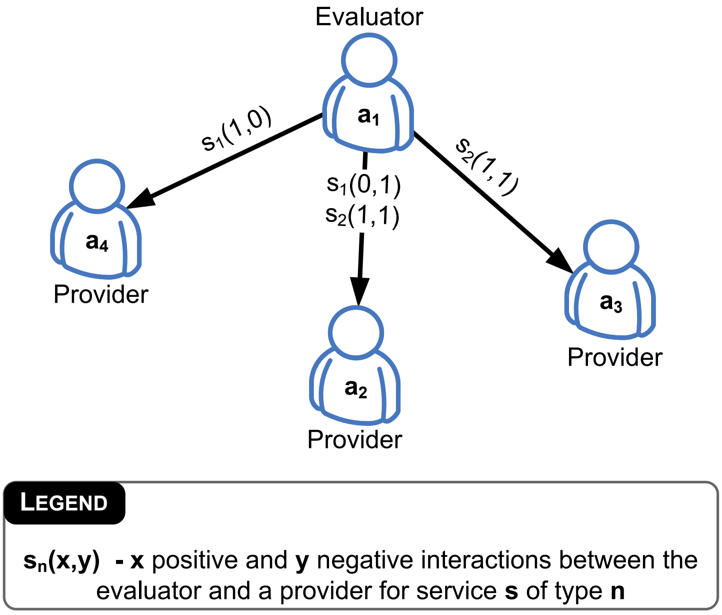

Evaluator
a_1
$s_1(1,0)$
$s_1(0,1)$
$s_2(1,1)$
$s_2(1,1)$
a_4
Provider
a_2
Provider
a_3
Provider
LEGEND
$s_n(x,y)$ - x positive and y negative interactions between the evaluator and a provider for service s of type n

Figure 4.1: Example of a Service-oriented Provider Graph.

Algorithm 4.1 updates the provider agent graph for a particular service. The list of providers the evaluator has already interacted with can be found in history H_{i_s} and for each new interaction, the evaluator checks against this list. If it is a new provider, a new edge is added, and in all cases, the count of interactions is incremented, depending on whether they were positive or negative.

4.2.2 Agent-oriented Provider Graph

An evaluator also builds an agent-oriented provider graph to record the overall interactions with different service providers, irrespective of the service type. The evaluator maintains one agent-oriented provider graph for each service provider. An example of such a graph is shown in Figure 4.2. Building and updating the graph is similar to that of the service-oriented provider graph, except that the agent-oriented graph is updated for each interaction between the evaluator and a provider, unlike the service-oriented

Algorithm 4.1 Updating the Service-oriented Provider Graph

```
for all interaction $i_s$ do
    if $a_p \notin H_{i_s}$ then
        add edge($a_p$, $a_e$)
    endif
    if $i_s$ = positive then
        increment $count^+$
    else
        increment $count^-$
    endif
endfor
```

provider graph which is updated only when there has been an interaction between the evaluator and a provider for a particular service type.

4.2.3 Service-oriented Witness Graph

The nodes represent the witnesses (direct or indirect) and the edges specify the links among agents, such as the type of recommendation, the number of opinions shared, and the number of accurate and inaccurate opinions. The witness graph is updated after the evaluator has requested and obtained the recommendation and has used it in its decision-making process to interact with the recommended agent.

The principal or direct witnesses give their own opinions to the evaluator or they are the first to be contacted by the evaluator if the recommendation is indirect. They are also structured around the evaluator in a star shape, with directed edges from the evaluator to the witness. The service-oriented witness graph concerns the opinions the evaluator stores about a particular service type. Indirect recommendations are represented as a chain linking the evaluator to the principal witness and another edge between the principal witness to the secondary witness. An example of such a graph is

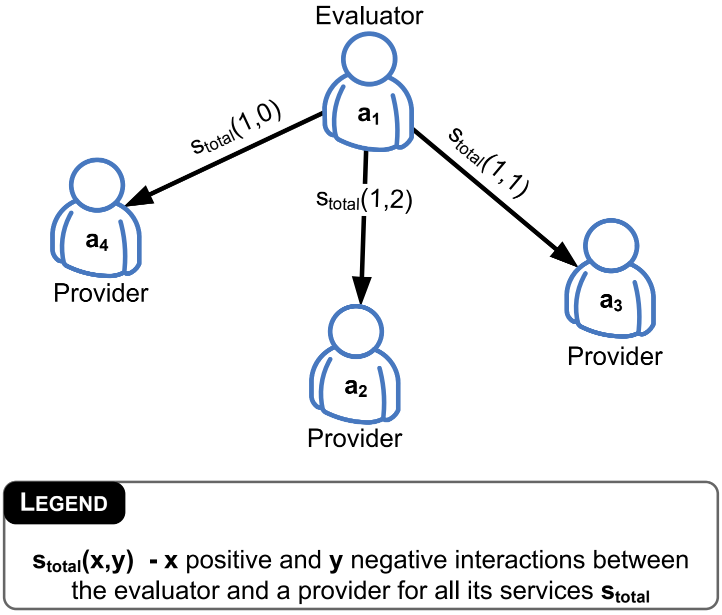

Evaluator
a1
$s_{total}(1,0)$
$s_{total}(1,2)$
$s_{total}(1,1)$
a4
Provider
a2
Provider
a3
Provider
LEGEND
s_{total}(x,y) - x positive and y negative interactions between the evaluator and a provider for all its services s_{total}

Figure 4.2: Example of an Agent-oriented Provider Graph.

depicted in Figure 4.3.

In this example, the evaluator a_1 received a number of direct recommendations from two direct witnesses a_2 and a_3. From a_2, it has received one recommendation, which was accurate when compared to the evaluator's subsequent interaction with the target. Witness a_3 has been giving recommendations about service type s_4, with one accurate, one inaccurate and one unused recommendation. The graph also shows that a_1 has received an indirect recommendation about service s_3 from secondary witness a_5, via principal witness a_2.

Each edge is directed from the recommendation requester to the witness, and the weight includes the number of accurate and inaccurate recommendations, as well as the number of unused recommendations.

$$edge : a_e \rightarrow a_r, count^{+}, count^{-}, count^{unused} \qquad (4.2)$$

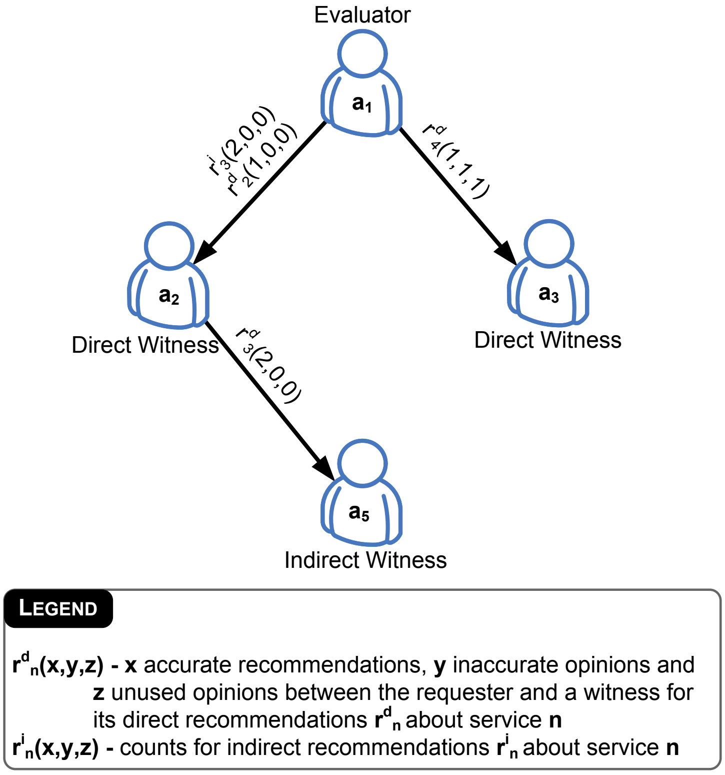

Evaluator
a1
r^i_3(2,0,0)
r^d_2(1,0,0)
r^d_4(1,1,1)
a2
Direct Witness
a3
Direct Witness
r^d_3(2,0,0)
a5
Indirect Witness
LEGEND
r^d_n(x,y,z) - x accurate recommendations, y inaccurate opinions and z unused opinions between the requester and a witness for its direct recommendations r^d_n about service n
r^i_n(x,y,z) - counts for indirect recommendations r^i_n about service n

Figure 4.3: Example of a Service-oriented Witness Graph.

The principal witness may request the opinion of secondary witnesses and thus form directed edges to the principal witness.

$$edge : a_{r'} \rightarrow a_r, count^{+}, count^{-}, count^{unused} \tag{4.3}$$

The pseudocode for building the graph of witnesses is presented in Algorithms 4.2 and 4.3. For a direct recommendation, an edge is created for every new witness and the recommendation count is incremented. If the recommendation is indirect, then the graph needs to be updated iteratively for every indirect recommendation. An edge needs to be created or updated from the further witness in the chain to the closer one. In Algorithm 4.3, $a_{r''}$ denotes the further witness in the chain.

Algorithm 4.2 updates the direct witness interactions in the witness graph. For a

Algorithm 4.2 Updating Direct Witnesses in the Witness Graph

```
for all direct recommendation $r^d$ do
  if $a_r \notin H_{i_r}$ then
    add edge($a_r, a_e$)
  endif
  if $r^d$ used do
    increment $count_{unused}$
  else
  // if recommendation is used for actual interaction
    if $r$ is close to actual $i_s$ then
      increment $count^+$ // accurate recommendation
    else
      increment $count^-$ // inaccurate recommendation
    endif
  endif
endfor
```

new witness, a new link is added from the evaluator to that witness. If the evaluator does not use the recommendation, it updates the $count_{unused}$ count, otherwise it updates the successful or failed interaction counts.

Algorithm 4.3 updates the indirect witness links in the witness graph. If the recommendation is unused, the $count_{unused}$ count is incremented. Otherwise, since it is an indirect recommendation, the direct links between the secondary witness and the target need to be updated.

4.2.4 Agent-oriented Witness Graph

An agent-oriented witness graph is maintained by an evaluator to record all the recommendations it receives about agents, irrespective of service type. An example is shown in Figure 4.4 and the graph building mirrors that for the service-oriented witness graph presented previously.

Algorithm 4.3 Updating Indirect Witnesses in the Witness Graph

```
for all indirect recommendation $r^i$ do
    if $a_{r\prime} \notin H_{i_r}$ then
        add edge($a_{r\prime}$, $a_e$)
    endif
    if $r^i$ unused do
        increment $count^{unused}$
    else
        repeat
        if $a_{r\prime\prime} \notin H_{i_r}$ then
            add edge($a_{r\prime\prime}$, $a_{r\prime}$)
        endif
        increment $count_{response}$
    until $r = r^d$ // direct recommendation
    endif
endfor
```

4.2.5 Combination and Extension of Graphs

From the provider and witness graphs, an agent can further extend and combine the graphs to help in identifying agent relationships not previously known. From the graph of witnesses, if an interaction results from the recommendation given via a chain of witnesses, the evaluator can identify the last witness in the chain to be the effective witness, who actually directly interacted with the target agent. As a result, the evaluator can extend its provider graph to add the target as a service provider for the ultimate witness.

From the witness graph, the evaluator can extend its provider graph. The additional providers are not its own direct providers, but the providers of its providers. This information can be valuable when analysing the relationships among the agents in its environment. Algorithm 4.4 shows the extension to the provider graph, by combining information from the witness graph, following an indirect recommendation. Along the

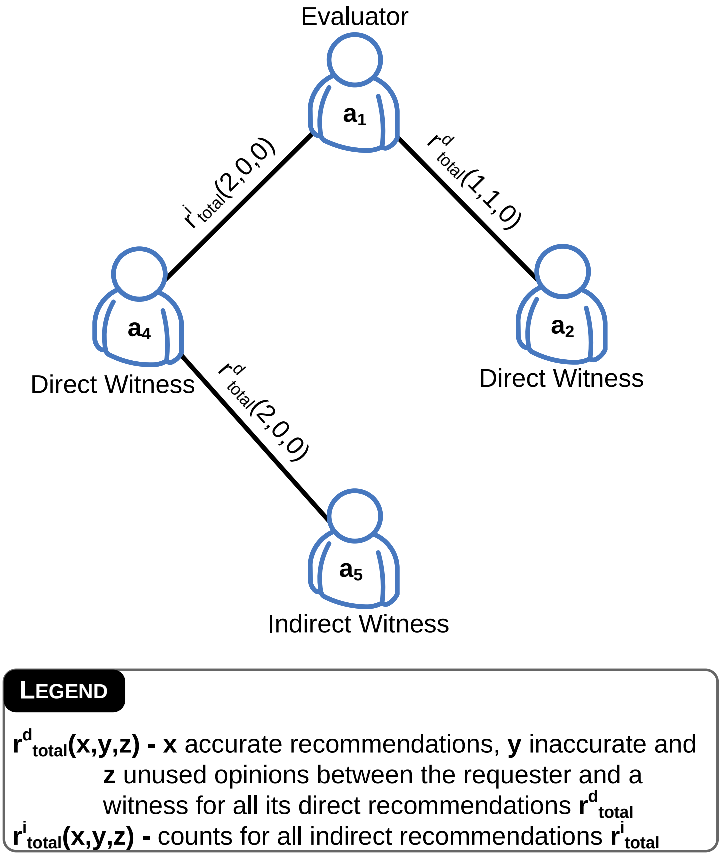

Evaluator
a1
$r^i_{total}(2,0,0)$
$r^d_{total}(1,1,0)$
a4
Direct Witness
a2
Direct Witness
$r^d_{total}(2,0,0)$
a5
Indirect Witness
LEGEND
r^d_{total}(x,y,z) - x accurate recommendations, y inaccurate and z unused opinions between the requester and a witness for all its direct recommendations r^d_{total}
r^i_{total}(x,y,z) - counts for all indirect recommendations r^i_{total}

Figure 4.4: Example of an Agent-oriented Witness Graph.

recommendation chain, the ultimate witness $a_{r''}$ provides a direct recommendation as it has interacted with the target agent a_t itself.

Algorithm 4.4 Extending the Provider Graph

```
add edge($a_{r''}$, $a_t$)
update $count_{interaction}$ // experience information provided by $a_{r''}$
```

For an agent interaction occurring as a partial result of indirect recommendations, an evaluator updates its provider and witness graphs as shown in Algorithm 4.5. The term r_μ is the currently processed recommendation. For a direct recommendation, an edge is created for each new witness and the recommendation count is incremented. Indirect recommendations are updated recursively, with edges created or updated from the further witness $a_{r''}$ in the chain to a closer one $a_{r'}$. Moreover, the evaluator a_c also updates its provider graph to include the link between $a_{r'}$ and $a_{r''}$, since $a_{r'}$ obtained a direct recommendation from $a_{r''}$. Every time an edge is added or updated, the number of accurate, inaccurate or unused recommendations is incremented; this is represented by $count_{response}$ in the algorithm.

Algorithm 4.5 Provider and Witness Graph Updates for Indirect Recommendations

```
for all indirect recommendation $r^i$ do
    if $r^i.a_{r'} \notin \mathbb{P}\, a_{r'}$ then
        add edge($a_{r'}, a_c$) in $a_e.witnessGraph$
    increment $count_{response}$
    repeat
        if $r^i.a_{r''} \notin \mathbb{P}\, a_r$ then
            add edge($a_{r''}, a_{r'}$) in $a_c.providerGraph$
            increment $count_{response}$
    until $r_\mu = r^d$
```

Let us consider the example shown in Figures 4.1 and 4.3. The evaluator a_1 can extend its service-oriented provider graph from information from the service-oriented witness graph, in this case about the interaction between a_2 and a_5. The extended graph is shown in Figure 4.5.

4.2.6 Agent Graph Elememts

In this section, we discuss the various considerations of the data collection for the agent network building process. The proposed trust and reputation model gathers sufficient information for an agent to be able to effectively select interaction partners, purely based on its own past interactions and the opinions of others. In order to build an extended

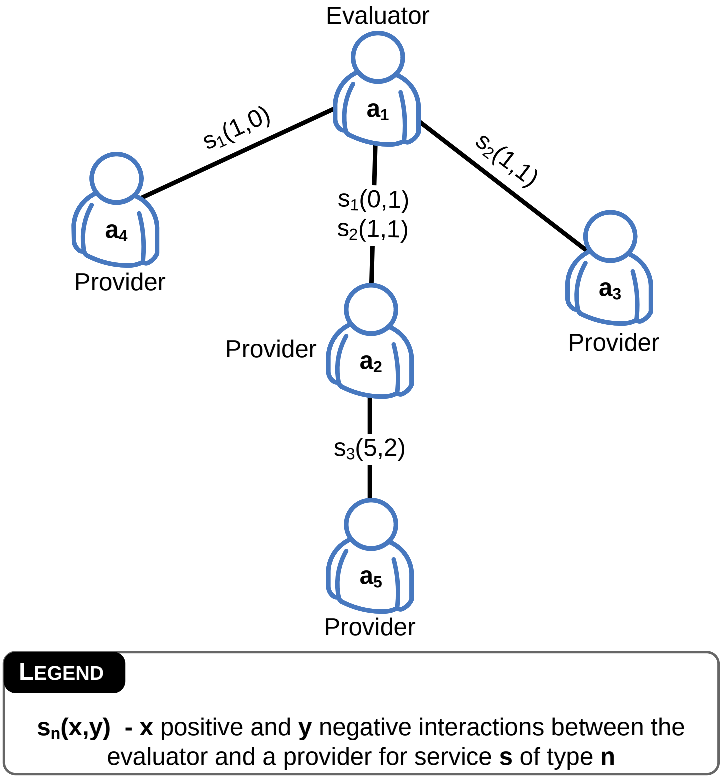

Evaluator
a1
s1(1,0)
s2(1,1)
s1(0,1)
s2(1,1)
a4
Provider
Provider
a2
a3
Provider
s3(5,2)
a5
Provider
LEGEND
sn(x,y) - x positive and y negative interactions between the evaluator and a provider for service s of type n

Figure 4.5: Example of an Extended Service-oriented Provider Graph.

view of their environment, agents will need to gather extra information to learn about other agents and their interactions.

4.2.6.1 Recommendations Count

Our trust and reputation model efficiently selects the most relevant recommendations based on criteria such as length of the recommendation chain and relevant experience. For agents to build accurate agent networks, we believe that evaluators should take into account all the recommendations they obtain, if only to gather information on how the different agents in its environment relate to one another. Consequently, our model allows agents to record all the recommendations they obtain since opinions are excellent indicators of agent activities.

4.2.6.2 Recording Intermediate Recommendations

While the identity of witnesses along a recommendation chain is not essential for an evaluator who assesses mainly its principal witness, it is an important piece of information for an evaluator building an extended view of its agent environment. Our trust and reputation model keeps a record of all intermediate witnesses and their recommendations. This is reflected in the witness graphs which show both direct and indirect recommendations. The trust model incorporates the assessment of indirect witnesses for trust evaluation. However, our evaluation considers the recommendation trust of principal witnesses, rather than that of all witnesses, as we aim to assess the combination of features in our model, especially the use of indirect recommendations.

4.3 Information Extraction

In this section, we demonstrate the usage of the agent network model by an evaluator to extract useful information for decision making. We show two types, namely, discovery of new providers, and learning about the interactions of other agents in the environment. The third type, collusion detection, will be covered in Chapter 5. Graphs presented in this section have been created by the visualisation tool we implemented to show the interactions between agents.

4.3.1 Discovering New Service Providers

The discovery of new service providers refers to an evaluator re-engaging in transactions with a service provider after a period of time, possibly due to previous unsuccessful or inadequate interactions which have lowered the evaluator's trust in the provider.

It may happen that agents previously performing poorly improve and become reliable providers. However, an agent can take longer to discover this change due to various factors, including a long memory, which helps to assess the trustworthiness of agents, but also causes the evaluator to keep this assessment for longer, despite the improved trustworthiness of that agent. One way of countering the downsides of a long memory is to make use of recommendation information to assess the recent behaviour and reliability of the agent of interest.

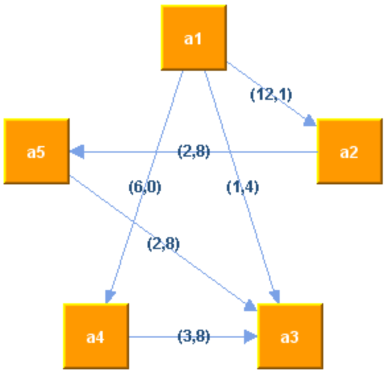


Figure 4.6: Agent-oriented Provider Graph: a_3 Untrustworthy.

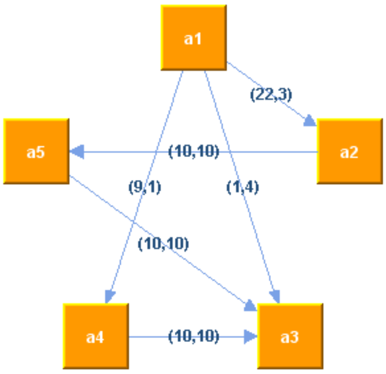


Figure 4.7: Agent-oriented Provider Graph: a_3 Becoming Trustworthy

As an example, let us consider a small population of five agents, with the evaluator being a_1. Figure 4.6 shows a_1's extended agent-oriented provider graph after a number of interactions. The number of agents is 5 in the graph while the population size is 10. This is due to the evaluator only interacting with agents it needs to provide services and recommendations. It directly interacts with service providers a_2, a_3 and a_4 and receives direct recommendations from a_4 and indirect ones from a_5. From the interaction count, a_3 is untrustworthy, with more failed interactions. As a result, evaluator a_1 starts relying more on a_2 for the same service type. Provider a_3 however improves its trustworthiness at a later point and evaluator a_1 can notice this change through the recommendations about a_3 that it continues requesting. The assumption here is that the cost of interaction failure with a_3 is high, such that a_1 is less likely to decay its trustworthiness towards a_3 quickly to attempt to renew a transaction in case of a positive behaviour change. Figure 4.7 shows the graph at a later point in time, when the witnesses are reporting an improvement in a_3's trustworthiness. For instance, a_5's recommendation of a_3 has improved from $(2, 8)$ to $(10, 10)$ for the pair of positive and negative interactions, showing that a_3 has been interacting more successfully since the graph in Figure 4.6.

4.3.2 Learning about Neighbouring Agent Interactions

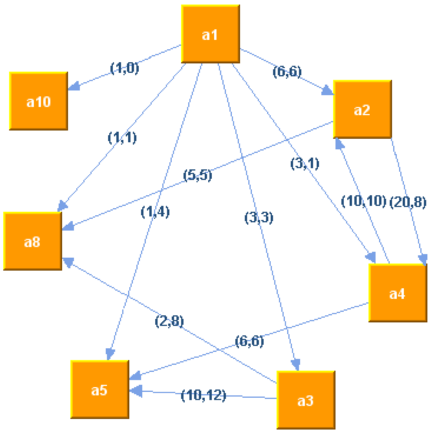


Figure 4.8: Agent-oriented Provider Graph.

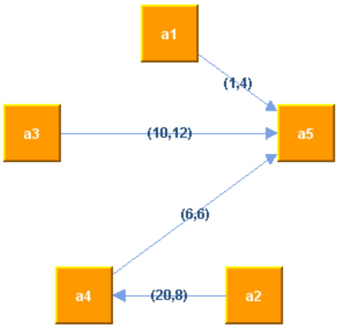


Figure 4.9: Service-oriented Provider Graph for Service s_4.

One of the limitations of decentralised multi-agent systems is that each individual agent has only a localised view of its environment. To allow for a more thorough understanding of its environment, an evaluator needs to gather as much information as it can about its neighbours and their interactions. As an example, let us consider a population of 10 agents, labelled a_1 to a_{10}, where a_1 is the evaluating agent. After a period of interaction with other agents, for services and recommendations, a_1 records these interactions and builds a model of its agent environment. Figure 4.8 shows the agent-oriented provider graph, while Figure 4.9 shows the service-oriented provider graph for service type s_4.

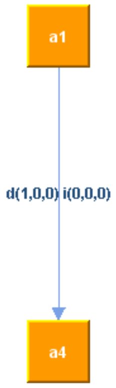


Figure 4.10: Agent-oriented Witness Graph.

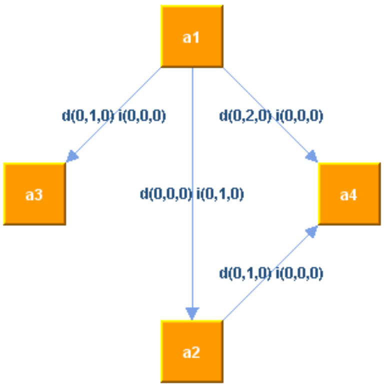


Figure 4.11: Service-oriented Witness Graph for Service s_4.

Evaluator a_1 also uses its recommendations to build witness graphs and updates the graph edges depending on the whether the recommendations are used for interactions with the target agent. Where the recommendations have been used, the evaluator compares its own interaction with the recommendation of the witness, thereby assessing the accuracy of the witness. Figure 4.10 shows a_1's agent-oriented witness graph. In this case, a_1 has received a direct recommendation from a_4 and it was accurate in comparison to a_1's own interaction with the target for which the recommendation was given. Figure 4.11 depicts the service-oriented witness graph for service type s_4. From this graph, the evaluator a_1 learns about agent a_2's interactions with a_4, from the indirect recommendation a_1 received from a_4 through the intermediate witness a_2. Such information helps the evaluator to understand how agents in its environment are interlinked and possibly influence its own interactions and behaviour. The edges of the witness graphs have two elements of information: the direct recommendations and the indirect ones. For example, as shown in Figure 4.10, a_1 has received one direct recommendation from a_4, but no indirect recommendations. The triple $(1, 0, 0)$ indicates that the recommendation was accurate, while there has been no inaccurate recommendations or unused opinions, as described in Figure 4.4.

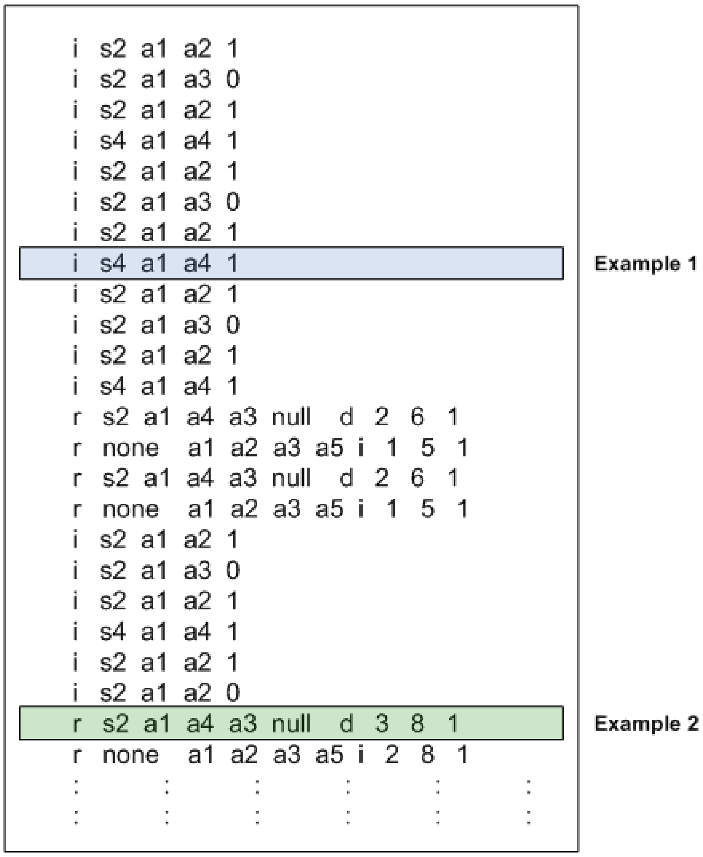
4.3.3 Visualisation Tool

As mentioned earlier in the chapter, the ability to view the agent graphs is a useful tool for human analysts. We have implemented a visualisation tool that enables users to restructure the agent graphs on the screen for easier viewing of the information. We first describe the network building process.

Trust and recommendation information is gathered in histories of interactions and recommendations, and this is represented in Figure 4.12. An entry for an interaction has the following format.

$$i \quad s_n \quad a_e \quad a_p \quad result$$

where i denotes a service interaction between the evaluator a_e and provider a_p for a service of type s_n. The $result$ in this example is a boolean value, 0 for a failed interaction, and 1 for a successful one. Note that the result of the interaction looks at the success dimension, but individual service characteristics can also be recorded for a more detailed analysis. In Figure 4.12, Example 1 shows an example of a service interaction entry. The letter 'i' denotes that this is a service interaction of type $s4$ that the evaluator $a1$ has requested provider $a4$ to do and the result (last column) was successful, as indicated by the number 1.



```
i  s2  a1  a2  1
i  s2  a1  a3  0
i  s2  a1  a2  1
i  s4  a1  a4  1
i  s2  a1  a2  1
i  s2  a1  a3  0
i  s2  a1  a2  1
i  s4  a1  a4  1                                Example 1
i  s2  a1  a2  1
i  s2  a1  a3  0
i  s2  a1  a2  1
i  s4  a1  a4  1
r  s2  a1  a4  a3  null   d  2  6  1
r  none   a1  a2  a3  a5  i  1  5  1
r  s2  a1  a4  a3  null   d  2  6  1
r  none   a1  a2  a3  a5  i  1  5  1
i  s2  a1  a2  1
i  s2  a1  a3  0
i  s2  a1  a2  1
i  s4  a1  a4  1
i  s2  a1  a2  1
i  s2  a1  a2  0
r  s2  a1  a4  a3  null   d  3  8  1            Example 2
r  none   a1  a2  a3  a5  i  2  8  1
:       :       :       :       :       :
:       :       :       :       :       :
```

Figure 4.12: Example Extract of Interaction and Recommendation Histories.

Recommendation history entries have the following format.

$$r \quad s_n \quad a_e \quad a_r \quad a_t \quad a_{r\prime} \quad r_{type} \quad r_{success} \quad r_{failure} \quad result$$

where r indicates a recommendation as requested by the evaluator a_e from principal witness a_r about the target a_t's service type s_n. If the recommendation is indirect, $a_{r\prime}$ represents the secondary witness and the recommendation type r_{type} reflects this. $r_{success}$ denotes the positive number of interactions from the witness' opinion. $r_{failure}$ gives the number of negative interactions the witness has had with the target. A recommendation entry is added after an evaluator has used the recommendation to interact with the target agent. Thus, the last column $result$ gives an indication of whether the recommendation was accurate compared to the actual interaction experienced by the evaluator; a value of 1 indicates that it was accurate, while 0 suggests that the recommendation was not accurate. In Figure 4.12, Example 2 shows a recommendation entry, identified by the letter 'r'. It is a direct recommendation requested by a_1 from a_4 about target provider a_3. The opinions received consist of 3 positive interactions and 8 failed ones. The 1 in the last column expresses similarity in the evaluator's own experience after interacting with a_3.

In our implementation of the agent network model, an agent gathers information about trust and reputation while interacting with others for services and recommendations. For every task, the agent records the outcomes of the interaction and recommendations related to that interaction. The appropriate graphs are thus updated to reflect the current state of the agent's perception of its environment. The events prompting each graph update can be represented as a collection of interaction and recommendation histories, as shown in Figure 4.12. We have developed a visualisation tool using the JGraph[1] and JGraphT[2] Java graph libraries to support the agent network model, particularly in verifying the links among agents, and discovering previously unknown links. The tool can potentially help users to better understand the agent environment, especially when information extraction enables retrospective analysis, for instance, for collusion detection. Sample graphs visualisations are shown in Figures 4.10 and 4.11. Figure 4.10 shows an agent-oriented witness graph, with a directed link from agent a_1 to a_4, with edge information recording that a_1 has received one direct recommendation from a_4 and the triple (1,0,0) shows that one was accurate, none was inaccurate, and none was unused. Figure 4.11 depicts a service-oriented witness graph for the service type s_4. Agent a_1 has requested for recommendation from three agents, a_3, a_2, and

[1] http://www.jgraph.com

[2] http://jgrapht.sourceforge.net

a_4. The only direct recommendation from a_3 was inaccurate. Similarly, a_1 received two recommendations from a_4 and both direct recommendations were inaccurate.

We include some examples of screenshots of the agent graphs, resulting from the use of the visualisation tool. Some graphs have been restructured for the clarity of the nodes and edge labels on the screenshots. Figure 4.13 shows the provider graph for evaluator $a1$ when the population size is 1000. Showing more clearly the links between agents, Figure 4.14 depicts the provider graph for evaluator $a1$ when the population size is 10. For a population of 25, Figure 4.15 shows one the service-oriented witness graphs for the evaluator $a1$.

4.4 Discussion and Conclusions

4.4.1 Related Work

Trust and reputation models, such as HISTOS [142], ReGreT [104], and FIRE [45] use a notion of social network for recommendations. HISTOS, for instance, is a pairwise rating system whereby users form a directed graph with the weighted edge representing the most recent reputation rating from one user to the other. In comparison, our mechanism makes use of service-oriented and agent-oriented provider and witness graphs to represent various relationships among agents. We also aim to use the social network information not only for reputation values, but also to extract other valuable information, such as new providers and collusive agents.

4.4.2 Conclusions

Individual agents making use of trust and reputation to select the most appropriate interaction partners can apply the information further to learn about their environment.

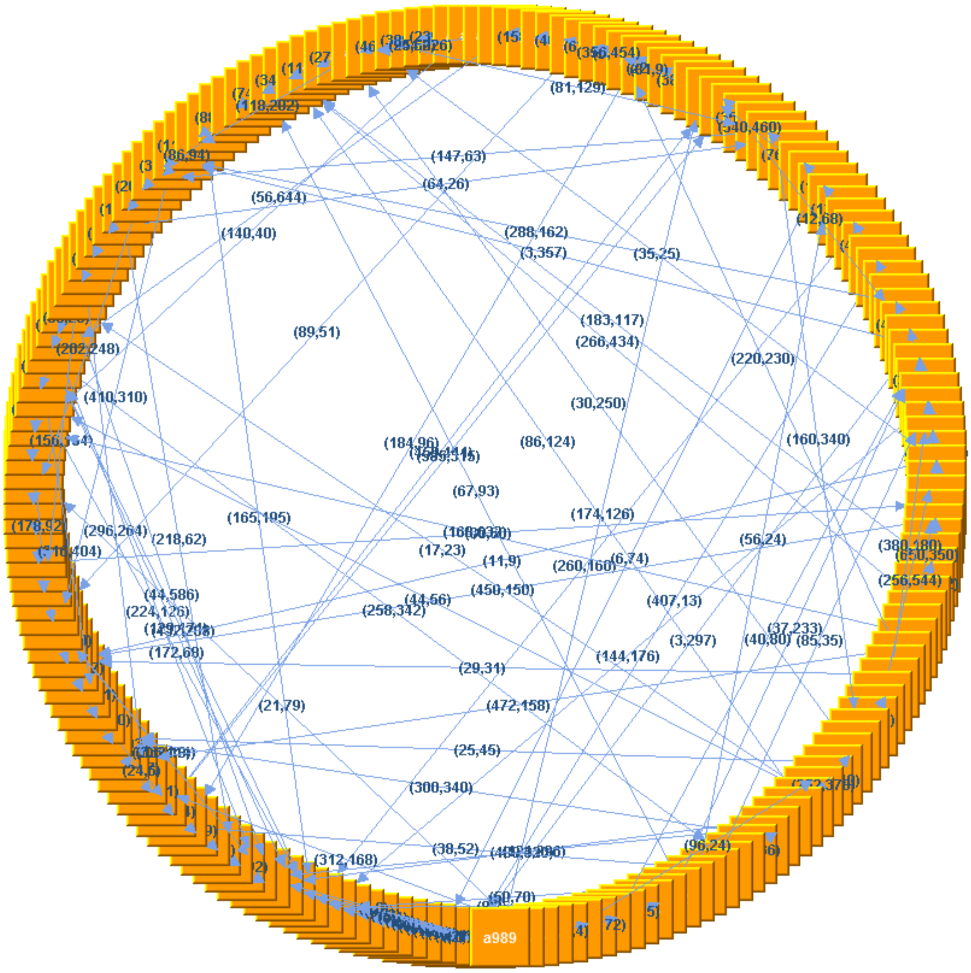


Figure 4.13: Agent-Oriented Provider Graph with a Population Size of 1000.

Decentralised agents suffer from localised and limited views of their environment, which is often insufficient to understand many of the agent interactions. With regard to this limitation, we proposed to enable agents to extend their local view by building a model of their agent network from information they are already gathering through agent interaction and recommendations. The agent network model can then be analysed and valuable information can be extracted about agent relationships and behaviours, that can be useful for the agents' future decision making. We have provided an approach by which agents can represent their agent networks from rich interaction and recommendation information. We have also built a visualisation tool to help human analysts to more easily view the agent population using the agent graphs and to better understand the types of links between agents, as well as their strength. For complex graphs, the tool is especially useful as it allows the graph structure to be restructured for better viewing.

This chapter outlined some of the types of information that can be extracted, from re-discovery of service providers to discovering the neighbourhood of a particular provider. The evaluator uses service-oriented and agent-oriented provider and witness graphs to represent the information collected while assessing the trustworthiness of agents. The agents' social networks can be used as the basis for detecting possible collusion, as will be described in the following chapter.

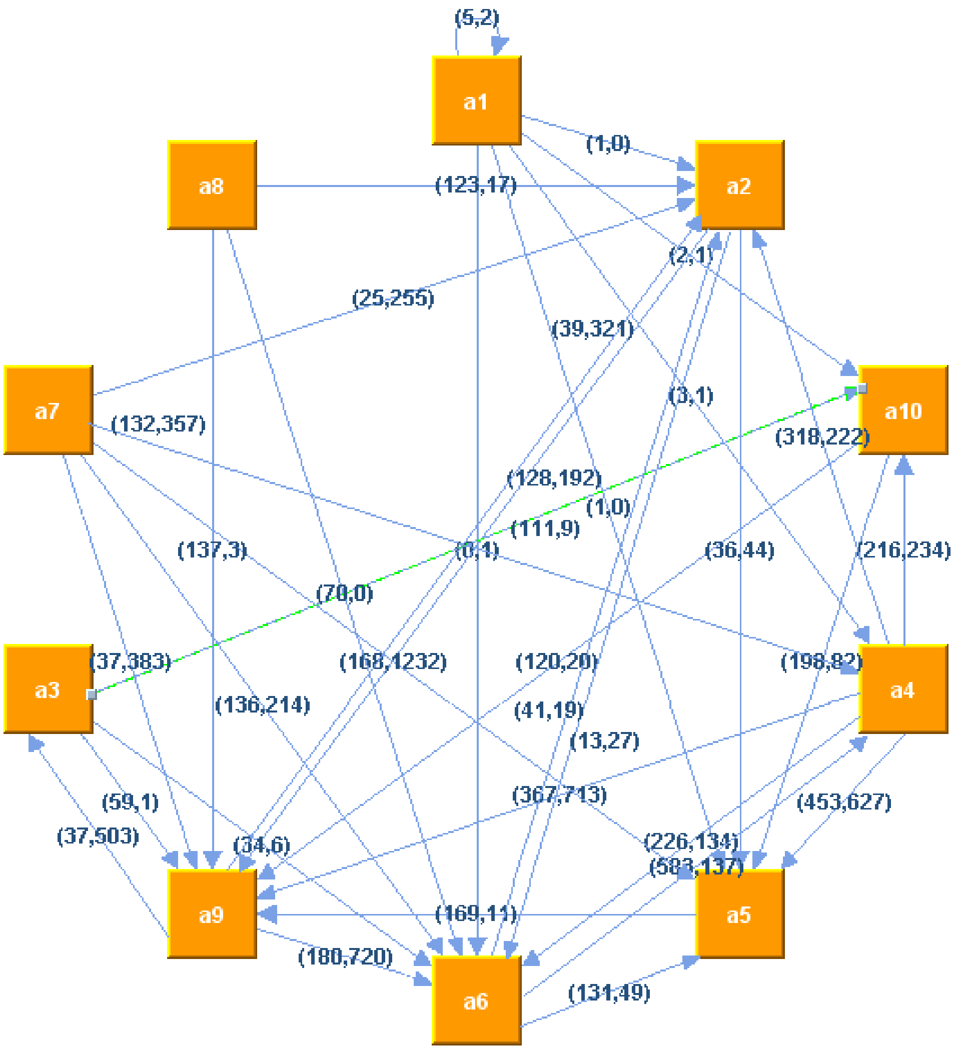


Figure 4.14: Agent-Oriented Provider Graph with a Population Size of 10.

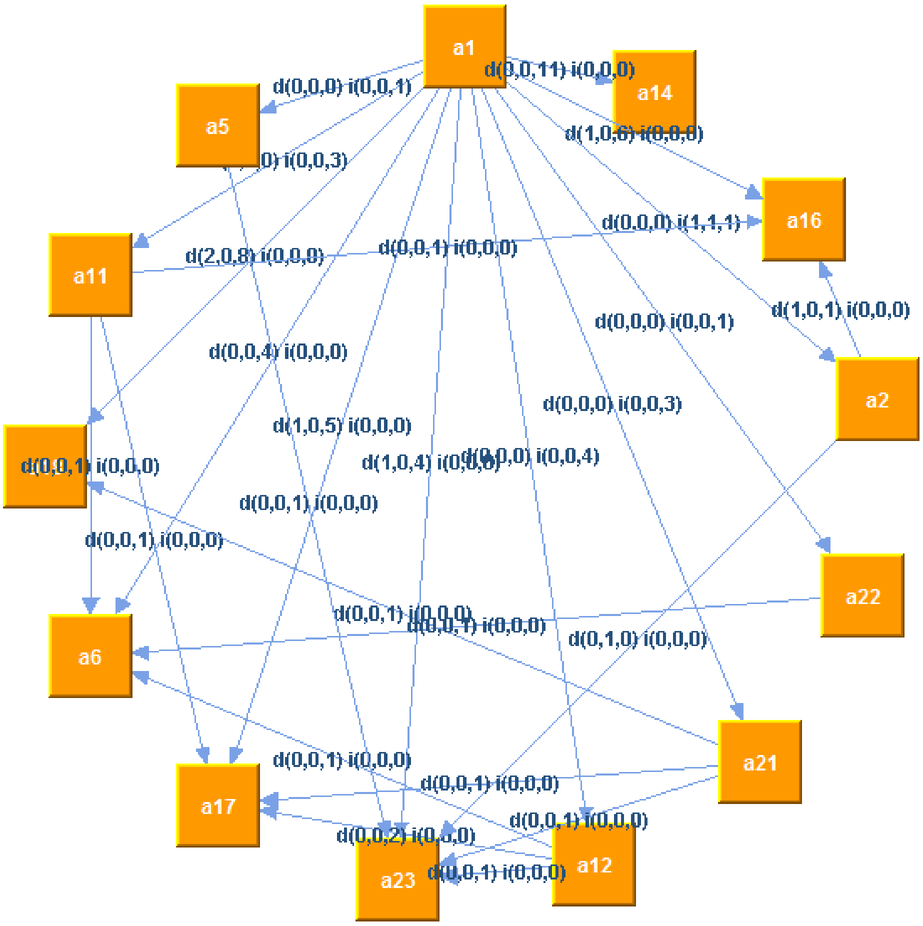


Figure 4.15: Service-Oriented Witness Graph with a Population Size of 25.

Chapter 5

Collusion Detection

5.1 Introduction

Collusion is a phenomenon where competition drives malicious agents to attempt to gain benefits at the expense of other agents in the environment. It is yet another source of uncertainty that self-interested agents experience when interacting in a heterogeneous open and distributed environment. The ability of agents to detect collusion potentially minimises further the uncertainty of interactions they face. Collusion detection is the first step towards finding a solution to this problem, including preventing future collusion. As introduced in Section 1.2, we aim to support collusion detection by identifying the characteristics of collusion in the e-commerce domain, in terms of the interactions and recommendations among agents. We also aim to use the rich set of information gathered about other agents, as well as their social networks to extract knowledge about collusive behaviour, by using existing data mining techniques.

The previous chapter described how building and maintaining agent network graphs helps agents to extract previously unknown information. Collusion detection is one such example and this chapter aims to demonstrate how agents can detect potentially collusive agents by using the information gathered from their interactions and analysing shared information and agent relationships. Collusion has long been a hard problem to solve in agent systems. In this work, we show how individual agents can use information about their agent environment to inform their decision making and be collusion-aware. Our approach aims to enable agents to identify potentially collusive agent pairs. The information obtained is useful for a human user to analyse further or for an agent to incorporate into its agent selection process for future interactions. We propose two main contributions in this chapter. Firstly, we present a new taxonomy of collusion for the e-commerce domain, which classifies collusion by type and by its characteristics. Secondly, we propose to use similarity measurement — a technique which has been commonly used in data mining — for collusion detection.

The rest of this chapter is organised as follows. Firstly, we define collusion and the context of its occurrence. Secondly, we describe our taxonomy of collusion in the e-commerce domain. Next, we look at the collusion detection techniques used in agent-based systems and in other fields. We then present our approach to collusion detection with the use of similarity measurement for an example collusion type. This approach is then evaluated and we conclude with findings and future work.

5.2 Defining Collusion

Collusion is defined as the cooperation among a group of agents to gain benefits at the expense of other agents. Agents exhibiting this behaviour use deception to achieve their goals. For example, a group of provider agents that are not performing well may collude with other agents in the population so that they would boost their reputation when recommendations are requested about those providers. The witnesses may benefit from this arrangement by benefiting from cheaper prices from the provider agents or can expect reciprocal behaviour about recommendations. The deceived party is the agent that has requested recommendations from the collusive witnesses and if these are used for decision making and the collusive providers subsequently used for service provision, the interaction will be below expectations.

Two main types of collusion discussed in the literature are tacit collusion and explicit collusion [129, 41, 101]. Tacit collusion, also known as implicit collusion is considered to be an agreement to collude without any communication among agents. In auctions, for example, bids usually show the intention of the bidders and this can be used to implicitly signal collusion, since there is no explicit communication among bidders about their strategies, in the form of "jump-bidding", "sniping" and withholding bids. Vragov defines jump-bidding as submitting a bid, which is larger than the current high bid plus two minimum increments [129]. Sniping is a phenomenon where bidders wait until the last minutes to submit their bids. Tacit collusion however, does not necessarily involve any collusion in the legal sense of the term, such as the antitrust laws in Articles 101 and 102 of the EU Treaty [15, 49]. Explicit collusion refers to different agents engaging in direct communication and obvious coordination to impact on the price and welfare of other agents [41]. The issue here is to be able to detect agents that are explicitly coordinating their behaviour through illegal means of communication.

In decentralised multi-agent systems, it is difficult for individual agents to be aware of all the communications that take place among agents and detect the ones that concern a collusive agreement. In our work, we are not seeking to have agents detect collusion as a legal requirement, but more as an added benefit to improve the success of their interactions. Agents, therefore do not need to differentiate between tacit and explicit collusion and are mainly concerned about being able to detect agent behaviour that has similar results to collusion. As the characteristics of explicit collusion are better defined than tacit collusion, we will focus on explicit collusion in this thesis.

5.3 Collusion Detection

The detection of explicit collusion has been studied from an economics perspective, for example, in the work by Harrington [41], where cartels are discovered using a structural approach or a behavioural method. The Cambridge online dictionary defines a cartel as "a group of similar independent companies who join together to control prices and limit competition"[1]. The structural approach of detection involves identifying markets with traits thought to be conducive to collusion, such as fewer firms, more homogeneous products and a more stable demand. In contrast, a behavioural approach involves either observing the means by which firms coordinate or observing the end result of that coordination. In some cases, it is the means by which companies colluded that leads to the discovery, for instance, proof of cartel meetings, or an employee involved in the conspiracy speaking out. Alternatively, the behavioural approach can focus on the market impact of that coordination, for instance, suspicions can arise from the pattern of the firms' prices or quantities or some other aspect of the market behaviour. As Harrington points out, governmental agencies and private corporations actively search for illegal activity, however, there are no analogous policies when it comes to illegal cartels. He believes that economic analysis can play a greater role in the detection of cartels. Even though economic analysis alone might not be sufficient to detect and prosecute cartels, it can play a more active role in identifying the industries worthy of closer inspection.

With respect to the above discussion on collusion and its detection, we believe

[1] http://dictionary.cambridge.org/

that the issue of collusion detection is still mostly unresolved in agent-based systems, which automate their physical world counterparts. In particular, collusion detection at the individual agent level is still an open problem. Very few interaction mechanisms can prevent or deal with collusion [100]. For instance, Brandt [7] proposes a private and secure auction protocol that ensures that malicious bidders that do not follow the publicly-verifiable protocol, are detected immediately and can be excluded from the set of bidders. However, this solution is a system-level solution, which detects collusion based on the rules of the auction protocol. Palshikar and Apte [95] propose a graph clustering algorithm to detect collusion sets in the stock trading domain. They then use the Dempster-Schafer theory of evidence to combine the candidate collusion sets detected by the individual algorithms. This solution has been designed to work on a trading database, which can be considered as requiring a global view of the system. On an individual level, detection of collusion by agents has been much less researched.

To be able to detect collusion, agents need to identify the characteristic behaviours that constitute collusion, which can be of different types. In the following section, we propose a taxonomy of collusion to classify collusion by type and characteristic features.

5.4 Taxonomy of Collusion

The detection of collusion necessitates the identification of the types of collusion and their characteristics. Smed *et al.* [117, 118] classify collusion among multi-player online games, such as poker and real-time strategy games, namely Age of Empires III[2], according to four main aspects related to the agreement among the colluders. The four main aspects are: (i) *consent* (the agreement on collusion, whether tacit or explicit), (ii) *scope*

[2] http://www.ageofempires3.com, Age of Empires III, Microsoft Corporation, 2005

(areas of the game affected by collusion), (iii) *duration* (identifies the start and end of the collusion activity), and (iv) *content* (specifies what is exchanged, traded, or donated during the collusion process). Other aspects may need to be included depending on the domain. For example, an additional aspect in the domain of online computer games is the *role* of the partakers — players and participants — in the game.

Smed *et al.* consider that collusion occurs only where cooperation is forbidden by the rules of the game. In the context of the distributed multi-agent systems, agent cooperation is essential as agents usually cannot achieve their individual goals on their own. However, collusion also exists in these systems due to certain types of cooperation being considered malicious, such as promoting an unreliable agent in exchange for some benefit.

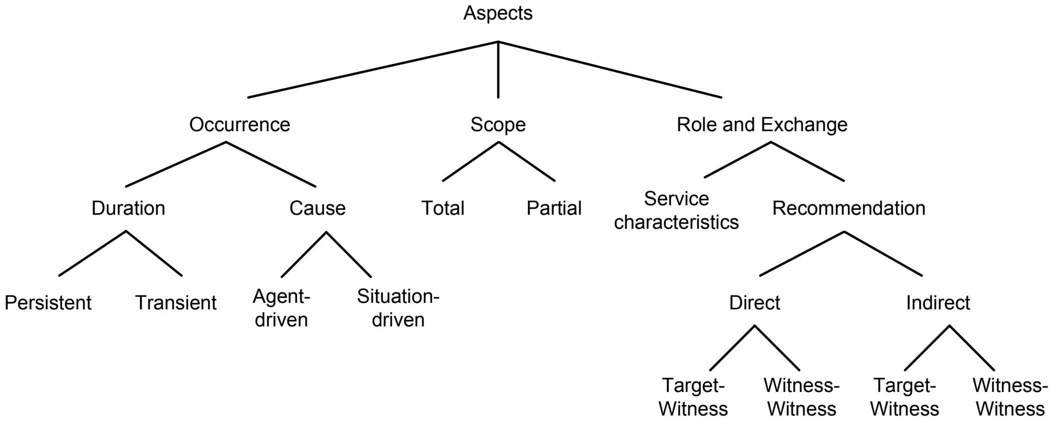


Figure 5.1: Classification of Collusion Aspects.

Drawing on the categorisation outlined by Smed *et al.*, we present a classification of collusion types that we believe is most suited for e-commerce, and with their particular characteristics. Figure 5.1 shows the aspects of collusion and the sub-categories we are considering. We are not making any distinction between tacit and explicit collusion, hence we do not include Smed *et al.*'s use of the Consent aspect. We use the remaining three aspects of scope, duration and content, presented by Smed *et al.* and also their consideration of roles in the collusion [117]. However, we adapt the aspects to the context of agent-based systems and propose a new classification, further detailed in the next section. Therefore, our taxonomy uses Smed *et al.*'s scope, duration, content and role aspects for classifying collusion types. In our taxonomy, the scope aspect is used as in Smed's, while we have included the duration aspect in a wider occurrence aspect that also takes into account the cause of the collusion. We have integrated the content and role aspects from Smed under the heading of "role and exchange" to describe the roles of the collusive partners and what they share during collusion in those circumstances. In contrast to Smed, we have broken down these aspects further, as presented in Figure 5.1.

5.4.1 Aspects of Collusion

Various aspects need to be considered when categorising the types of collusion occurring within decentralised multi-agent systems. The classification of aspects is depicted in Figure 5.1.

5.4.1.1 Occurrence Aspect

The occurrence aspect of collusion indicates the duration of collusion within the interaction period, and the causes of the collusive behaviour. Under the duration sub-aspect, two types can be identified: persistent collusion refers to an agreement that spans the entire duration of the interaction period, while transient collusion occurs temporarily, as agents collude, and disband afterwards.

The causes of collusion occurrence are divided into agent-driven and situation-driven. Agent-driven collusion occurs when the agent's intrinsic state triggers collusive behaviour (poor performance might increase the agent's likelihood to collude with other agents in exchange for a better reputation, for instance). Meanwhile, situation-driven collusion occurs when agents identify that they can benefit from a particular set of opportunistic events or interactions.

5.4.1.2 Scope Aspect

The scope defines the extent of the collusive behaviour. Total collusion suggests that agents will collude in all their common areas of interest, partial collusion only concerns some areas of interest, while the agents will compete with one another in other areas.

5.4.1.3 Impact (Role and Exchange) Aspect

This aspect concerns the roles that collusive agents play in the collusion and what gets exchanged as part of the agreement. The service characteristics sub-aspect refers to the agreement of agents to adopt certain levels of service, in dimensions such as success, timeliness, cost and quality. For example, price fixing is an example of collusion in the cost service characteristic.

Recommendation is a subcategory of the impact aspect and describes the exchange of opinions among agents. Since recommendations can involve both direct and indirect opinions, the exchange of such recommendations occurs in Target-Witness (TW) collusion and Witness-Witness (WW) collusion. TW collusion involves collaboration between the target agent (the agent being evaluated by the observing agent) and a witness (the agent giving a recommendation about the target to the evaluator). WW collusion occurs among witness agents, when they are giving recommendations about the target agent to the evaluator. In both types, the nature of the agreement can be to promote or demote the agents involved, with the view of benefiting the colluding set of agents.

Different types of collusion are characterised by different sets of aspects. In the following section, we will look at the different types of collusion in our domain of interest.

5.4.2 Types of Collusion

Table 5.1 presents the different types of collusion and their corresponding set of aspects. The label TW refers to Target-Witness collusion and WW means Witness-Witness collusion. We now describe the different types of collusion, grouped by common characteristics.

5.4.2.1 Recommendation Collusion

In this group, the types of collusion involve recommendations being exchanged between witnesses and the evaluator agent, who is requesting opinions about a target agent. Agents participate in collusive activity at various levels, with the most relevant examples detailed below.

Type/Aspect		**Scope**		**Occurrence**				**Impact (Role and Exchange)**				
Group	Type	Total	Partial	Duration		Cause		Service-characteristics	Recommendation			
				Persistent	Transient	Agent-driven	Situation-driven		Direct		Indirect	
									TW	WW	TW	WW
Recommendation Collusion	Persistent TW	o	o	x			x	^	*		*	
	Transient TW	o	o		x	*	*	^	*		*	
	Persistent WW	o	o	x			x	^		*		*
	Transient WW	o	o		x		x	^		*		*
	Transient TWW		x		x	*	*	^	*	*	*	*
	Direct Witnesses	o	o	o	o	*	*	^	*	*		
Provider Collusion												
	Price fixing		x	x			x	x				
	Market division	o	o	o	o		x	x		^		^
Customer-involvement Collusion												
	Customer-Provider	o	o	o	o	*	*	x				
	Customer-Witness		x	o	o	*	*	x		*		*

Table 5.1: Characteristics of Collusion Types.

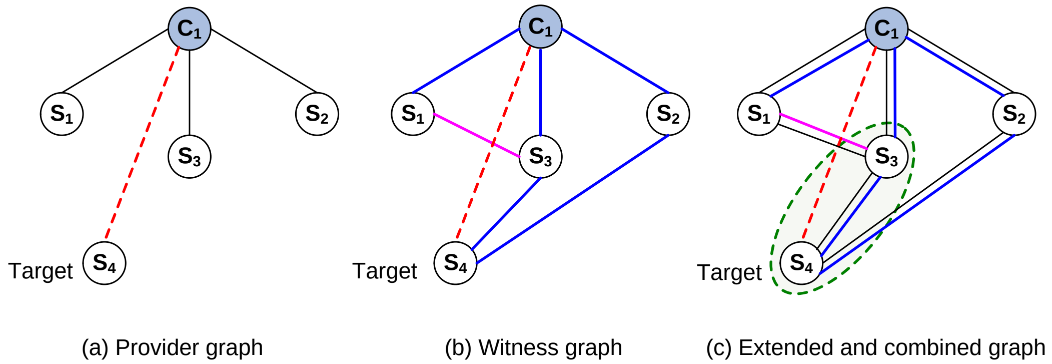
Legend:

○	within the higher-level aspect, an exclusive-or applies to the sub-aspects
×	this aspect is mandatory
∗	one or more of the sub-aspects can occur
∧	this aspect is optional

Persistent Target-Witness Collusion (Persistent TW)

This type of collusion occurs between a target and a witness, where the witness promotes or demotes the target with regards to the evaluator systematically, that is, every time a recommendation is requested. The scope of the collusion can be partial or total in that the witness will always have the same behaviour towards the collusive target for some or all services. The collusion is persistent, suggesting that the collusive agreement spans the interaction period. Let us consider an example where the witness promotes the target. The agreement consists of promoting the target via recommendations in one or more of the relevant service characteristics. Figure 5.2 depicts the Persistent TW collusion based on the e-supply chain scenario. The evaluator is Customer C_1, which is already using the services of three providers, Suppliers S_1, S_2, and S_3. Now C_1 needs a new type of service, which is offered by Supplier S_4. However, C_1 has never interacted with S_4 and therefore decides to request for recommendations from agents who have. Figure 5.2(a) shows C_1's provider graph. The solid lines represent direct interactions between two agents, while the dashed line shows the target agent that the evaluator is considering for interaction. Agent C_1's witness graph, Figure 5.2(b) shows the witnesses it uses, through the bold solid lines in the diagram. For instance, S_1 has not interacted directly with S_4 and therefore only gives an indirect recommendation to C_1, via S_3.

The combination of the provider and witness graphs gives Figure 5.2(c), from which the evaluator can extract information not previously known about certain agent relationships. An additional provider graph edge, between S_1 and S_3 can be derived from the provider and witness graphs. Since S_1 has provided an indirect recommendation to C_1 and S_3 is the only secondary witness, this implies that S_1 and S_3, have direct service interactions. The dashed line circling S_3 and S_4 shows potential collusion between the target S_4 and the witness S_3. The evaluator C_1 requests recommendations about



(a) Provider graph (b) Witness graph (c) Extended and combined graph

Figure 5.2: Target-Witness Collusion.

target S_4 from its three service providers, S_1, S_2 and S_3, who can be considered to be trustworthy enough to take their opinions into consideration. From the combined graph Figure 5.2(c), the evaluator C_1 observes over a period of interaction that S_1 and S_3 have similar recommendations about S_4, as compared to the recommendations of S_2. The emergent information is that S_1's indirect recommendation has been obtained along a recommendation chain of length 2, via S_3. Subsequently, as the recommendations from S_3 are more positive than that of S_2, and from its own initial direct interactions with S_4, C_1 can suspect that S_3 is colluding with S_4 to promote S_4 as a trustworthy provider.

Without the agent network, the evaluator, using only trust and recommendations, would eventually have a low recommendation trust in both witnesses S_1 and S_3, without identifying that S_3 was the dishonest agent. Recommendation trust ensures that the evaluator can distinguish between those witnesses giving accurate opinions, when these are compared to the actual interaction with the target, if the recommendation is followed. However, low recommendation trust gives no indication of the reason behind the inaccuracy, whether it is only due to differing experiences or due to malicious intent. Differences in recommendations do not necessarily indicate collusion. However, an agent may want to be aware of the difference in agent behaviour for future interactions. Recommendation trust will be lower for witness S_3 and from the collusion detection, the evaluator can identify that the pair of target and witness involved are S_4 and S_3.

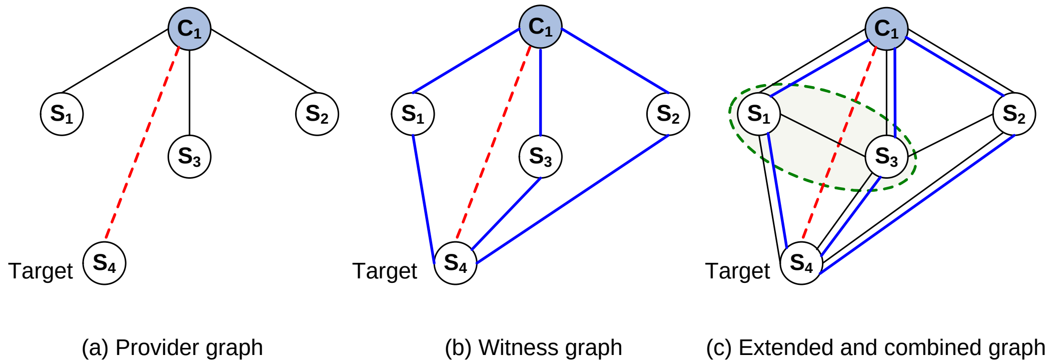
Transient Target-Witness Collusion (Transient TW)

A transient collusion between a target and a witness is similar to the persistent TW collusion, except for the duration of the agreement. In this case, the collusion group forms, collaborates and disbands during the interaction period. The collusion may be agent-driven as well as situation-driven, with the agreement being initiated, for example, by the target agent to temporarily increase its reputation.

Persistent Witness-Witness Collusion (Persistent WW)

In a persistent collusion between two witnesses, the colluding agents collaborate to give information to the requesting agent that will, for example, lower the reputation of the target agent, and the agreement lasts throughout the interaction period. For every recommendation request that these witnesses get about the target, they will give an opinion that will demote the target, and this can involve one or more service characteristics for the services the target offers. Direct or indirect recommendations can be involved in this type of collusion. An example of witness-witness collusion is described below, between suppliers S_1 and S_3, as shown in Figure 5.3.

The evaluator C_1 obtains direct recommendations about target S_4 from witnesses S_1, S_2, and S_3. Again, C_1 has had no past interactions with S_4. Figure 5.3(a) shows C_1's provider graph, with the solid lines representing direct service interactions and the dashed line indicates C_1's interest to interact with S_4. Figure 5.3(b) is different from Figure 5.2(b) as the recommendations obtained are all direct recommendations about S_4.



(a) Provider graph (b) Witness graph (c) Extended and combined graph

Figure 5.3: Witness-Witness Collusion.

The extended and combined graph, Figure 5.3(c), shows the additional information that the evaluator C_1 can infer from the trust and reputation information gathered. Frequent similarity of recommendations from S_1 and S_3, compared to other recommenders, could suggest a potential case of collusion between these witnesses, especially if the opinions are inaccurate compared to the actual agent interactions. This is depicted by the dashed line circling S_1 and S_3 in Figure 5.3(c). Although S_2 and S_3 appear to have similar links as S_1 and S_3, the comparison of their recommendations helps determine that S_1 and S_3 are potentially collusive, while S_2 and S_3 are not considered to be in this category. Witnesses collude, for example, to lower the trustworthiness of the target as viewed by the evaluator to prevent the target from being swamped with interaction requests, which could potentially increase competition for the witnesses to interact with the target as a supplier. Again, the similarity or dissimilarity of recommendations does not necessarily imply collusion. However, the evaluator can identify those agents that are behaving differently early on and can act on this information to confirm or disprove the existence of collusion.

Transient Witness-Witness Collusion (Transient WW)

Two witnesses can agree to collude with each other in order to affect the reputation of a target agent for a certain duration, for instance, when one or both want to limit the target's transactions with agents other than themselves for the services concerned. Compared to a persistent collusion, the decision to enter a collusive agreement can arise from an opportunity to temporarily benefit from the agent interactions. The decision to adopt a transient strategy may also be due to the higher risk of being detected in a longer-term agreement.

Transient Target-Witnesses Collusion (Transient TWW)

In this collusion scenario, the target agent colludes with witnesses such that these witnesses promote the target during recommendations about specific services. This can be driven by mutual benefits or by the target's aim to increase its reputation. Exchanges can occur in a combination of direct, indirect TW and WW agreements. Other collusion characteristics include being transient and the scope of the collusion concerning a partial set of services that the collusive group are involved in.

Direct Witnesses Collusion

Collusion among principal witnesses is a subset of the witness-witness and target-witness collusion types, where recommendations are involved. The scope of the collusion can be partial or total, the agreement persistent or transient.

5.4.2.2 Provider Collusion

Provider collusion involves the service providers, where they agree on terms that will provide them with benefits over other service providers and have a larger share of the market. Two examples are price fixing and market division among competitors.

Price Fixing

Price fixing involves a group of competing provider agents agreeing to raise or maintain the sale price of their services. In a particular scenario, the aspects of the collusion can be as follows: the collusive set of providers agrees to apply the collusion to only part of their overall services. For instance, the group could be selling certain services at the same price, while for others, competition is in force as would be the norm. The agreement spans the interaction period and is situation-driven as a result of the collusive group wanting to have the better share of the market and hence profits.

Market Division

Also known as market allocation scheme, this type of collusion involves the competing agents agreeing how to divide the market among themselves. They allocate specific customers or types of customers, products or territories among themselves [92]. An example of such a scheme, as in Table 5.1, can involve the participation of a group of service providers either for all of their common services, or a subset.

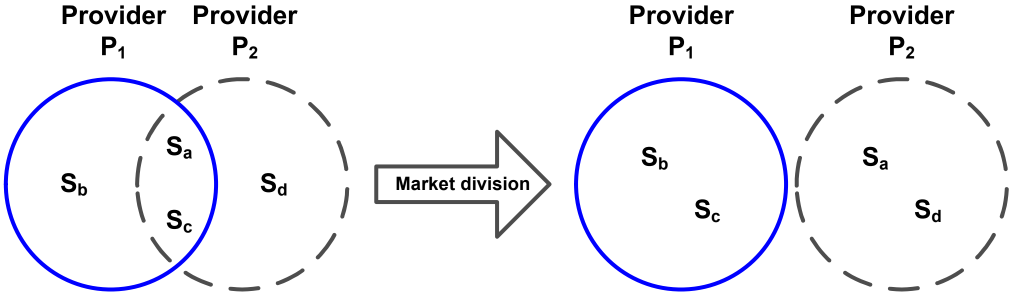


Figure 5.4: Market Division Collusion.

Let us consider the case of two providers P_1 and P_2, each offering these respective services: $P_1(S_a, S_b, S_c)$, and $P_2(S_a, S_c, S_d)$. In normal circumstances, both providers would be sharing the market for services S_a and S_c, which they both offer. However, if P_1 and P_2 agree to collude to divide the market for services S_a and S_c for exclusivity, then, P_1 might be the only one providing S_c, while P_2 might be the only provider of S_a, as shown in Figure 5.4. Other characteristics of this type of collusion include a persistent or transient agreement based on reducing the competition between the two providers and this can be achieved for example by P_1 advertising very high prices for service S_a, which leads to customers preferring P_2. Other service characteristics besides price can also be used to achieve market division. Witnesses could play a role in reinforcing the positions of each provider in their respective market allocations.

5.4.2.3 Customer-involvement Collusion

This type of collusion involves the customer agent, who is interested in a particular type of service and collaborates with witnesses and providers to achieve its goals. We now describe two particular types of collusion where the customer agent is involved.

Customer-Provider Collusion

In the customer-provider collusion, the agreement can involve one or more service characteristics for the services of interest. For instance, a colluding provider can fail in its service provision, but the customer will lie about this, in return for other benefits.

Customer-Witness Collusion

A customer agent can collude with witnesses (to other potential customers of a particular service) to favour a specific target instead of another, for the purpose of having the provider free to accept service requests from the customer. If more than one witness is involved, there might exist WW collusion as well.

5.5 Collusion Detection Techniques

Having discussed various types of collusion, we now explore some of the collusion detection approaches and discuss the most appropriate ones. We first discuss the collusion detection techniques used in agent-based systems, followed by the approaches used in other related fields such as data mining and intrusion detection.

5.5.1 Agent-Based Solutions to Collusion Detection

In agent-based systems, solutions to collusion typically require an agent to have a global view of its environment to be able to apply system-wide measures, such as in that proposed by Jurca [52], or require the use of additional procedures, such as transaction proofs [119]. Similarly, argumentation can confirm or contradict the occurrence of lies in agent communication [90, 91]. Lying is a form of malicious behaviour that may contribute to agent collusion. We describe some of these solutions below and discuss why they are not suitable for our purposes.

Online reputation systems rely on honest feedback from users, and some self-interested agents may find benefit in lying. Jurca [53, 52] proposed a feedback payment scheme to ensure that users get more benefit when telling the truth compared to lying. In P2P systems, TrustGuard [119] has been designed to be an efficient reputation system in the face of malicious behaviour, by tackling fake transactions with transaction proofs. An unforgeable transaction proof uses a public key cryptography-based scheme and serves as a proof that the transaction took place when sharing feedback. Some work has also been done in grid computing systems, where a collusion detection algorithm has been proposed for the outcomes of votes conducted as a result of majority voting for the purpose of verification [121].

The solutions discussed above either require a global view of the system or need additional processes in place to detect collusive behaviour. In our case, individual agents need to be able to detect collusion from their local view of their environment, using information they have gathered during interaction and recommendations for the agent selection. We propose a solution that allows individual agents to retrospectively assess whether collusion has occurred and to predict future behaviour on this basis. Jurca's incentive-compatible collusion resistant payment mechanism, as well as TrustGuard, relies on having a mechanism in place to provide incentives for agents to comply with a certain behaviour, or have checks in place to ensure compliance. Our approach is based more on a form of social compliance, where agents are expected to share information out of reciprocity, but also when punishments for malicious behaviour are applied by individual agents. For instance, if an evaluator detects collusion among two other agents, it may choose to stop interacting with them for a period of time, and this action is likely to influence the reputation of these two agents in the agent population.

5.5.2 Approaches in Related Areas

Collusion detection can be viewed as an attempt to identify the underlying structure of heterogeneous data containing collusive behaviour. One way of identifying this structure is in the grouping of data elements, or clustering. Schaeffer [109] surveys graph clustering and the methodologies commonly applied in data mining, including the similarity measures used for clustering. Although similarity measures are used in graph clustering, they can be used on their own to compare values, for instance, to check if two agents are collusive.

Similarity measurement and clustering are existing data mining techniques that enable agents with common or differing characteristics to be identified, thus highlighting their similarity or dissimilarity. Differences in experience among agents occur for many reasons, including different priorities, goals, agent assessment mechanisms and collusion. When collusion occurs there will be a difference in the experiences reported by colluding agents in the form of recommendations, and those experienced directly by the evaluator. As such, any difference in experience may be indicative of collusion. However, irrespective of the reasons behind the differences in experience, it is desirable for the evaluator to identify these differences and incorporate them into its reasoning for future interactions. Therefore, similarity measurement is a suitable approach to help identify differences in agent experience, including those that arise from certain types of collusion.

5.5.2.1 Graph Clustering

Clustering methods are used to group elements with similar characteristics together. A graph is a structure made up of a set of vertices and a set of edges that connect pairs of vertices. Graph clustering is the task of grouping the vertices of a graph into clusters taking into consideration the edge structure of the graph in such a way that there should be many edges within each cluster and relatively few between the clusters [109]. Although a clear definition of what constitutes a cluster has not been agreed, clusters in graphs need to have some desirable properties. Firstly, each cluster should be connected, where there is at least one path connecting each pair of vertices within a cluster. Secondly, the paths should be internal to the cluster. As mentioned in works such as [28, 57], a good cluster is a subgraph where the link density is greater among members of the cluster than between members and the rest of the graph. Figure 5.5 shows what constitutes the properties of a good cluster. In this diagram, the cluster members are drawn in black and their edges are thicker than other edges. The cluster on the left is one of good quality, as it is dense and has more internal edges within the cluster than links outside (referred to as being introvert). The middle cluster is a worse cluster than the left one, as even though it has the same number of internal edges, it has many more external links to vertices outside the cluster. The third cluster, on the right, cannot be categorised as a good cluster as it has few external connections, and even fewer internal links, making the internal density low.

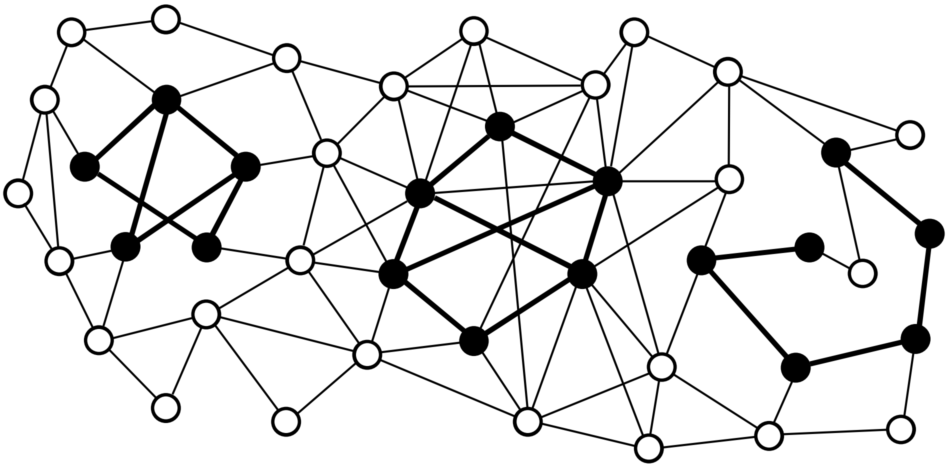


Figure 5.5: Example Graph with Clusters [109, Figure 3].

The two main approaches to identifying clusters are: (i) computing values relevant to the vertices and then classifying the vertices into clusters based on those values, and (ii) computing a fitness measure over the set of possible clusters and choosing among the candidate clusters those that optimise the measure used [109]. Clustering algorithms using the first approach are based on the similarity between vertices. Distance and similarity measures (discussed in Section 5.5.2.2), form part of this approach, as well as adjacency-based and connectivity measures. Adjacency-based measures look at the edges between vertices, rather than the properties of the vertices themselves. For example, vertices can be viewed as similar based on adjacency information if they have overlapping neighbourhoods. Meanwhile, connectivity measures use the number of paths between each pair of vertices to determine similarity. Vertices belonging to the same cluster should be highly connected to one another, as proposed by Hartuv and Shamir's Highly Connected Subgraphs algorithm [42]. The second approach concerns the cluster fitness measures, which are functions to rate the quality of the clusterings. The criteria used include direct density measures, and cut-based measures, which measure the connectivity with the rest of the graph. Some examples of graph clustering algorithms include Markov clustering, minimum-cut trees, and k-means algorithms.

Graph clustering is a well-explored field in unsupervised learning in structured domains, thus providing a way to discover groups of colluding agents. From the similarity measure, pairs of similar agents can be found, and these can be further grouped into sets of collusive agents through clustering techniques. Such techniques have been used in work by Palshikar and Apte [95], where two graph clustering algorithms are used (shared nearest neighbour and mutual nearest neighbour) [33, 50]. The algorithms presented in their work are oriented towards detecting relatively dense subgraphs, and will not always be able to detect sparse subgraphs.

5.5.2.2 Similarity Measurement

The graph clustering methods discussed previously make use of similarity measures. Similarity measures are commonly used in data mining decisions [43], for example in returning relevant documents following a search engine query. Since similarity measures can be used to compare two objects, it is reasonable to consider that they can be used to compare the behaviours and interactions of two agents.

Similarity and dissimilarity are used in many data mining techniques and they refer to the function of the proximity between the corresponding attributes of two objects.

Similarity is a numerical measure of the degree to which the two objects are alike, while dissimilarity is the measure of their difference [124]. We can apply these measures to determine how close or far apart agents' behaviours are. Differences in agent behaviour can give an indication of potential collusion within a subgroup of agents, as previously illustrated by Figure 5.2 in Section 5.4.2.1. The choice of an appropriate similarity measure is influenced by the domain, the characteristics of the data and the purpose of the similarity measure. This is mainly due to similarity measures being suitable for particular types of data characteristics. Tan *et al.* [124] present the data characteristics that need to be considered when selecting similarity measures. Based on these characteristics, we analyse the data characteristics in our domain and determine the most appropriate similarity measures to use. A summary is shown in Table 5.2. In the following sections, we first describe the main characteristics of data that are used in data mining. Secondly, we present the most commonly used similarity measurements. Finally, we analyse the domain and the purpose of the similarity measurement for collusion detection and motivate our choice of measurement.

Similarity Measure **Data Characteristic**	Euclidean distance	Mahalanobis distance	Jaccard Coefficient	Extended Jaccard	Cosine	Correlation (Pearson)	Simple Matching Coefficient
Binary			●				●
Non-binary	●	●		●	●	●	
Sparse			●	●	●		
Dense	●	●				●	
Correlated		●					
Uncorrelated	●		●	●	●	●	
Different ranges		●					
Magnitude important	●						
Magnitude unimportant					●		

Table 5.2: Matching Similarity Measures to Data Characteristics.

5.5.2.3 General Data Characteristics

In this section, we look at the general data characteristics, as listed in Table 5.2. The objects being compared have one or more attributes. For example, the quality and price of a product are two attributes of that product. *Binary* attributes only have true or false values, represented by 1 or 0. For example, in a test, the answers to true or false questions are binary. In comparison, non-binary data have continuous values, such as the frequency of a word in a document. *Sparse* data occurs when the objects do not link to most of the other objects in the domain. For example, if the objects being compared are words in a document, the data is sparse as the frequency of most words in the document is low. Dense data, in contrast, refers to large amounts of links between the objects being compared. For example, a time series for the daily average temperature in London has a dense characteristic. The *correlation* of the attributes is another characteristic to consider. Correlation is a measure of the linear relationship between the attributes. For instance, the proportion of correct answers for a test is perfectly correlated to the proportion of incorrect answers, as the number of correct and incorrect answers to a test are dependent variables (as one increases, the other decreases and vice-versa). If the attributes have *different ranges*, this must be taken into account for a similarity measure such that all attributes have an equal impact on the calculation. As an example, if age and income are two attributes used to compare the similarity of staff, income would have a greater impact on the measure. The *importance of the magnitude* of the data needs to be taken into account as different similarity measures handle this differently.

5.5.2.4 Similarity Measurements

Table 5.2 outlines some of the most commonly-used similarity measurements and the data characteristics for which they are more suited [124, 138]. The general data characteristics that match particular similarity measurements are indicated by a dot (•) in the table.

Euclidean distance: Distance between two points x and y in one or more dimensional space given by the equation:

$$Euclidean(x,y) = \sqrt{\sum_{k=1}^{n}(x_k - y_k)^2)}$$

where n is the number of dimensions and x_k and y_k are, respectively, the k^{th} attributes of x and y.

Mahalanobis distance: A generalisation of the Euclidean distance which normalises the attributes using a covariance matrix, thus removing the issue of the differences in scales of the different attributes. The distance between objects x and y is given by:

$$Mahalanobis(x,y) = (x-y)\Sigma^{-1}(x-y)^T$$

where $(x-y)^T$ is a multivariate vector and $\sum^{-1}$ is the inverse covariance matrix.

Jaccard coefficient: Measures the number of similar elements in two sets x and y compared to the diversity of elements they both hold for binary attributes. The coefficient is given by:

$$Jaccard(x,y) = \frac{x \cap y}{x \cup y}$$

Extended Jaccard coefficient: Also known as the Tanimoto coefficient, it applies to non-binary sets x and y and is calculated using the equation:

$$ExtendedJaccard(x,y) = \frac{x \cdot y}{\mid x \mid^2 + \mid y \mid^2 - x \cdot y}$$

where $x \cdot y$ is the dot product of x and y and $\mid x \mid$ is the magnitude of x.

Cosine similarity measure: Measures similarity by calculating the cosine of the angle between two vectors, represented by x and y, given by:

$$Cosine(x, y) = \frac{x \cdot y}{\mid x \mid\mid y \mid}$$

Pearson correlation: A measure of the correlation between two objects x and y and it reflects the degree of linear relationship between them. The correlation is defined by the equation:

$$Pearson(x, y) = \frac{covariance(x, y)}{standard\_deviation(x) * standard\_devistion(y)}$$

Simple Matching Coefficient: Measures the number of matching attribute values compared to the total number of attributes in two objects x and y both consisting of binary attributes. Given that p is the number of true attributes for both x and y, q is the number true attributes for x and false for y, r is the number of false attributes for x and true attributes for y and s is the number of false attributes for both x and y, the coefficient is given by the equation:

$$SMC(x, y) = \frac{p + s}{p + q + r + s}$$

5.5.2.5 Similarity Measurement for Collusion Detection in the E-Commerce Domain

The previous sections have introduced some common similarity measurements and the data characteristics for which they are more applicable. We have identified the typical characteristics of the data collected by an agent about its environment, within the e-commerce domain. These include mainly positive and negative interactions as experienced by the agents themselves or shared as recommendations. The purpose of the similarity measure is to compare interaction experience between agents and dissimilarities indicate possible collusion. These are now described and they are shown as shaded rows in Table 5.2.

Non-binary: The interaction history and recommendation accuracy are non-binary values. These are counts that depend on service requirements and the number of witness recommendations shared.

Sparse: The data is considered to be sparse as agents do not interact with every other agent in the population.

Uncorrelated: The attributes are considered to be uncorrelated as they are counts that are independent of each other. If the attributes were proportions instead, they would be correlated as the division by the sum of counts makes the proportions dependent on one another.

Similar ranges: The ranges of values for the attributes are similar. For example, the agent-oriented graphs have edges between a customer and a provider with the positive and negative interaction counts as the weight on the edges. If the total number of interactions is 100, the range of positive interactions is [0,100] and that of the negative interactions is also [0,100]. For attributes with different ranges, the values will need to be normalised, while paying attention to any changes to the other data characteristics. For instance, normalising the attributes may make the attributes correlated.

Magnitude unimportant: Considering the positive and negative interaction counts, the magnitude of those values only indicates the amount of transactions between the agent pair. Even a small number of interactions can give an indication of the kind of interactions between the agents. Additionally, the interaction counts shared depend on the size of the interaction window as used by the agent to store past interactions, and this differs from agent to agent. Consequently, we do not consider magnitude of the data for choosing similarity measurements for collusion detection.

Following the analysis of the data characteristics of the domain and the different similarity measures, the cosine similarity measure is the most appropriate with the largest number of matching data characteristics. With reference to Table 5.2, Pearson correlation and Extended Jaccard are the next most suitable similarity measures.

5.5.3 Cosine Similarity Measure

Based on the above analysis of similarity measures and data characteristics, the cosine similarity measure is the most appropriate to compare the similarity of agent behaviours. This technique is popular in the related field of intrusion detection, such as the work of Liao and Vemuri [67], and Sharma *et al.* [116]. The similarity of text is the focus of these works; however, parallels can be found with regard to the identification of characteristics to compare. Cosine similarity has also been used as a technique for user profile-item matching, in the area of intelligent recommender systems on the Internet [85]. The user profile is used to recommend new items considered relevant to the user. Content-based filtering systems use direct comparisons between the user profile and new items, thus requiring a user profile-item matching technique. Cosine similarity can be used, along with a number of other techniques, including standard keyword matching, nearest neighbour, and classification.

The cosine similarity, also known as the Ochini coefficient, is a common measure that uses the dot product and the angle between vectors to compute the similarity.

Vector representation of data consists of one or more dimensions that are considered for each data item. The cosine similarity between two vectors representing the behaviour of two agents, $a_\alpha = (a_{\alpha,1}, a_{\alpha,2}, \ldots, a_{\alpha,n})$ and $a_\beta = (a_{\beta,1}, a_{\beta,2}, \ldots, a_{\beta,n})$ is calculated as:

$$\begin{aligned} \mathrm{Cosine}(a_\alpha, a_\beta) &= \frac{a_\alpha \cdot a_\beta}{\mid a_\alpha \mid\mid a_\beta \mid} \\ &= \frac{a_\alpha \cdot a_\beta}{\sqrt{\sum_{k=1}^{n}(a_{\alpha,k}^2)}\sqrt{\sum_{k=1}^{n}\left(a_{\beta,k}^2\right)}} \end{aligned} \tag{5.1}$$

The resulting measure is an angle in $[0, \pi)$, where the most dissimilar value is $\pi/2$ and zero is the best possible similarity [109]. For agents using our mechanism, the comparison concerns the service interactions they have had. For example, the number of positive and negative interactions can be used for comparison, and these represent two dimensions in the vector representation. In this case, the cosine similarity is calculated as follows:

$$\begin{aligned} \mathrm{Cosine}(a_\alpha, a_\beta) &= \frac{a_\alpha \cdot a_\beta}{\mid a_\alpha \mid\mid a_\beta \mid} \\ &= \frac{(x_1 \times x_2) + (y_1 \times y_2)}{\sqrt{x_1^2 + y_1^2}\sqrt{x_2^2 + y_2^2}} \end{aligned} \tag{5.2}$$

where (x_1, y_1) and (x_2, y_2) are the two points representing the behaviour of two agents. As the angle between the two vectors decreases, the cosine angle approaches 1, meaning that the similarity increases.

5.6 The Collusion Detection Process

As discussed earlier, the cosine similarity measure is the most suitable measurement based on the data characteristics of the domain area. In this section, we describe the detection process for the Persistent Target-Witness collusion type (PTW). This type of collusion has been chosen as it includes the agent interactions that are observed by the agents using trust and reputation as part of their decision making.

As illustrated in Table 5.1, PTW collusion occurs between a target provider agent and a witness agent. The persistent aspect of the collusion refers to the occurrence of the collusion in terms of its duration, which is throughout the interaction period of the agents involved. The characteristics of this type of collusion can be summarised as follows:

Scope → **Total** : The collusion occurs for all the services provided by the target.

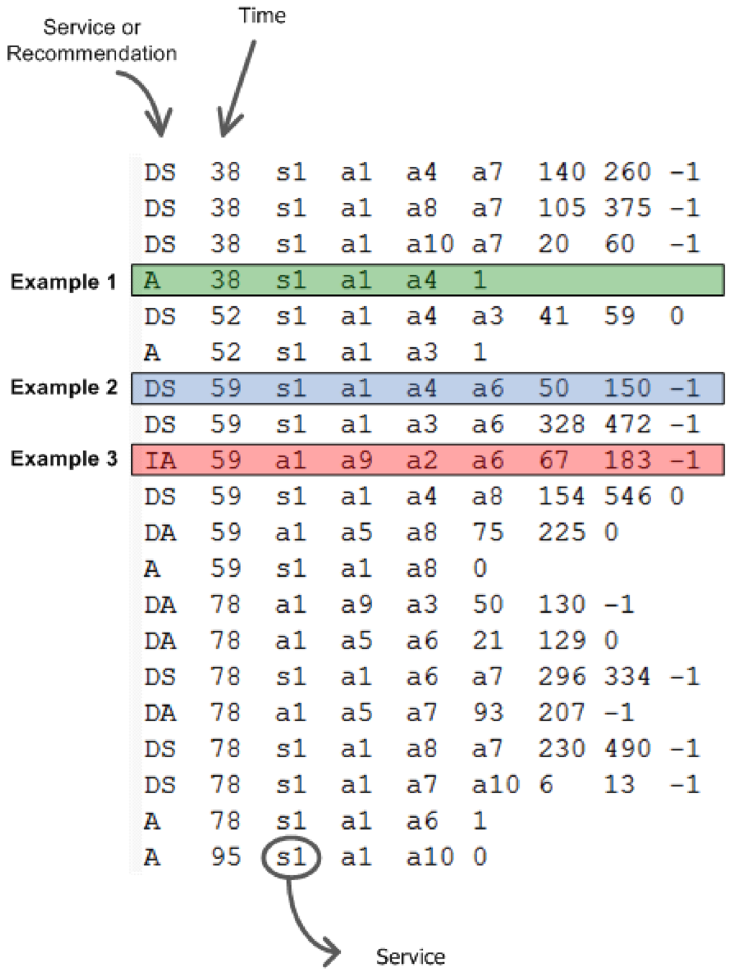
Occurrence → **Duration** → **Persistent** : The collusion lasts throughout the interaction period.

Occurrence → **Cause** → **Situation-driven** : The agents are colluding to bring benefit to the target to increase its reputation.

Impact → **Service characteristics** → **Success dimension** : The recommendation information shared consists of the number of positive and negative interactions as experienced by the witness.

Impact → **Recommendation** → **Direct TW** : The witness has interacted directly with the target and is sharing its own history of interactions.

Three steps in collusion detection are needed to enable collusion detection, namely the recording of interaction histories, the building of agent graphs and finally,



```
DS  38  s1  a1  a4  a7  140 260 -1
DS  38  s1  a1  a8  a7  105 375 -1
DS  38  s1  a1  a10 a7  20  60  -1
A   38  s1  a1  a4  1
DS  52  s1  a1  a4  a3  41  59  0
A   52  s1  a1  a3  1
DS  59  s1  a1  a4  a6  50  150 -1
DS  59  s1  a1  a3  a6  328 472 -1
IA  59  a1  a9  a2  a6  67  183 -1
DS  59  s1  a1  a4  a8  154 546 0
DA  59  a1  a5  a8  75  225 0
A   59  s1  a1  a8  0
DA  78  a1  a9  a3  50  130 -1
DA  78  a1  a5  a6  21  129 0
DS  78  s1  a1  a6  a7  296 334 -1
DA  78  a1  a5  a7  93  207 -1
DS  78  s1  a1  a8  a7  230 490 -1
DS  78  s1  a1  a7  a10 6   13  -1
A   78  s1  a1  a6  1
A   95  s1  a1  a10 0
```

Figure 5.6: Partial Interaction History File Generated For a Population Configuration.

collusion detection from these graphs. An agent records its past interactions with other agents and uses interaction and recommendation information to build a model of its agent environment. In this set of experiments, we focus on how agents use the agent graphs to extract information. Since agents can use any model of agent interaction for decision making, we generate interaction histories for an agent using a trust and reputation model with direct and indirect recommendations. The components of the three components for collusion detection are further described below.

5.6.1 Generation of Interaction Histories

A population configuration defines the parameters of the agent population, such as its size, the trustworthiness composition, the number of services and the number of collusive agents. For each selected population configuration, the agents are created and their interactions are simulated and recorded from the point of view of an evaluator agent. These interactions include direct service interactions, as well as direct and indirect recommendations. In our implementation, the agents are not considered to be using any particular trust and reputation model, but can share their opinions about their own experiences and recommendations.

An example extract of an interaction history file for the evaluating agent is described in Figure 5.6. This file is used for an evaluator to build its agent network graphs and detect collusion. The complete example file has been included in Appendix C.1. In Figure 5.6 each row of the file represents the interactions of an evaluating agent. The tab-separated elements on each row describe the particular interaction in that row. Various types of interactions as previously described in Chapter 3 are shown. For instance, the row identified as **Example 1** is an example of a service interaction (denoted by an A in the line shown) at time unit 38, where provider agent *a4* has executed service type *s1* for customer agent *a1*, and this interaction was successful as denoted by 1 in the last column. **Example 2** depicts a direct service recommendation (denoted by DS in that row) from witness *a2* to evaluator *a1* about the service *s1* for potential provider *a6*. The recommendation provided is made up of 50 successful interactions and 150 failed interactions, however the evaluator has not used this recommendation for its decision making at that point, as denoted by -1 in the last column. The third example, **Example 3** shows an indirect agent recommendation from the secondary witness *a2* to the evaluator *a1*, via the principal witness *a9*. The recommendation of 67 successful interactions and 183 failed interactions concerns the provider *a6*.

5.6.2 Agent Graph Building

An evaluator builds agent network graphs to represent its environment through service-oriented graphs for agents providing a particular service, and agent-oriented graphs that show the overall performance of the agents. Provider graphs gather information about service providers, while recommender graphs represent the opinions shared by witnesses. These graphs are built from the history of past interactions as recorded by the evaluator, as described in Chapter 4.

5.6.3 Collusion Detection

The agent network models maintained by an evaluator enables it to access valuable information about potential providers of services and recommendations. Collusion detection is an example of the result of such information extraction and it arises from analysing the agent graphs and uncovering previously unknown information about collusive agents. In our experiments, a list of actual collusive pairs of agents is compared to the list of pairs that the evaluator detects.

As part of the analysis of the information extracted from an agent's interaction and recommendation history to detect collusion, Algorithm 5.1 outlines the partial collusion detection process after target a_β has just provided service s_β following recommendations. Initially, the set of potential colluders will include all the direct recommenders for target a_β about the service s_β. This set then needs to undergo further selection to ultimately obtain the smallest group of potential colluders. Based on this information, the evaluator can decide on subsequent interactions with the members of the suspected collusive group.

Algorithm 5.1 Partial Witness and Target Collusion Detection

```
for all direct recommendations $r^d$ do
    if $(r^d.a_t = a_\beta)$ AND $(r^d.s = s_\beta)$ then
        for all dimensions $d \in r^d.s$ do
            if $d_a < d_e$ then
                add $a_r$ to $\mathbb{P}\ colluders$
            endif
        endfor
    endif
endfor
```

More specifically, for the detection of Target-Witness collusion using the cosine similarity measure, Algorithm 5.2 outlines how the agent graphs are used to extract relevant information.

Algorithm 5.2 Target-Witness Collusion Detection using Cosine Similarity

```
for each service $s$ in serviceProviderGraph do
    providerSet ← FINDPROVIDERS($a_e$, $s$)
    customerSet ← FINDCUSTOMERWITNESSES($a_e$, $a_p$, providerSet)
    if COMPARESERVICE($a_e$, $a_p$, $a_r$, $s$) then
        collusivePairs ← collusivePairs + pair($a_p$, $a_r$)
    endif
endfor
```

The evaluator a_e maintains a graph structure for providers of each service type, referred to as `serviceProviderGraph`. For each service s, `providerSet` represents all the providers of that service, as experienced by the evaluator. As well as direct interactions, the evaluator may also have received recommendations about providers, potentially along a recommendation chain, and the last witness in the chain is a customer of the provider and these customers are stored in `customerSet`. Then, the evaluator's own direct experience with interacting with a provider is compared with a witness' experience through the COMPARESERVICE function. In this case the function makes use of the cosine similarity measurement to compare interactions. Based on this, the pair of agents (provider and witness) may be added to the list of possible collusive pairs for target-witness collusion.

5.7 Evaluating Collusion Detection

This section describes the experiments we have set up to evaluate the detection of Persistent Target-Witness collusion using the cosine similarity measurement approach. Following the description of the setup, we describe and discuss the experimental results.

5.7.1 Experimental Setup

For the evaluation of Persistent Target-Witness collusion detection, we first describe the experimental setup, which includes the composition of the agent population, the usage of the cosine similarity measurement and the experimental results that are recorded.

5.7.1.1 Agent Population Parameters

In our experiments, the agent population parameters are varied in order to generate a comprehensive range of configurations for a representative population with agents of heterogeneous behaviours. 8,778 population configurations have been used (i.e. all the possible population configurations that are generated) and for each configuration we have obtained the average of 10 runs of the experiment. The population parameters are as follows.

1. Population size (*PopulationSize*) indicates the total number of agents in the population. In our experiments, the size ranges from 5 to 200 agents in varying steps. The set of population sizes used is 5,10,20,30,40,50,70,90,200.

2. Trust configuration determines the proportion of the population with one of three trust characteristics. The three proportions add up to 100% of the population size.

    - High trust suggests that an agent is likely to provide the expected outcome between 80% and 100% of the time.
    - Average trust indicates that an agent is 40% to 80% likely to provide the expected result.
    - Low trust agents are likely to match their expected outcome between 0% and 40% of the time.

3. Number of services available *ServiceCount*, from $(0.25 * PopulationSize)$ to $(0.75 * PopulationSize)$ in steps based on Algorithm 5.3. These steps ensure that the number of services within the population of agents is proportional to the population size.

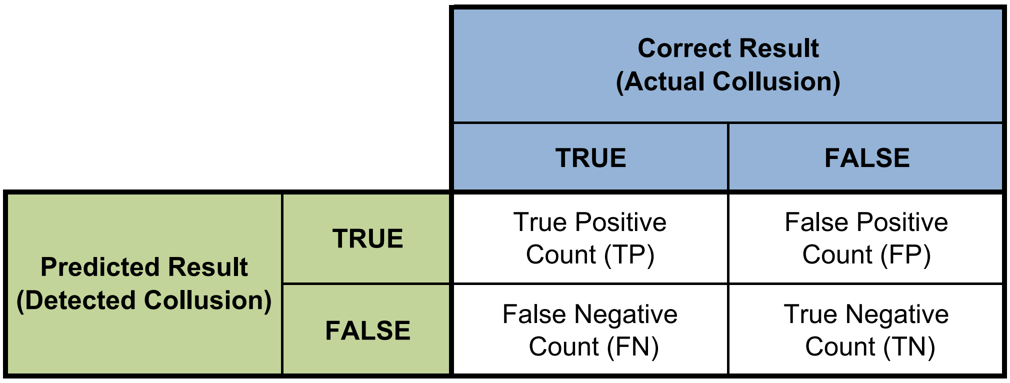
Algorithm 5.3 Incremental Steps of the Service Count

```
if ServiceCount <10 then ServiceCount += 5
else if ServiceCount <50 then ServiceCount += 10
else if ServiceCount <100 then ServiceCount += 20
else if ServiceCount <250 then ServiceCount += 100
else if ServiceCount >250 then ServiceCount += 200
```

4. Number of collusive pairs, ranging from 0 to $(0.5 * PopulationSize)$, with values from 1 increasing in steps similar to Algorithm 5.3. This range of values has been chosen such that the maximum number of collusive pairs possible is $(PopulationSize - 1) * (PopulationSize - 2)$, which can be very large for bigger populations.

5.7.1.2 Applying Cosine Similarity Measurement

As described in Section 5.5.3, two agents have increasingly similar experiences when the cosine angle between their behaviour vectors approaches 1. Conversely, the cosine angle approaches 0 when the agents have dissimilar experiences. The evaluator needs to decide on the point at which agent experiences are dissimilar enough to consider potential collusion. One approach is to use a threshold value, $CollusionThreshold$ such that, if $Cosine(a_\alpha, a_\beta) \leq CollusionThreshold$, then, the pair of agents (a_α, a_β) belongs to the list of potentially collusive pairs as detected by the evaluator, $DetectedCollusion$. A cosine similarity value greater than the threshold will consider then agent pair to be non-collusive. As a result of empirical experiments we have placed the threshold between collusion and non-collusion at a cosine similarity value of 0.75. This means that cosine similarity values of less than the threshold result in the agent detecting collusion. Cosine similarity measures range from 0 to 1, with 0 meaning complete similarity, and 1 meaning complete dissimilarity. We discuss the choice of other cosine similarity thresholds later in this chapter (Section 5.7.2.4).



		Correct Result (Actual Collusion)	
		TRUE	FALSE
Predicted Result (Detected Collusion)	TRUE	True Positive Count (TP)	False Positive Count (FP)
	FALSE	False Negative Count (FN)	True Negative Count (TN)

Figure 5.7: Classification Matrix.

5.7.1.3 Metrics for Collusion Detection

Each interaction history is generated based on a time length of 2000 units (in each unit of time, agents can execute part of a service task). After the agent graphs are built and analysed for collusion, the experimental results are stored. For each population configuration we have obtained an average from 10 runs.

The aim of these experiments is to assess whether an agent can accurately identify collusive pairs of agents from the population of agents within which it evolves. The actual list of collusive agent pairs is denoted as $ActualCollusionList$ and the list of detected agent pairs by the evaluator is referred to as $DetectedCollusionList$. The 4 types of outcomes can be represented as a classification matrix [93], adapted to the collusion detection process, as shown in Figure 5.7. The correct decisions made are represented by the numbers along the diagonal from upper-left to lower-right. The other two numbers are the errors in collusion detection.

- **True positive count** (TP) is the number of collusive agent pairs correctly detected, such that $TP = ActualCollusionList \cap DetectedCollusionList$
- **False positive count** (FP) is the number of non-collusive agent pairs wrongly detected, such that $FP = DetectedCollusionList \setminus ActualCollusionList$
- **False negative count** (FN) is the number of collusive agent pairs not detected, such that $FN = ActualCollusionList \setminus DetectedCollusionList$
- **True negative count** (TN) is the number of non-collusive agent pairs correctly not detected, such that $TN = (ActualCollusionList \cap DetectedCollusionList)'$

Two commonly used performance measures are precision and recall. We use these two performance measures to assess our approach to collusion detection using information gathered from agent interactions.

- **Precision** is the proportion of correctly detected pairs out of all the pairs identified by the evaluator, defined as:

$$Precision(P) = \frac{TP}{TP + FP}$$

- **Recall** is the proportion of correctly detected pairs out of all the actual collusive pairs in the population, defined as:

$$Recall(R) = \frac{TP}{TP + FN}$$

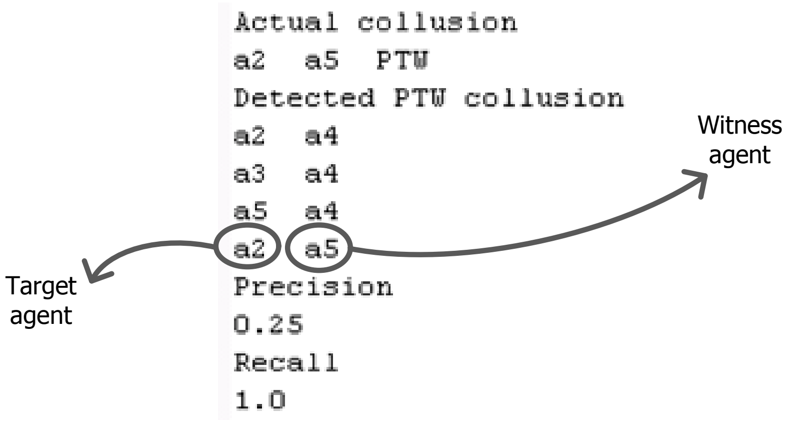
5.7.1.4 Experimental Approach

After the collusion detection process, the interaction history file (see Figure 5.6) is appended to add the collusion detection information, as shown in Figure 5.8 to illustrate the collusion detection information for a particular population configuration run. In this example, there is one pair of collusive agents for the PTW collusion type, between potential target agent *a2* and witness *a5*. The number of agent pairs in the actual collusion list is denoted as *ActualCollusionCount*. The evaluator has detected 4 pairs of agents, one of which is the correct pair. The number of agent pairs detected as collusive by the evaluator is referred to as *DetectedCollusionCount*.

Each population configuration is run 10 times and the results are averaged over the 10 runs. The collusion detection results file contains the following elements per population configuration (row).

1. The six population characteristics described in Section 5.7.1.1.

2. Average true positive count (average number of detected collusive pairs that are actually collusive over the 10 runs).



```
Actual collusion
a2  a5  PTW
Detected PTW collusion
a2  a4
a3  a4
a5  a4
a2  a5
Precision
0.25
Recall
1.0
```

Figure 5.8: Partial Interaction History File with Collusion Detection Information.

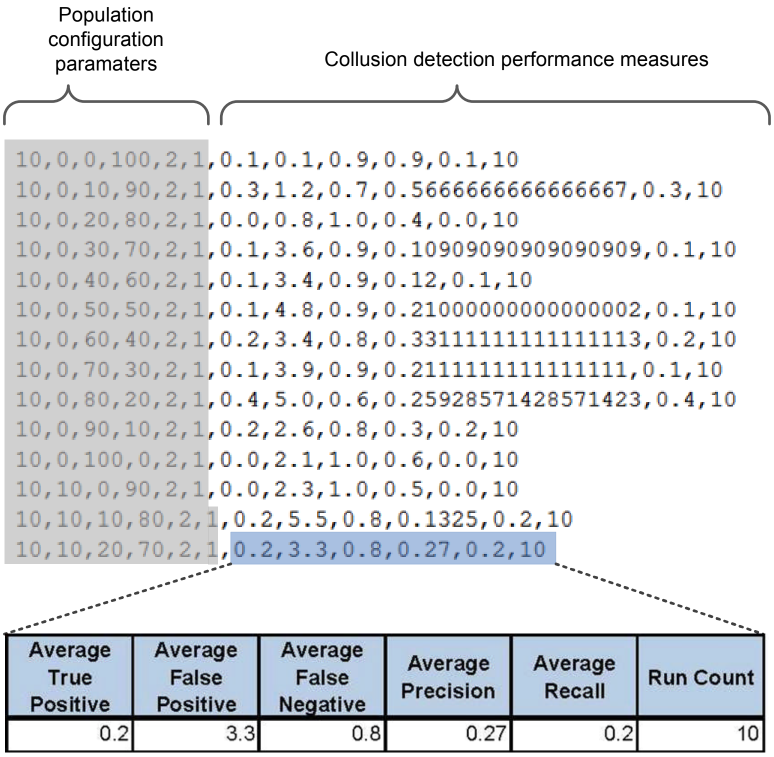
3. Average false positive count (average number of non-collusive pairs wrongly detected as collusive).

4. Average false negative count (average number of collusive pairs that have not been detected as collusive).

5. Average precision over 10 runs.

6. Average recall over 10 runs.

An example of a partial output file for one population configuration of size 10 is shown in Figure 5.9 to illustrate the values recorded. The complete output file is given in Appendix C.2. Figure 5.9 divides the elements in two groups. The first group consists of the population configuration parameters (first 6 columns) shown by the grey box. The remaining elements concern the collusion detection measures. As an illustration, the elements of this group are annotated in Figure 5.9 for the population configuration in the last row.


Population configuration paramaters
Collusion detection performance measures

```
10,0,0,100,2,1,0.1,0.1,0.9,0.9,0.1,10
10,0,10,90,2,1,0.3,1.2,0.7,0.5666666666666667,0.3,10
10,0,20,80,2,1,0.0,0.8,1.0,0.4,0.0,10
10,0,30,70,2,1,0.1,3.6,0.9,0.10909090909090909,0.1,10
10,0,40,60,2,1,0.1,3.4,0.9,0.12,0.1,10
10,0,50,50,2,1,0.1,4.8,0.9,0.21000000000000002,0.1,10
10,0,60,40,2,1,0.2,3.4,0.8,0.33111111111111113,0.2,10
10,0,70,30,2,1,0.1,3.9,0.9,0.2111111111111111,0.1,10
10,0,80,20,2,1,0.4,5.0,0.6,0.25928571428571423,0.4,10
10,0,90,10,2,1,0.2,2.6,0.8,0.3,0.2,10
10,0,100,0,2,1,0.0,2.1,1.0,0.6,0.0,10
10,10,0,90,2,1,0.0,2.3,1.0,0.5,0.0,10
10,10,10,80,2,1,0.2,5.5,0.8,0.1325,0.2,10
10,10,20,70,2,1,0.2,3.3,0.8,0.27,0.2,10
```

Average True Positive	Average False Positive	Average False Negative	Average Precision	Average Recall	Run Count
0.2	3.3	0.8	0.27	0.2	10

Figure 5.9: Partial Result File For a Population Configuration With 10 Agents.

5.7.2 Experimental Results

We evaluate the cosine similarity collusion detection approach, described in Section 5.5.2.2, for the Persistent Target-Witness collusion type. The aim is to assess the collusion detection performance of an agent using our trust-informed approach. For PTW collusion detection, an evaluating agent can identify the pairs of potential providers (targets) and witnesses involved. The complete set of results, a sample of which is shown in Figure 5.9, includes 8,778 rows of values (i.e. all configurations of the population parameters). Five trust configurations have been selected to illustrate the spectrum of trustworthiness that a population can exhibit. A trust configuration consists of a triple in the form (High,Average,Low) to represent the proportions of agents in the population with these behaviours.

- (0,100,0) (100% of the the agents have average trust).
- (10,0,90) (10% of the population have high trust and 90% have low trust).
- (10,70,20) (10% of the population have high trust, 70% average and 20% low trust).
- (30,40,30) (30% of the population have high trust, 40% have average trust, while 30% have low trust).
- (90,0,10) (90% of the population have high trust and 10% have low trust).

5.7.2.1 The Effect of Population Size

Precision and recall values for each of the five trust configurations have been plotted against population size, as shown in Figure 5.10 and Figure 5.12 respectively. These graphs are presented to show the relationship between precision and recall of collusion detection with the population size. The statistical software package PASW Statistics (SPSS Statistics) 18 has been used to draw the graphs and process the results.

Figure 5.10 shows that the average precision for the five trust configurations is higher for smaller and larger population sizes, while precision is lower for medium-sized populations of 20 to 50 agents. The maximum precision for the most efficient collusion detection is 1 and this value is reached in two ways. Firstly, when there is no collusion in the population ($TP = 0$) and no non-collusive agents have been wrongly detected ($FP = 0$). Secondly, precision is 1 when the agent has correctly detected all the collusive agent pairs and only the actual collusive pairs ($TP = ActualCollusionCount$, $TP > 0$ and $FP = 0$). From Figure 5.10, the precision is higher for two trust configurations, namely (0,100,0) and (10,0,90). The proportion of agents in average or low

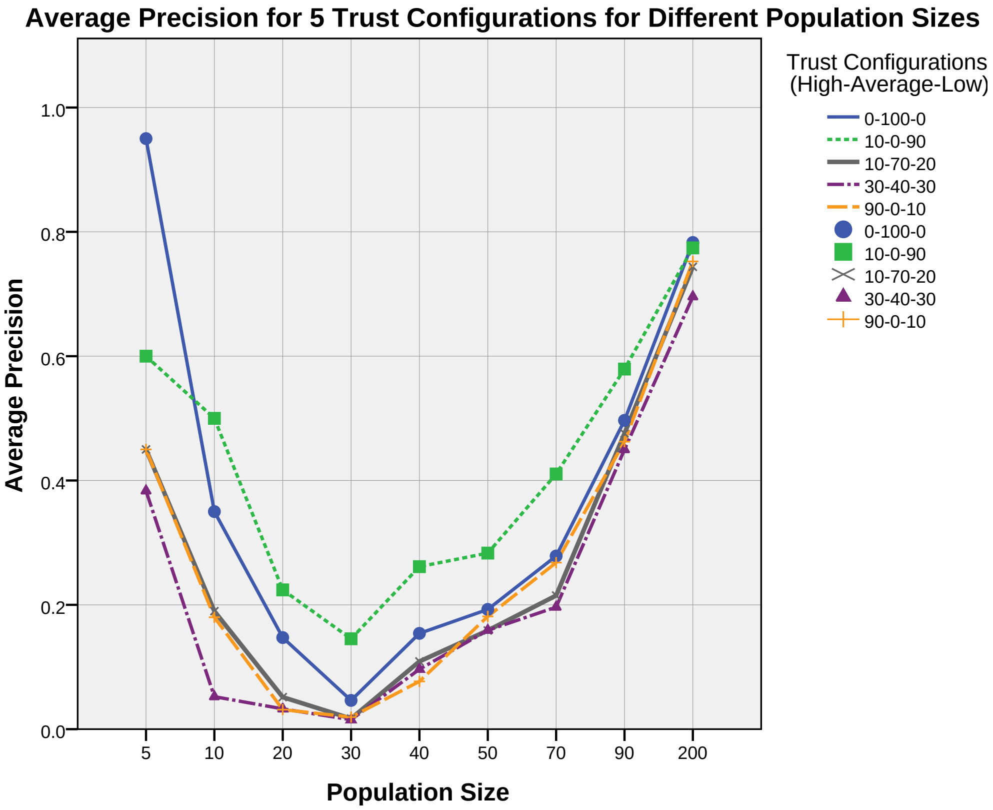


Figure 5.10: Average Precision with Increase in Population Size.

trust is higher in both these trust configurations, which suggests that an evaluator has higher probability of detecting collusion efficiently for trust configurations with lower proportions of high trust.

Each line on Figure 5.10 is an average of all the individual population configurations for that particular trust configuration at various population sizes. Let us consider one of those lines — trust configuration (0,100,0) — to further explore the values that are represented. Figure 5.11 shows a boxplot for the range of values that precision can have. The mean precision for the trust configuration (0,100,0) is larger than the mean recall by 0.407 (Table 5.3).

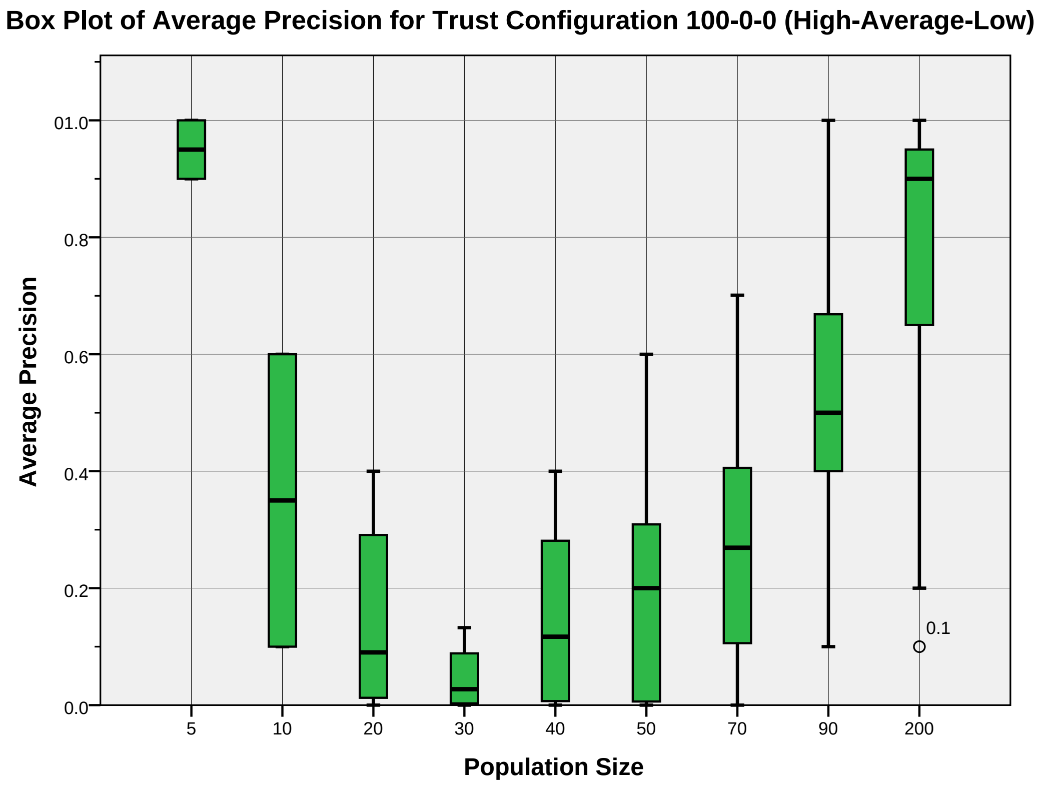
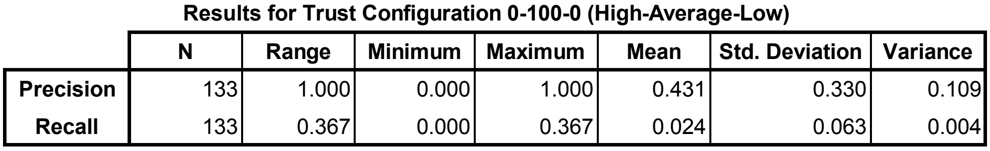


Figure 5.11: Boxplot of Precision for Trust Configuration (0,100,0).

A box plot is a graphical display that simultaneously describes several important features of the data set, such as the minimum, first quartile, median, third quartile and maximum values for the precision values at each population size. Additionally, it displays the distribution of the precision variable, with the T-bars (whiskers) extending from the box to show where approximately 95% of the data lies for a normal distribution. Outliers are denoted by circles and extreme outliers are shown as crosses, and they show cases where the data is beyond a whisker but is within three times the height of the box or more than 3 times the interquartile range from the box respectively [86].

Figure 5.12 shows the average recall for the five trust configurations for different population sizes. The average recall is higher for the small population sizes and then



Results for Trust Configuration 0-100-0 (High-Average-Low)

	N	Range	Minimum	Maximum	Mean	Std. Deviation	Variance
Precision	133	1.000	0.000	1.000	0.431	0.330	0.109
Recall	133	0.367	0.000	0.367	0.024	0.063	0.004

Table 5.3: Precision and Recall Results for Trust Configuration (0,100,0).

decreases with larger population sizes. The maximum recall for the widest breadth of accurate collusion detection is 1 and this value is also reached in two ways. The first case occurs when there is no collusion in the agent population ($TP = 0$) and hence there are no collusive pairs to be correctly detected by the evaluator ($FN = 0$). As division by 0 is undefined, the largest value that recall can have is 1. The second case arises when the evaluator accurately detects all the collusive agent pairs ($TP = ActualCollusionCount$ and $FN = 0$). In the graph (Figure 5.12), as the population size increases, the evaluator is less able to detect only the relevant collusive agent pairs. Even if it detects the actual collusive pairs, it is also detecting many other agent pairs incorrectly as being collusive. Of the five trust configurations, (10-0-90) performs slightly worse than the other configurations for small to medium population sizes. This trust configuration consists of a higher proportion of low trust agents (90% of the population). Therefore, the evaluator performs less well on collusion detection recall when a higher proportion of the population are less trustworthy.

5.7.2.2 The Impact of the Extent of Collusion

Next, we analysed the impact of the extent of collusion by looking at average precision and recall of collusion detection by the evaluator agent $a1$ with respect to an increase in the number of collusive agent pairs in the population. Again, the best scores for average precision and recall over 10 runs is $P = 1$ and $R = 1$.

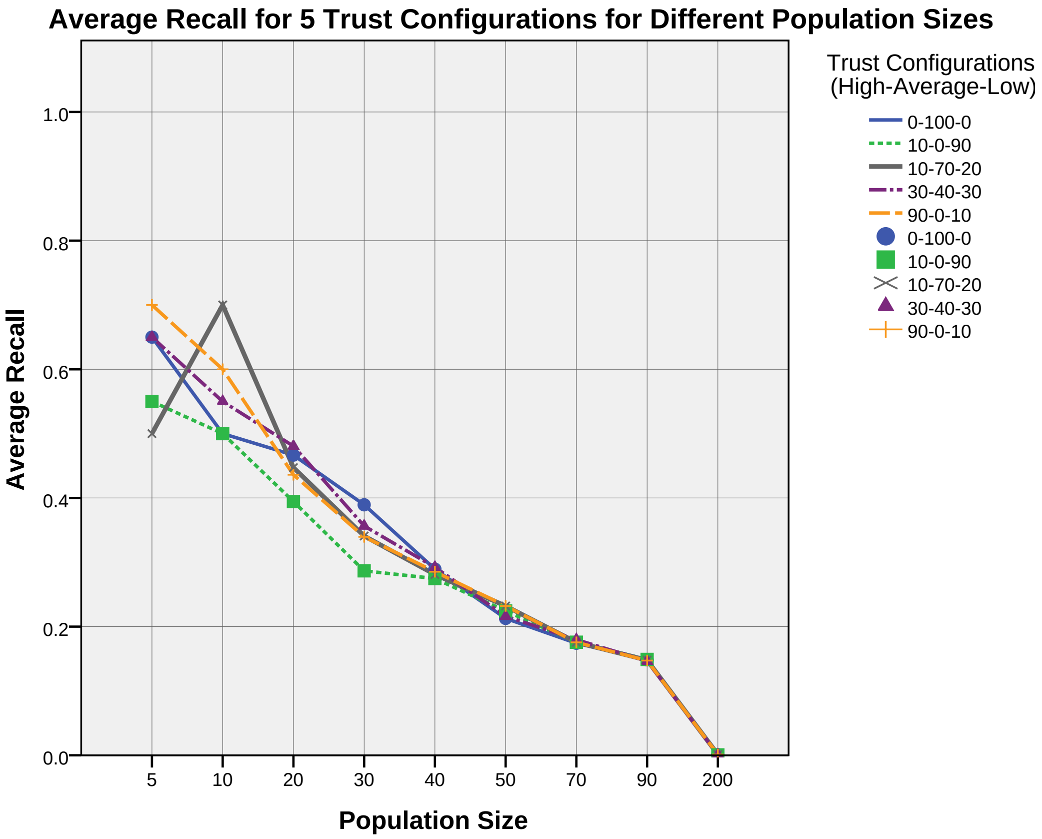


Figure 5.12: Average Recall with Increase in Population Size.

Figure 5.13 shows the average precision with an increasing number of collusive pairs for each of the five trust configurations. The average precision increases as the number of collusive agent pairs increases in the population. As the experiments have been set up such that the range of collusive pairs that can be generated reflects the population size, it is not surprising that the precision increases with a larger number of collusive pairs. This is due to the number of possible collusive pairs in the population increasing quadratically with the population size, as shown by the quadratic graph in Figure 5.14. This function applied for population sizes greater than 1, since we need at least two agents for a pair and for population size 2, the number of possible collusive

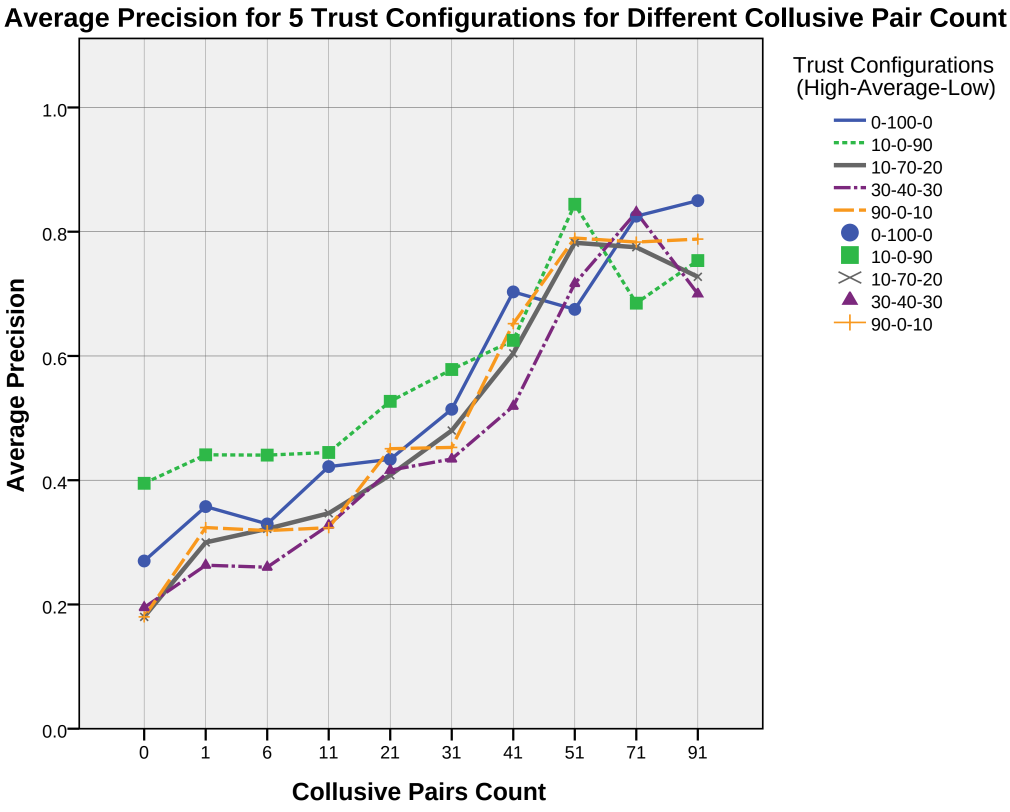


Figure 5.13: Average Precision with Increase in Collusive Agent Pairs.

pairs is 0 as the evaluator is neither a target nor a witness.

We next consider how the average recall of collusion detection varies with respect to the number of collusive agent pairs in the population. Figure 5.15 presents the plots for the five trust configurations. The average recall for all five trust configurations is low for all the collusive pair counts when there is at least one collusive pair, with the maximum average recall being 0.058 for one collusive pair. The low recall values suggest that the evaluator is not detecting a high proportion of the actual collusive pairs. As with the average precision shown in Figure 5.13, the number of collusive pairs is proportional to the population size and as the population increases, the number of possible collusive

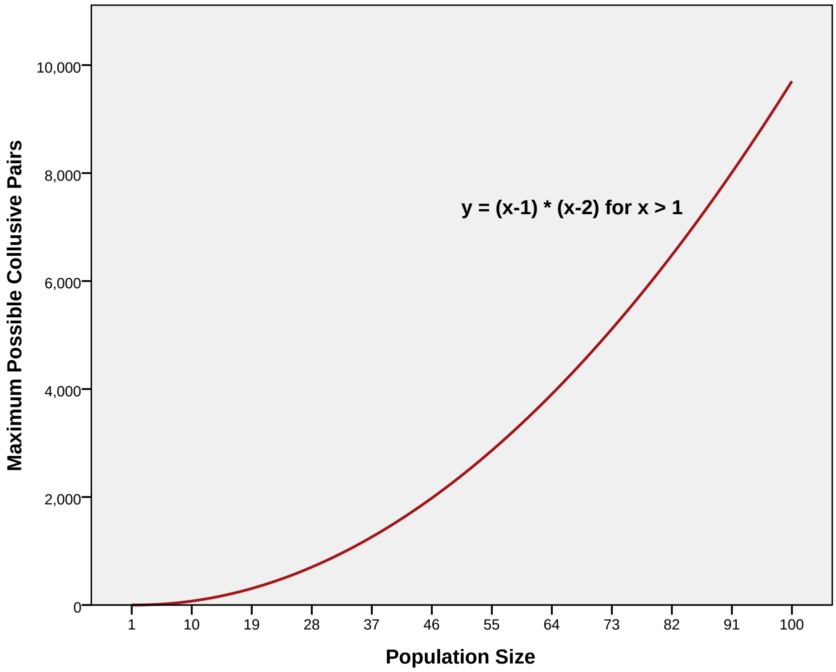


Figure 5.14: Plot of the Quadratic Function $y = x^2 - 3x + 2$, for $x > 1$.

pairs increases quadratically.

5.7.2.3 Balancing Precision and Recall

High values for both precision and recall for collusion detection indicate that the collusion detection mechanism is performing as desired, such that an evaluator detects most of the collusive agents in the population and not the non-collusive agents. From the experiments carried out and described in the previous sections, the precision and recall for the experimental setup and collusion detection technique vary significantly and recall especially is relatively low. We now analyse for which population configurations both the precision and recall are sufficiently high for accurate collusion detection. We are assuming in this analysis that both precision and recall have equal weight in collusion detection. In certain cases, the balance between precision and recall can be different.

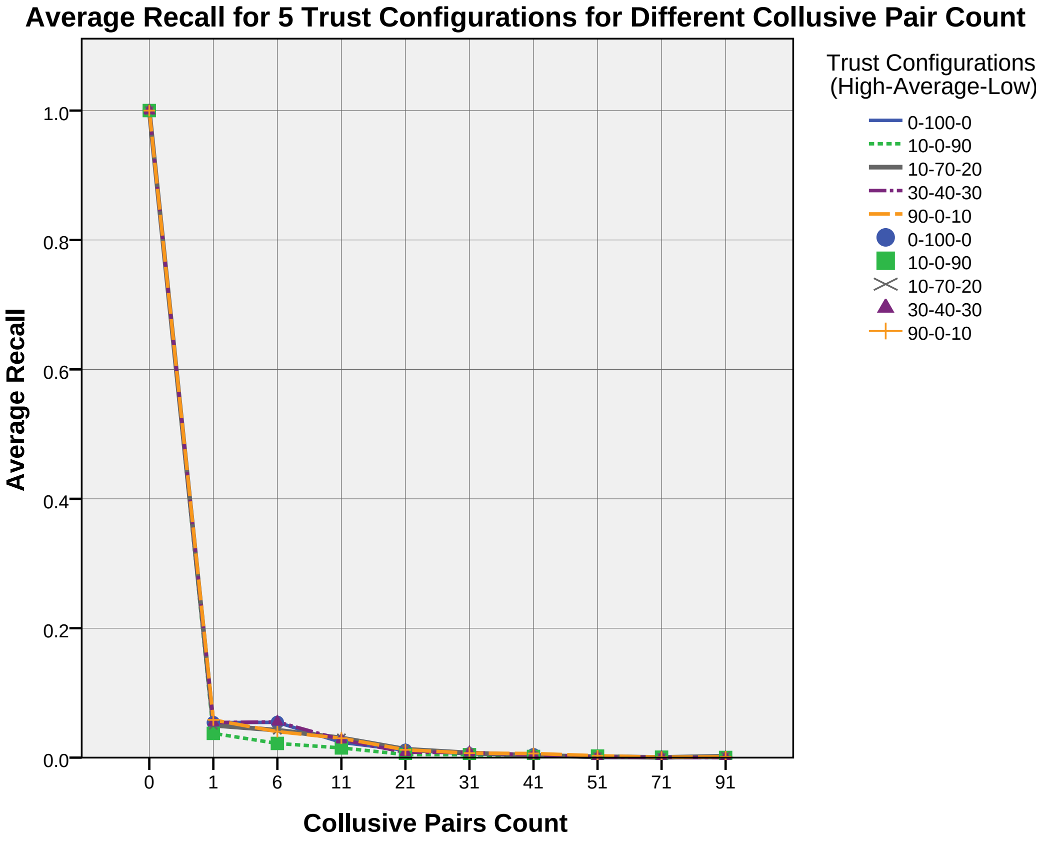
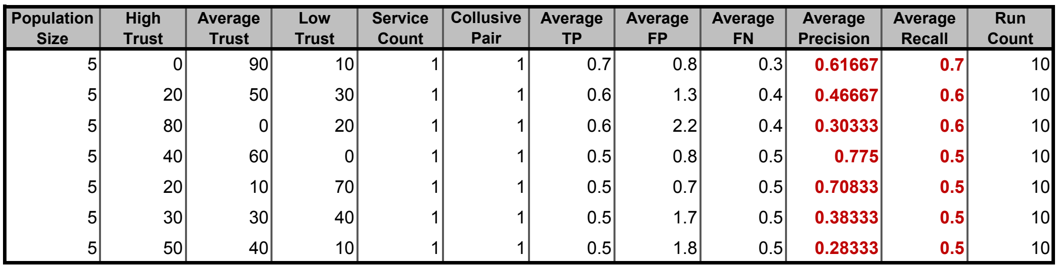


Figure 5.15: Average Recall with Increase in Collusive Agent Pairs.

For instance, in a domain where an evaluator needs to detect relevant collusive agents in as few searches as possible, the precision needs to be high. This means that a high proportion of the detected pairs are among the correct collusive pairs in the population. However, in a more critical situation where the collusive agent pairs need to be found, the recall needs to be high, such that a high proportion of the actual collusive pairs are found by the agent.

Table 5.4 presents the population configurations containing collusion when both the precision and recall are ≥ 0.25. Note that the number of collusive pairs is ≥ 1 because we want analyse the precision and recall when there is collusion. We can



Population Size	High Trust	Average Trust	Low Trust	Service Count	Collusive Pair	Average TP	Average FP	Average FN	Average Precision	Average Recall	Run Count
5	0	90	10	1	1	0.7	0.8	0.3	**0.61667**	**0.7**	10
5	20	50	30	1	1	0.6	1.3	0.4	**0.46667**	**0.6**	10
5	80	0	20	1	1	0.6	2.2	0.4	**0.30333**	**0.6**	10
5	40	60	0	1	1	0.5	0.8	0.5	**0.775**	**0.5**	10
5	20	10	70	1	1	0.5	0.7	0.5	**0.70833**	**0.5**	10
5	30	30	40	1	1	0.5	1.7	0.5	**0.38333**	**0.5**	10
5	50	40	10	1	1	0.5	1.8	0.5	**0.28333**	**0.5**	10

Table 5.4: Population Configurations where $(P \geq 0.25)$ *AND* $(R \geq 0.25)$

observe that the population configurations in this set involve small populations sizes, as well as small numbers of collusive pairs. Larger populations, an increasing number of services, and an increasing number of collusive pairs lead to lower precision and recall. While some population configurations achieve higher precision or higher recall, both high precision and high recall are not achieved together in these configurations.

5.7.2.4 The Effect of the Cosine Similarity Measurement Threshold

In the previous results we used a cosine similarity measurement threshold of 0.75 in our experiments, determined by empirical evaluation, as discussed in Section 5.7.1.2. For the purposes of comparison and validation of this threshold we have also run another set of experiments with other threshold values to assess whether there is any statistical significance among these values and if so, which values are more suitable. Our aim is to determine whether or not the precision and recall of collusion detection using a cosine similarity threshold value of 0.75 is different from the performance measures obtained for other threshold values.

Experimental Setup and Approach to Testing Significance

Three threshold values of 0.25, 0.50, 0.95 are compared to the threshold of 0.75 used in our previous experiments. The population configuration has been set up with the following parameters, with only the cosine similarity threshold being different in each set of runs.

- Trust configurations (High,Average,Low): (0,100,0), (10,0,90), (10,70,20), (30,40,30), (90,0,10).
- Service count: 5.
- Collusive pair count: 2.
- Number of runs for each population configuration: 5.
- Population size: 5, 10, 15, 20, 25, 30, 35, 40, 45, 50.

To measure the statistical significance of the differences in precision and recall when using different cosine similarity thresholds, paired *t*-tests can be used with pairs of thresholds. The paired *t*-test is appropriate as we use the following assumptions.

- The value pairs are independent: the experiments have been set up to run with the same configurations, with only the threshold changing and the two sets of experiments giving the two sets of values are run separately.
- The sample data is drawn from a normal population. We can reasonably make this assumption as the population configuration selected for this set of experiments is representative of the population. The *t*-test also performs well when this assumption is violated [44].

The test procedure involves analysing the differences between precision and recall using each cosine similarity threshold value. If there is no difference between the performance measures (precision and recall), then the mean of the differences should be 0.

Let (X_{11}, X_{21}), (X_{12}, X_{22}), $\ldots$, (X_{1n}, X_{2n}) be a set of n pairs where we assume that the mean and variance of the population X_1 are μ_1 and σ_1^2, and the mean and and variance of the population X_2 are μ_2 and σ_2^2. The difference between each pair of performance measures is defined as $D_j = X_{1j} - X_{2j}$, where $j = 1, 2, \ldots, n$. A hypothesis for the threshold T takes the following form.

$H_{T\alpha z}$: $\mu_D = \mu_{Mt_1} - \mu_{Mt_2} = 0$

where α is the hypothesis identifier, z is 0 or 1 for null or alternative hypothesis and M is the performance measure (precision or recall). Each population X_{Pt} represents the set of precision values for when the threshold is value t, and population X_{Rt} represents the set of recall values with threshold t. Our list of null hypotheses for precision are as follows.

H_{Ta0}: $\mu_D = \mu_{P0.75} - \mu_{P0.05} = 0$

H_{Tb0}: $\mu_D = \mu_{P0.75} - \mu_{P0.25} = 0$

H_{Tc0}: $\mu_D = \mu_{P0.75} - \mu_{P0.5} = 0$

H_{Td0}: $\mu_D = \mu_{P0.75} - \mu_{P0.95} = 0$

Similarly, our null hypotheses for recall are listed below.

H_{Te0}: $\mu_D = \mu_{R0.75} - \mu_{R0.05} = 0$

H_{Tf0}: $\mu_D = \mu_{R0.75} - \mu_{R0.25} = 0$

H_{Tg0}: $\mu_D = \mu_{R0.75} - \mu_{R0.5} = 0$

H_{Th0}: $\mu_D = \mu_{R0.75} - \mu_{R0.95} = 0$

The alternative hypotheses for precision and recall can also be described and the validity of those hypotheses would indicate that there is a difference between performance measures when different cosine similarity thresholds are used. The following is the set of hypotheses that we will be investigating.

H_{Ta1}: $\mu_D = \mu_{P0.75} - \mu_{P0.05} \neq 0$

H_{Tb1}: $\mu_D = \mu_{P0.75} - \mu_{P0.25} \neq 0$

H_{Tc1}: $\mu_D = \mu_{P0.75} - \mu_{P0.5} \neq 0$

H_{Td1}: $\mu_D = \mu_{P0.75} - \mu_{P0.95} \neq 0$

H_{Te1}: $\mu_D = \mu_{R0.75} - \mu_{R0.05} \neq 0$

H_{Tf1}: $\mu_D = \mu_{R0.75} - \mu_{R0.25} \neq 0$

H_{Tg1}: $\mu_D = \mu_{R0.75} - \mu_{R0.5} \neq 0$

H_{Th1}: $\mu_D = \mu_{R0.75} - \mu_{R0.95} \neq 0$

Results and Discussion

Tables 5.5 and 5.6 show the hypotheses results for precision and recall of collusion detection respectively. In the tables, the t-statistic value is the standardised sample mean, the degrees of freedom (df) is the number of independent comparisons made from the sample, while the mean value is the average difference between performance measures with two different cosine similarity thresholds. Finally, the P-value is the probability of obtaining the t-statistic whose absolute value is equal to or greater than the t-statistic value obtained. As we assume a confidence of 95%, a significance value for a difference in the use of two threshold is less than 0.05. Complete results are shown in Appendices C.3 and C.4

Hypothesis	t-statistic	df	Mean	P
H_{Ta1}	-8.137	49	-0.402	0.000
H_{Tb1}	-5.339	49	-0.192	0.000
H_{Tc1}	-2.425	49	-0.066	0.019
H_{Td1}	1.016	49	0.020	0.315

Table 5.5: Hypotheses Results for Precision

From Table 5.5, we can observe that for hypotheses H_{Ta1}, H_{Tb1} and H_{Tc1}, $P < 0.05$, which suggests that we can reject the null hypotheses H_{Ta0}, H_{Tb0} and H_{Tc0}. Thus, we can conclude that there is a difference in precision between using a cosine similarity threshold of 0.75 and thresholds 0.05, 0.25 and 0.5. However, there is no significant difference in precision when using a threshold of 0.75 and one of 0.95. Despite there being a significance for three of the thresholds, they all resulted in higher precision than with threshold 0.75. However, we should also consider recall in the choice of threshold.

Hypothesis	t-statistic	df	Mean	P
H_{Te1}	5.429	49	0.156	0.000
H_{Tf1}	4.543	49	0.136	0.000
H_{Tg1}	2.794	49	0.092	0.007
H_{Th1}	-4.128	49	-0.184	0.000

Table 5.6: Hypotheses Results for Recall

The results of the hypothesis testing in Table 5.6 show that for all the four hypotheses, $P < 0.05$, which indicates that as we reject the null hypotheses, there is a significant difference between recall values when one of the four cosine similarity thresholds 0.05, 0.25, 0.5 and 0.95 are used compared to the recall values using a threshold value of 0.75. Of these four thresholds, only the value of 0.95 resulted in a higher recall than for threshold 0.75.

As we can see from the above results, precision and recall are conflicting. A lower cosine similarity threshold favours higher precision of collusion detection, while a higher cosine similarity threshold is needed for a higher recall for this population configuration set. To achieve both high precision and recall, we need to find the right balance for the cosine similarity threshold. In light of the hypothesis testing results, we consider that the choice of the threshold value 0.75 is appropriate for our investigation, as it results in both precision and recall to be >0.1, despite the overall low recall values. If a better precision is needed at the expense of recall, then a lower threshold could be used.

5.7.2.5 Summary of Results

The results show that the precision of collusion detection is higher for small and large populations, while recall decreases with population size. A similar trend has been observed for the extent of collusion in the agent population. Precision of detection increases with an increase in collusion agents, while recall decreases. As both precision and recall need to be high for effective collusion detection, based on our results, we have observed that population configurations that achieve both a precision and recall of ≥ 0.25 consist of small population sizes and small number of collusive agents.

We have carried out the experiments based on a cosine similarity measurement threshold of 0.75, determined by empirical evaluation. Since the cosine similarity measurement threshold is determinant in the calculation of precision and recall, we have also evaluated precision and recall of collusion detection for several other threshold values.

The statistical significance tests show that precision and recall are conflicting — a lower threshold gives higher precision, while a higher threshold results in a higher precision. From our results, we have shown that a threshold of 0.75 is appropriate for balancing precision and recall as it allows both values to be > 0.1.

5.8 Conclusions

In this chapter, we have addressed some of the issues of collusion detection in agent-based systems. Firstly, we proposed a new taxonomy of collusion to classify the relevant aspects of collusion in agent-based systems. The taxonomy has been inspired by the collusion characteristics identified by Smed *et al.* [117, 118] for the field of online gaming. With reference to the taxonomy, we have also outlined several types of collusion that can occur among heterogeneous, communicating agents in such domains as e-commerce and e-supply chains. We have discussed various collusion detection techniques that can be used to identify potential collusive pairs and collusive sets and the choice of cosine similarity measurement as the most appropriate to use for the domain under consideration.

The evaluation of the PTW collusion detection using cosine similarity highlighted the potential of this technique for collusion detection. However, precision and recall values were relatively low, suggesting that many combined factors in the population configuration may be affecting the performance of the evaluating agent. Low recall is the result of a large false positive count in many cases, that is, the evaluator is wrongly detecting non-collusive agents. In our current implementation, there is no upper limit for the number of collusive pairs that the evaluator can detect. Limiting the number of pairs detected and ensuring that the most relevant pairs are kept in the collusion list could increase the recall for collusion detection.

In Section 5.7.2.4, we performed an experiment to test the significance of the cosine similarity threshold that has been used to differentiate the possibility of existence or absence of collusion. The assumption was that we want to maximise both the precision and recall of collusion detection. The findings indicate that precision and recall are conflicting and there is a need to choose a suitable cosine similarity threshold to balance them. Further work is necessary to discover whether a different approach to the threshold, such as a fuzzy logic would help in achieving this balance.

In this work, we have shown the detection of one type of collusion, namely the Persistent Target-Witness collusion. For some population configurations, the detection works well and the evaluator agent benefits from being able to identify the target-witness pairs that are behaving in a similar way as in PTW collusion. Some of the interesting challenges that need further research include the identification of the most appropriate collusion detection mechanism that applies for different types of collusion and for different population configurations. These will result in the collusion detection process being more adaptive to the different situations that agents may face.

Chapter 6

Summary and Conclusions

Agent-based systems are increasingly being used in various domains due to their robustness, flexibility of structure, and capability of supporting complex processes involving the consideration of many factors. In many cases, agents would be used more often and more effectively to autonomously perform tasks on behalf of their users if the uncertainty involved in agent interactions in open, distributed systems could be better managed. Trust and reputation have been proposed as solutions to the issue of agent interaction uncertainty. They enable agents to assess the trustworthiness of potential interaction partners, before selecting the most suitable agent to perform a task towards the achievement of their goals.

In this thesis, we started off with a number of research objectives and in light of these objectives, we have made the following contributions:

- We proposed a trust and reputation model that allows agents to collect a rich set of routine interaction and recommendation information, to support further analysis of their social networks. Our model also improves trust assessment in certain cases, and it performs as well as other trust and reputation models in other cases.

- We represented agents' social networks using graphs and described how agents can use the interaction and recommendation information gathered to build and maintain social network models. We also provided visualisation tools to enable further reasoning about agent relationships by human analysts.

- As a first step to collusion detection, we identified the types of collusion that exist in the e-commerce domain and presented a taxonomy to characterise them in terms of agent interactions and recommendations.

- We have shown how cosine similarity measurement can be used to detect target-witness collusion by identifying pairs of collusive agents in the evaluator's social network. Collusion detection is supported by knowledge of the agent's social network and rich interaction and recommendation information.

Despite there being many widely adopted trust models, there are still areas of uncertainty in agent interactions that make it difficult for agents to be used more widely.

- The complexity of agent interactions is largely due to the number of agents in the population, the different services being requested and offered, and the complex relationships among agents for the roles of provider of services and witness of information. This often makes it difficult for system architects to easily understand agent systems and identify clear patterns of behaviour, which are often necessary to tune system parameters.

- Malicious behaviour, such as collusion, remains an issue in agent-based systems, especially in decentralised systems, as we have discussed in Sections 2.8.2.2 and 5.5.1. In systems where individual agents have to deal with collusion themselves (compared to collusion being detected and punished by an appointed authority), it is difficult for these agents to individually identify malicious behaviour and use this information in their future interactions.

These two key issues led us to investigate how to support both human analysts, such as system architects responsible for developing and managing agent systems, and agents themselves to more effectively handle malicious behaviour with a minimum amount of overhead. This question has been subdivided into more specific questions which have been answered in Chapters 3, 4 and 5.

Agents using a trust model to represent the trustworthiness of other agents already collect some information about their environment as a result of their interactions, as we have seen in Chapter 3. We explored how agents can collect additional information during their interactions to acquire a more complete view of their environment, as well as to better inform their agent selection decisions. Information from direct interactions with provider agents is as valuable as recommendations from witnesses. Although less accurate than direct service interactions, due to some degree of subjectivity, direct and indirect recommendations have their place in trust models as they provide crucial agent relationship information. The accuracy of trustworthiness evaluations is enhanced by taking into account the aspects of trust and reputation in individual service dimensions, considering the recency of interactions, and assessing the relevance of witnesses when using their recommendations. We have incorporated the above considerations into our trust and reputation model as the basis for gathering a richer set of information about the agent environment and for more accurately assessing agent trustworthiness.

A related issue is the representation of the information gathered and how to support the understanding of complex agent systems by system architects who are responsible for fine-tuning parameters for the best performance. Given the complexity of the interactions between agents, especially in large dynamic environments, it is difficult for human analysts to analyse the vast amounts of data that are represented in the agent interaction histories. Tools to support users to better understand agent systems are essential for maintenance and analysis. As we have seen in Chapter 4, agent graphs, built and maintained by agents, together with visualising tools, allow for an easier way of understanding how agents are interconnected and what information they share. Consequently, a better understanding of the agent environment leads to the extraction of previously unknown information that can be useful for decision making, both for system architects and the agents themselves. We have identified the link between the representation of the richer set of interaction data and the extraction of previously unknown information, such as identifying malicious behaviour.

In Chapter 5, we further explored the issue of malicious behaviour by looking more closely at collusion. While the evaluation of trust by agents can filter out poor performing agents, using trust alone does not allow for the more complex analysis of which agents are colluding and the type of collusion involved. Towards this end, we proposed a taxonomy of collusion characteristics that helps in identifying particular types of collusion occurring in multi-agent systems in the e-commerce domain. Additionally, we have proposed and evaluated a novel way of using similarity measurement to detect collusion, based on the information gathered by agents during their interactions with other agents.

In the remainder of this chapter we consider our contributions in turn, and discuss the limitations of our approach and identify areas for future extension.

6.1 Trust and Reputation Model

6.1.1 Summary of Contributions

In Section 2.3 we identified the essential characteristics of an effective trust model, namely trust information sources (trust, direct and indirect recommendations), consideration of trust and reputation at service-level and in different service dimensions, recency of interactions, and relevance of recommendations. Based on these characteristics, we proposed a trust and reputation model in Chapter 3. Using this model, agents gather a richer set of information that in previous trust models, allowing for a more accurate trust assessment and supporting a richer range of reasoning and analysis both by agents and humans. We have shown that our model performs at least as well as other trust models and in some cases it performs better (Section 3.5.2). This richer set of information enables further information discovery, notably for collusion detection.

6.1.2 Limitations

There are a number of limitations to our proposed trust model implementation and evaluation. Firstly, the principal witness will request recommendations from an agent that has previously directly interacted with the target agent. For simplicity we have restricted our implementation, to use only two degrees of separation between the evaluator and the secondary witness to provide a proof of concept that indirect recommendations have a positive impact on trust assessment.

Secondly, in our evaluation, we have demonstrated that our trust and reputation model works well for small to medium population sizes (10 to 50 individuals). Due to the extensive number of population configurations generated (over 8500), it was not practical to evaluate complete experimental sets for larger population sizes. We have carried out initial experiments on larger population sizes, considering 10 to 100 agents in 10 agent increments for Behaviour A for three population configurations (Population 1, Population 2, Population 3). The paired *t*-test results we obtained, gave similar results to those in Section 3.5.2, and are included Appendix B.2.

6.1.3 Future Directions

In Section 2.6, we introduced some of the existing work related to trust propagation and the notion of six degrees of separation in small world networks. One area of importance that has not been fully explored is the impact of the number of links between two agents in a network for sharing recommendations. The effect of longer recommendation chains needs to be further investigated in order to identify the optimal chain length in such networks.

For technologists aiming to improve the performance of agents using trust and reputation, one potential direction is to investigate the adaptation of agents to more dynamic changes in agent behaviours. As demonstrated by the evaluation of our model, an agent population has a number of parameters that affect an evaluator's ability to accurately assess the trustworthiness of another agent. An important research topic is how different combinations of population parameters impact on the evaluator's performance and the ways that it can adapt its trust evaluation to take these parameters into account.

6.2 Agent Graphs and Visualisation Tool

6.2.1 Summary of Contributions

In Section 1.2.2, we hypothesised that agents should be more fully equipped to gain a better understanding of their environment by capturing the data gathered through agent interactions in a clear and systematic way. We have shown in Chapter 4 how an agent can more expressively represent its agent interactions and relationships in the form of agent graphs. Four types of graph can be built and maintained by an evaluator to understand the distribution of providers and witnesses, as well as the relationships between agents in terms of service provision or general agent interactions. We also discussed several examples of new information that agents can extract from the agent graphs, such as collusion, and the re-discovery of service providers after a period of them being untrustworthy,

To facilitate the representation of the agent population to human analysts, we have provided a visualisation tool for viewing the status of an agent's view of the environment. The graph nodes show the agents involved and the edges linking the nodes provide information on how agents are linked, whether by service provision or by recommendation sharing. The visualisation tool enables users to rearrange the agent graph to make it easier and convenient to read.

6.2.2 Limitations

The agent graphs are intended to represent a snapshot of an agent's view of its environment. The scope of this snapshot includes history of the agent's own interactions at that time along with the recommendations being received by the agent. Snapshots of the agent's environment at regular intervals are not taken and recorded. Agents will refer to the current state of the graphs when they need to analyse their environment. However, information on the evolution of the environment may be of importance.

Another limitation is that the graphs contain information from the point of view of only one agent and from the recommendations it receives. Human analysts may need to view the entire population or groups of agents.

The graphs and the visualisation tool only show the current state of agents' interactions. They do not show any interpretation of agent behaviours that can be extracted from the data. Annotations by human users about agent relationships cannot currently be added to the graphs to increase the quality and accuracy of the data that the agent holds.

6.2.3 Future Directions

Information about the evolution of an agent's environment may provide vital clues in understanding its interactions and decision making over time. A future development for the agent graphs is to include a feature to enable human analysts to visualise the agent population as a whole, rather than from an individual agent's point of view, to give a richer understanding of how the agents in the population are working together. Similarly, enhanced visualisation can be developed to highlight particular agent activities of interest, such as clusters of high agent interactions.

6.3 Collusion Detection

6.3.1 Summary of Contributions

Malicious behaviour in agent populations, such as collusion, increases the risk associated with agent interactions, as discussed in Section 1.2.4. Collusion detection begins with the identification of the characteristics of collusion among agents. We proposed a taxonomy of collusion characteristics to identify collusion and associate collusion types occurring in e-commerce systems with their characteristics. Furthermore, we presented a novel way of using similarity measurement for the detection of persistent collusion between a witness agent and the target agent being assessed by the evaluator. We have shown in Section 5.7 that cosine similarity measurement works well in some circumstances for detecting collusion of this type. Evaluator agents are able to detect pairs of colluding target and witness agents, which is an additional benefit to only identifying that these two agents are untrustworthy, but with no cause for their untrustworthiness.

6.3.2 Limitations

In our work, we assessed only one type of collusion, persistent target-witness collusion. We chose this collusion type as a proof of concept for using cosine similarity for collusion detection in agent systems. Additionally, the agent relationships necessary for detection are also well represented in the agent graphs.

According to the literature [124, 138], cosine similarity measurement is the most appropriate measure for service-oriented domains, including e-commerce. We used cosine similarity measurement in our approach, but we did not compare it with other techniques. A complete validation would have involved extensive work for the evaluation sets and we chose to extensively evaluate the cosine similarity measurement since our aim was to assess whether similarity measurement could be used to detect collusion.

In our approach, we do not limit the number of potentially collusive agent pairs that the evaluator can detect. We argue that it is useful for an evaluator to identify all the potentially collusive agent pairs in its environment. In some cases the number of detected pairs can be large and taking note of all the potentially collusive pairs may not be as useful as applying a ranking according to the likelihood of the pairs being collusive.

6.3.3 Future Directions

Different types of collusion may occur in service-oriented agent systems, as we have seen in Section 5.4.2 for the e-commerce domain. Immediate future work will consist of investigating the occurrence of those types of collusion and how agents can use their interaction data and agent graphs to detect them. It is highly likely that different collusion detection mechanisms would suit different types of collusion better. Interesting challenges include the identification of other suitable collusion detection mechanisms that make use of the data already gathered from agent interactions. The aim is to enhance the ability of individual agents to detect collusion themselves.

To improve on accurate collusion detection, a confidence value can be attached to the detected collusive pair, such that a ranking method can be used to help an evaluator manage its list of detected collusive pairs. The motivation is to enable agents to act on the collusion detection information to influence future decision making. If the list of potentially collusive agent pairs is long, it will be difficult for an evaluator to effectively prevent or manage the effects of future collusion without a limited ordered list of which collusive agent pairs to prioritise for punishment.

As mentioned previously, collusion detection is the essential precursor to collusion prevention and management. It is therefore necessary to consider ways in which agents can use collusion detection information to inform their trust assessment mechanism for agent selection. Techniques, such as machine learning could be used to improve collusion detection and subsequently improve trust assessment.

6.4 Final Conclusions

Throughout this thesis, based on literature, implementation and evaluation, we have shown that the uncertainty of agent interactions in open, distributed, and heterogeneous systems can be reduced with a combination of approaches to improve trust assessment, to better represent an agent's environment, and to detect collusion. Our work has shown that an individual agent can reduce its number of failed interactions with other agents in this way, through more effectively using the data gathered during its interactions.

Our proposed model of trust and reputation facilitates both the trust assessment of potential interaction partners and the gathering of a richer set of data for additional information extraction. While our model performs as effectively as some of the other trust models, it has the added benefit of ensuring that the evaluator has sufficient information about its environment for analysis. Our use of multiple trust dimensions, indirect recommendations, recency of interactions and relevance of witnesses contributes to this richer data set and in some cases helps to improve trust assessment. A number of agent population parameters influence the way in which agent interactions take place. Evaluator agents need to be able to identify the parameter combinations that apply and also the dynamic changes in agent behaviour. Future work into these themes will ensure that agents can assess the trustworthiness of other agents quickly and efficiently to inform their decision making.

In terms of analysing the interaction data gathered from the service interactions, we have shown that the data can be effectively represented as graphs which, when used individually or in combination, helps human analysts to better understand the underlying interactions within a multi-agent system. The agent graphs are represented further by way of a visualisation tool that assists users in restructuring the graphs for ease of access. Further work in this area should focus on presenting complete information to the user, such as snapshots of the agent environment over time. Enabling the user to highlight particular agent behaviour, such as confirming or rejecting detected collusion, can be a means to allow the user to incorporate verified information into the agent selection process.

Collusion among agents is one of the sources of uncertainty in agent interactions. Very little work has previously been done in giving more power to the agents themselves to detect collusion with little overhead. Using Kleinberg's argument [57] that a partial view of the agent environment may be sufficient to extract collusion information, we have built on the interaction data gathered and the agent graphs to detect collusion. Similarity measurement techniques are suited for collusion detection since similarity or dissimilarity between particular behaviours indicates a high probability of collusion or malicious behaviour that results in similar effects as collusion. We have demonstrated that using cosine similarity measurement for collusion detection is effective in some cases for persistent target-witness collusion. The important next steps in this research need to focus on ensuring that an evaluator agent can detect this type of collusion accurately every time it occurs in its environment. The agent population configuration influences the detection process but it is not yet clear how and hence requires to be further investigated.

In the future, collusion detection needs to be included within the trust assessment so that the information can be used to inform future interactions. With collusion detection being a complex process, the role of human analysts will be crucial in helping the agent system to learn to detect collusion more accurately.

To conclude, we can say that a trust model with a richer set of interaction data can help individual agents to better represent their agent environment and subsequently extract previously unknown information, including that relating to collusion, with the aim of reducing the uncertainty in agent interactions. Tools to support human analysts to better understand agent systems can greatly help towards improving the way agents can truly become autonomous and used more widely. Hence, further research needs to take both the agent and human components into account to resolve some of the more complex issues within agent-based systems.

Appendix A

Review of Trust Model Characteristics

Model	Trust model characteristics							
	Trust information sources				Service-level		Recency	Relevance
	Trust	Reputation	Direct recommendation	Indirect recommendation	Trust multi-dimension	Recommendation multi-dimension		
Castelfranchi & Falcone	Y	N	N	N	N	N	N	N
SIR (Mezzetti)	Y	Y	Y	N	Y	N	Y	N
Marsh	Y	N	N	N	N	N	N	N
Ntropi	Y	Y	Y	Y	N	N	N	Y
ReGreT (Sabater et al.)	Y	Y	Y	Y	Y	Y	Y	N
TrustNet (Schillo)	Y	Y	Y	N	N	N	N	N
Mui et al.	Y	Y	Y	Y	N	N	N	Partial
Braynov & Sandholm	Y	N	N	N	N	N	N	N
Wu & Sun	Y	N	N	N	N	N	N	N
Witkowski et al.	Y	N	N	N	N	N	Y	N
Sen & Dutta	Y	N	N	N	N	N	N	N
Sen & Sajja	Y	Y	Y	N	N	N	N	N
SPORAS (Zacharia et al.)	Y	Y (global)	Y	N	N	N	Y	N
HISTOS (Zacharia et al.)	Y	Y	Y	Y	N	N	Y	Y
Griffiths & Luck	Y	N	N	N	Y	N	N	N
MDT-R (Griffiths)	Y	Y	Y	N	Y	Y	Y	Partial
FIRE (Huynh et al.)	Y	Y	Y	Y	N	N	Y	Y
TRAVOS (Teacy et al.)	Y	Y	Y	N	N	N	N	N
Walter et al.	Y	Y	Y	Y	N	N	N	N
L.I.A.R. (Muller & Vercouter)	Y	Y	Y	Y	N	N	N	Partial
Yu & Singh	Y	Y	Y	Y	N	N	N	Y

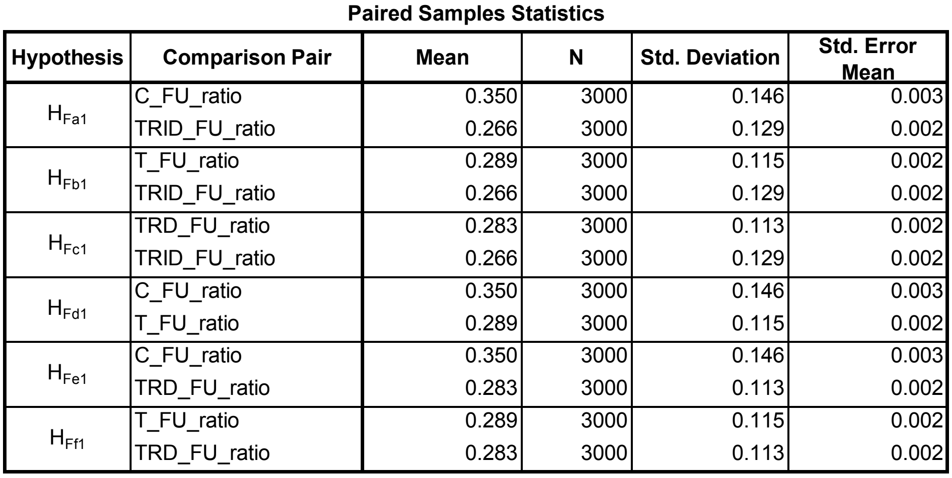
Table A.1: Summary of Trust Model Characteristics.

Appendix B

Trust and Reputation Model Evaluation

B.1 Paired t-Tests for Population Sizes 10 to 50 for Behaviour Profiles A–D

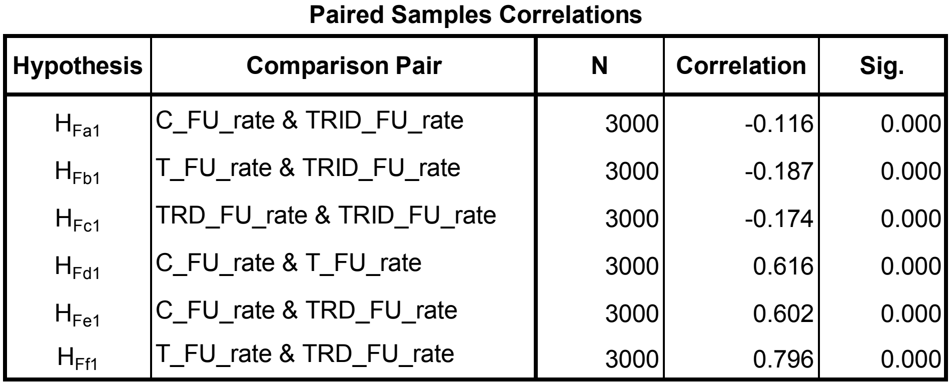
B.1.1 Hypotheses Results for Failed Task Ratios



Paired Samples Statistics

Hypothesis	Comparison Pair	Mean	N	Std. Deviation	Std. Error Mean
H_{Fa1}	C_FU_ratio	0.350	3000	0.146	0.003
	TRID_FU_ratio	0.266	3000	0.129	0.002
H_{Fb1}	T_FU_ratio	0.289	3000	0.115	0.002
	TRID_FU_ratio	0.266	3000	0.129	0.002
H_{Fc1}	TRD_FU_ratio	0.283	3000	0.113	0.002
	TRID_FU_ratio	0.266	3000	0.129	0.002
H_{Fd1}	C_FU_ratio	0.350	3000	0.146	0.003
	T_FU_ratio	0.289	3000	0.115	0.002
H_{Fe1}	C_FU_ratio	0.350	3000	0.146	0.003
	TRD_FU_ratio	0.283	3000	0.113	0.002
H_{Ff1}	T_FU_ratio	0.289	3000	0.115	0.002
	TRD_FU_ratio	0.283	3000	0.113	0.002

Table B.1: Failed Tasks Hypotheses Results for Behaviour Profile A (Statistics)



Paired Samples Correlations

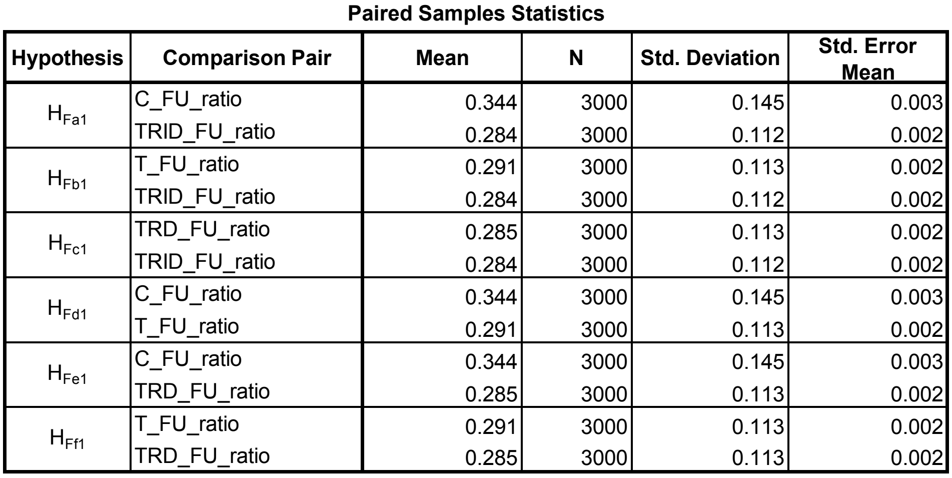
Hypothesis	Comparison Pair	N	Correlation	Sig.
H_{Fa1}	C_FU_rate & TRID_FU_rate	3000	-0.116	0.000
H_{Fb1}	T_FU_rate & TRID_FU_rate	3000	-0.187	0.000
H_{Fc1}	TRD_FU_rate & TRID_FU_rate	3000	-0.174	0.000
H_{Fd1}	C_FU_rate & T_FU_rate	3000	0.616	0.000
H_{Fe1}	C_FU_rate & TRD_FU_rate	3000	0.602	0.000
H_{Ff1}	T_FU_rate & TRD_FU_rate	3000	0.796	0.000

Table B.2: Failed Tasks Hypotheses Results for Behaviour Profile A (Correlations)

Paired Samples Test

Hypothesis	Mean Difference	Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
					Lower	Upper			
H_{Fa1}	C_FU_rate - TRID_FU_rate	0.084	0.206	0.004	0.077	0.091	22.348	2999	0.000
H_{Fb1}	T_FU_rate - TRID_FU_rate	0.023	0.189	0.003	0.016	0.029	6.590	2999	0.000
H_{Fc1}	TRD_FU_rate - TRID_FU_rate	0.016	0.186	0.003	0.010	0.023	4.839	2999	0.000
H_{Fd1}	C_FU_rate - T_FU_rate	0.061	0.118	0.002	0.057	0.066	28.554	2999	0.000
H_{Fe1}	C_FU_rate - TRD_FU_rate	0.068	0.119	0.002	0.063	0.072	31.089	2999	0.000
H_{Ff1}	T_FU_rate - TRD_FU_rate	0.006	0.073	0.001	0.004	0.009	4.702	2999	0.000

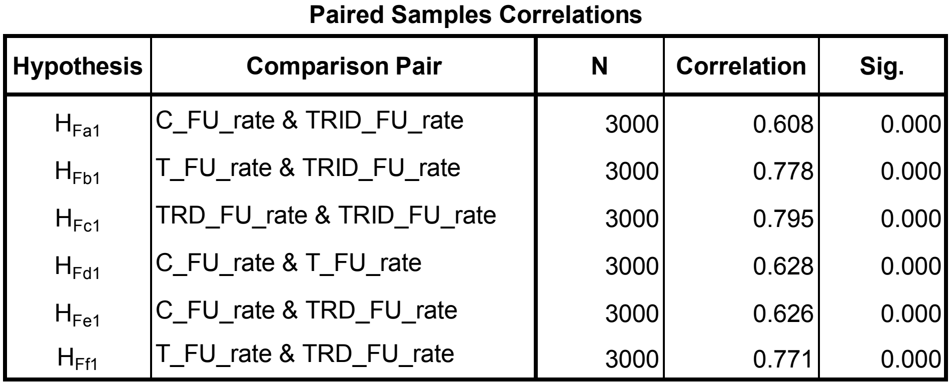
Table B.3: Failed Tasks Hypotheses Results for Behaviour Profile A (Tests)



Paired Samples Statistics

Hypothesis	Comparison Pair	Mean	N	Std. Deviation	Std. Error Mean
H_{Fa1}	C_FU_ratio	0.344	3000	0.145	0.003
	TRID_FU_ratio	0.284	3000	0.112	0.002
H_{Fb1}	T_FU_ratio	0.291	3000	0.113	0.002
	TRID_FU_ratio	0.284	3000	0.112	0.002
H_{Fc1}	TRD_FU_ratio	0.285	3000	0.113	0.002
	TRID_FU_ratio	0.284	3000	0.112	0.002
H_{Fd1}	C_FU_ratio	0.344	3000	0.145	0.003
	T_FU_ratio	0.291	3000	0.113	0.002
H_{Fe1}	C_FU_ratio	0.344	3000	0.145	0.003
	TRD_FU_ratio	0.285	3000	0.113	0.002
H_{Ff1}	T_FU_ratio	0.291	3000	0.113	0.002
	TRD_FU_ratio	0.285	3000	0.113	0.002

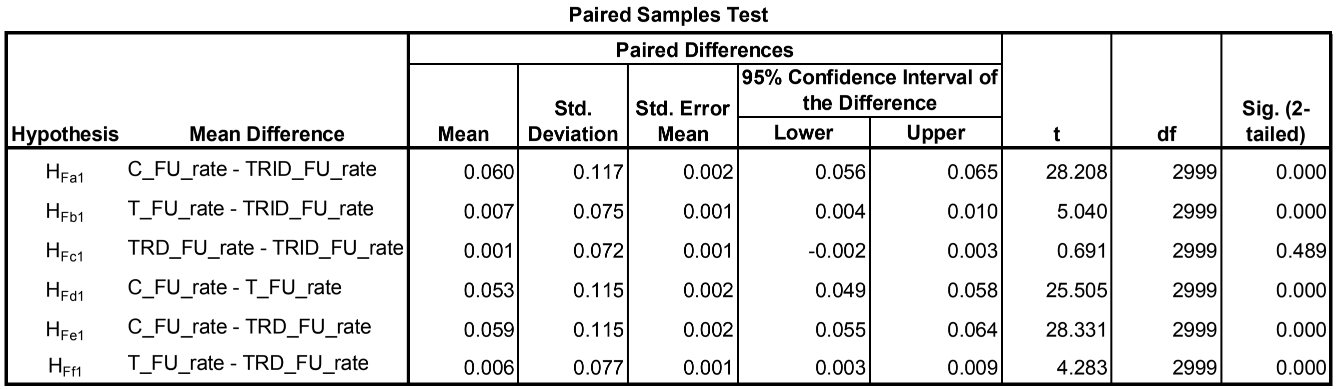
Table B.4: Failed Tasks Hypotheses Results for Behaviour Profile B (Statistics)



Paired Samples Correlations

Hypothesis	Comparison Pair	N	Correlation	Sig.
H_{Fa1}	C_FU_rate & TRID_FU_rate	3000	0.608	0.000
H_{Fb1}	T_FU_rate & TRID_FU_rate	3000	0.778	0.000
H_{Fc1}	TRD_FU_rate & TRID_FU_rate	3000	0.795	0.000
H_{Fd1}	C_FU_rate & T_FU_rate	3000	0.628	0.000
H_{Fe1}	C_FU_rate & TRD_FU_rate	3000	0.626	0.000
H_{Ff1}	T_FU_rate & TRD_FU_rate	3000	0.771	0.000

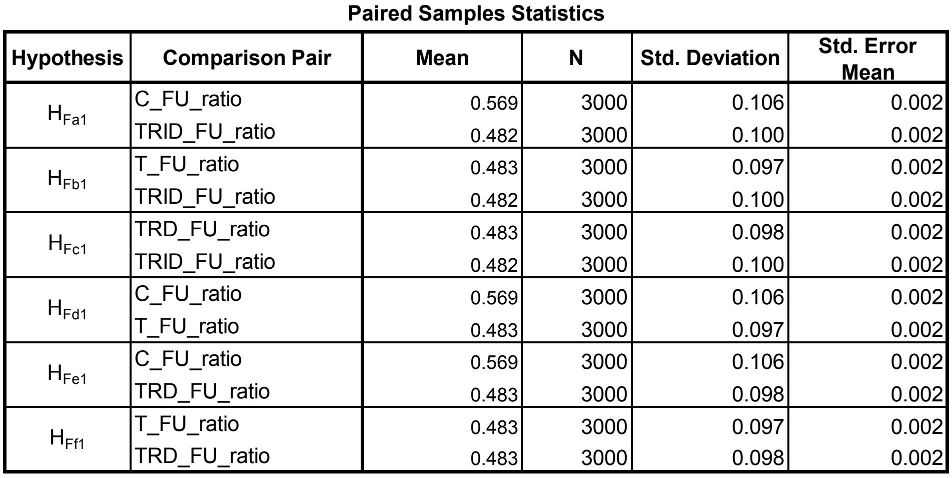
Table B.5: Failed Tasks Hypotheses Results for Behaviour Profile B (Correlations)



Paired Samples Test

Hypothesis	Mean Difference	Paired Differences					t	df	Sig. (2-tailed)
					95% Confidence Interval of the Difference				
		Mean	Std. Deviation	Std. Error Mean	Lower	Upper			
H_{Fa1}	C_FU_rate - TRID_FU_rate	0.060	0.117	0.002	0.056	0.065	28.208	2999	0.000
H_{Fb1}	T_FU_rate - TRID_FU_rate	0.007	0.075	0.001	0.004	0.010	5.040	2999	0.000
H_{Fc1}	TRD_FU_rate - TRID_FU_rate	0.001	0.072	0.001	-0.002	0.003	0.691	2999	0.489
H_{Fd1}	C_FU_rate - T_FU_rate	0.053	0.115	0.002	0.049	0.058	25.505	2999	0.000
H_{Fe1}	C_FU_rate - TRD_FU_rate	0.059	0.115	0.002	0.055	0.064	28.331	2999	0.000
H_{Ff1}	T_FU_rate - TRD_FU_rate	0.006	0.077	0.001	0.003	0.009	4.283	2999	0.000

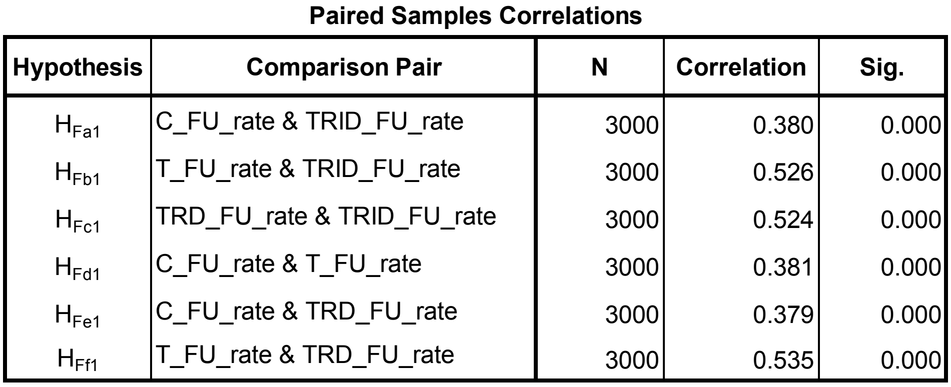
Table B.6: Failed Tasks Hypotheses Results for Behaviour Profile B (Tests)



Paired Samples Statistics

Hypothesis	Comparison Pair	Mean	N	Std. Deviation	Std. Error Mean
H_{Fa1}	C_FU_ratio	0.569	3000	0.106	0.002
	TRID_FU_ratio	0.482	3000	0.100	0.002
H_{Fb1}	T_FU_ratio	0.483	3000	0.097	0.002
	TRID_FU_ratio	0.482	3000	0.100	0.002
H_{Fc1}	TRD_FU_ratio	0.483	3000	0.098	0.002
	TRID_FU_ratio	0.482	3000	0.100	0.002
H_{Fd1}	C_FU_ratio	0.569	3000	0.106	0.002
	T_FU_ratio	0.483	3000	0.097	0.002
H_{Fe1}	C_FU_ratio	0.569	3000	0.106	0.002
	TRD_FU_ratio	0.483	3000	0.098	0.002
H_{Ff1}	T_FU_ratio	0.483	3000	0.097	0.002
	TRD_FU_ratio	0.483	3000	0.098	0.002

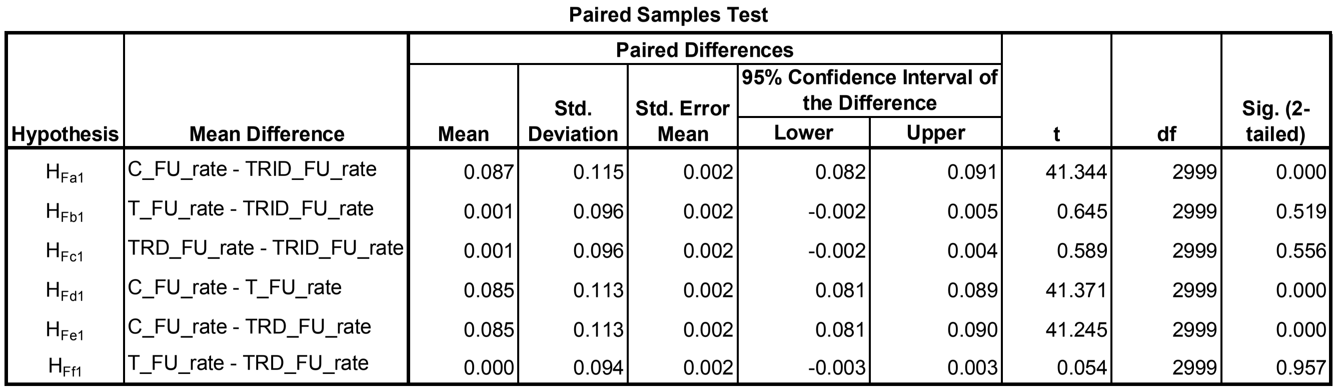
Table B.7: Failed Tasks Hypotheses Results for Behaviour Profile C (Statistics)



Paired Samples Correlations

Hypothesis	Comparison Pair	N	Correlation	Sig.
H_{Fa1}	C_FU_rate & TRID_FU_rate	3000	0.380	0.000
H_{Fb1}	T_FU_rate & TRID_FU_rate	3000	0.526	0.000
H_{Fc1}	TRD_FU_rate & TRID_FU_rate	3000	0.524	0.000
H_{Fd1}	C_FU_rate & T_FU_rate	3000	0.381	0.000
H_{Fe1}	C_FU_rate & TRD_FU_rate	3000	0.379	0.000
H_{Ff1}	T_FU_rate & TRD_FU_rate	3000	0.535	0.000

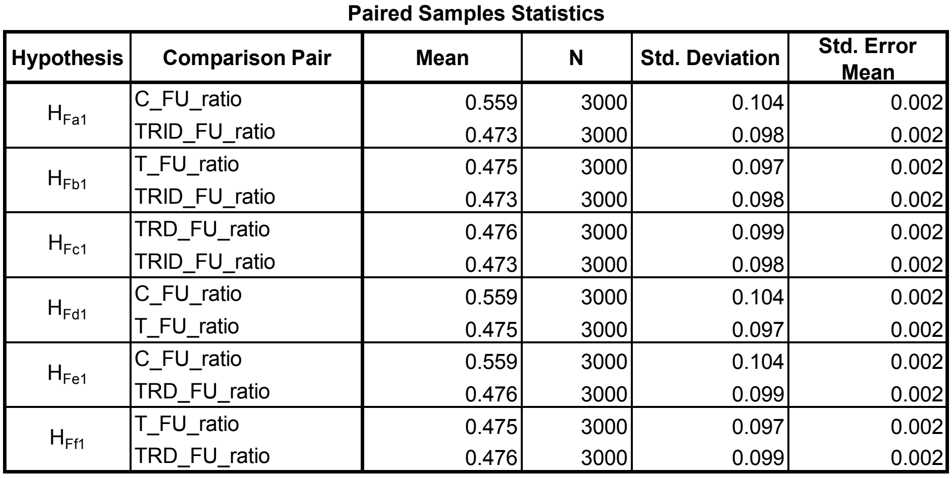
Table B.8: Failed Tasks Hypotheses Results for Behaviour Profile C (Correlations)



Paired Samples Test

Hypothesis	Mean Difference	Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
					Lower	Upper			
H_{Fa1}	C_FU_rate - TRID_FU_rate	0.087	0.115	0.002	0.082	0.091	41.344	2999	0.000
H_{Fb1}	T_FU_rate - TRID_FU_rate	0.001	0.096	0.002	-0.002	0.005	0.645	2999	0.519
H_{Fc1}	TRD_FU_rate - TRID_FU_rate	0.001	0.096	0.002	-0.002	0.004	0.589	2999	0.556
H_{Fd1}	C_FU_rate - T_FU_rate	0.085	0.113	0.002	0.081	0.089	41.371	2999	0.000
H_{Fe1}	C_FU_rate - TRD_FU_rate	0.085	0.113	0.002	0.081	0.090	41.245	2999	0.000
H_{Ff1}	T_FU_rate - TRD_FU_rate	0.000	0.094	0.002	-0.003	0.003	0.054	2999	0.957

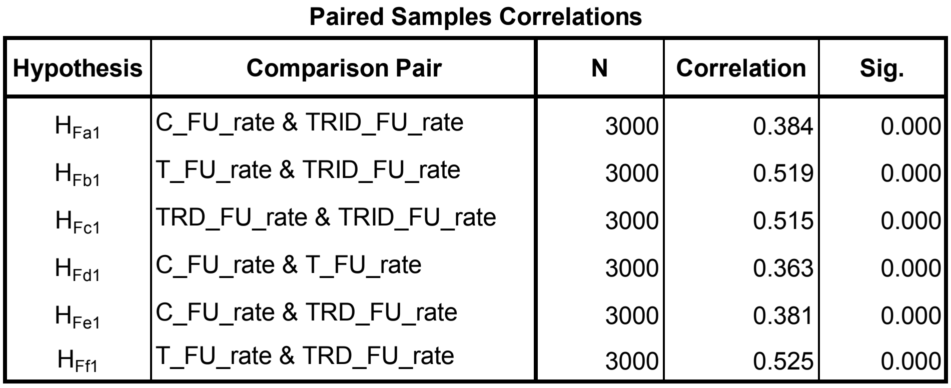
Table B.9: Failed Tasks Hypotheses Results for Behaviour Profile C (Tests)



Paired Samples Statistics

Hypothesis	Comparison Pair	Mean	N	Std. Deviation	Std. Error Mean
H_{Fa1}	C_FU_ratio	0.559	3000	0.104	0.002
	TRID_FU_ratio	0.473	3000	0.098	0.002
H_{Fb1}	T_FU_ratio	0.475	3000	0.097	0.002
	TRID_FU_ratio	0.473	3000	0.098	0.002
H_{Fc1}	TRD_FU_ratio	0.476	3000	0.099	0.002
	TRID_FU_ratio	0.473	3000	0.098	0.002
H_{Fd1}	C_FU_ratio	0.559	3000	0.104	0.002
	T_FU_ratio	0.475	3000	0.097	0.002
H_{Fe1}	C_FU_ratio	0.559	3000	0.104	0.002
	TRD_FU_ratio	0.476	3000	0.099	0.002
H_{Ff1}	T_FU_ratio	0.475	3000	0.097	0.002
	TRD_FU_ratio	0.476	3000	0.099	0.002

Table B.10: Failed Tasks Hypotheses Results for Behaviour Profile D (Statistics)



Paired Samples Correlations

Hypothesis	Comparison Pair	N	Correlation	Sig.
H_{Fa1}	C_FU_rate & TRID_FU_rate	3000	0.384	0.000
H_{Fb1}	T_FU_rate & TRID_FU_rate	3000	0.519	0.000
H_{Fc1}	TRD_FU_rate & TRID_FU_rate	3000	0.515	0.000
H_{Fd1}	C_FU_rate & T_FU_rate	3000	0.363	0.000
H_{Fe1}	C_FU_rate & TRD_FU_rate	3000	0.381	0.000
H_{Ff1}	T_FU_rate & TRD_FU_rate	3000	0.525	0.000

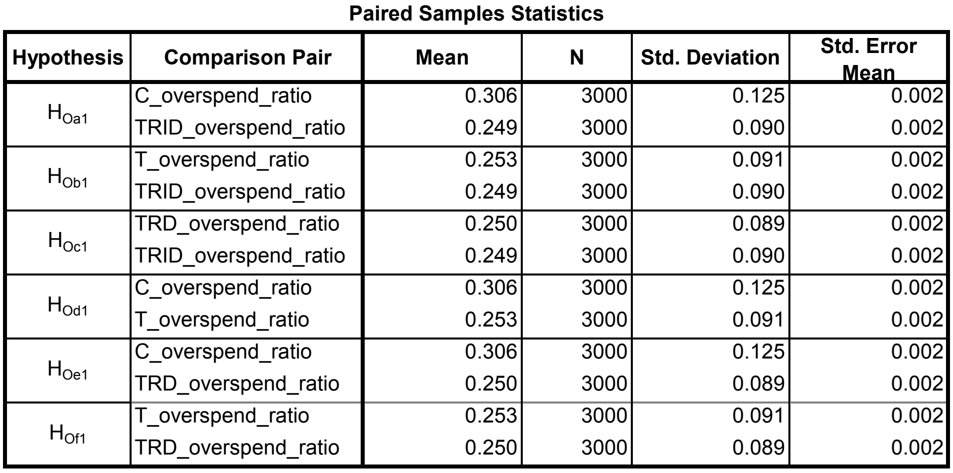
Table B.11: Failed Tasks Hypotheses Results for Behaviour Profile D (Correlations)

Paired Samples Test

		Paired Differences							
					95% Confidence Interval of the Difference				
Hypothesis	Mean Difference	Mean	Std. Deviation	Std. Error Mean	Lower	Upper	t	df	Sig. (2-tailed)
H_{Fa1}	C_FU_rate - TRID_FU_rate	0.085	0.113	0.002	0.081	0.089	41.369	2999	0.000
H_{Fb1}	T_FU_rate - TRID_FU_rate	0.002	0.096	0.002	-0.002	0.005	0.919	2999	0.358
H_{Fc1}	TRD_FU_rate - TRID_FU_rate	0.003	0.097	0.002	-0.001	0.006	1.572	2999	0.116
H_{Fd1}	C_FU_rate - T_FU_rate	0.084	0.114	0.002	0.079	0.088	40.141	2999	0.000
H_{Fe1}	C_FU_rate - TRD_FU_rate	0.082	0.114	0.002	0.078	0.086	39.753	2999	0.000
H_{Ff1}	T_FU_rate - TRD_FU_rate	-0.001	0.096	0.002	-0.005	0.002	-0.678	2999	0.498

Table B.12: Failed Tasks Hypotheses Results for Behaviour Profile D (Tests)

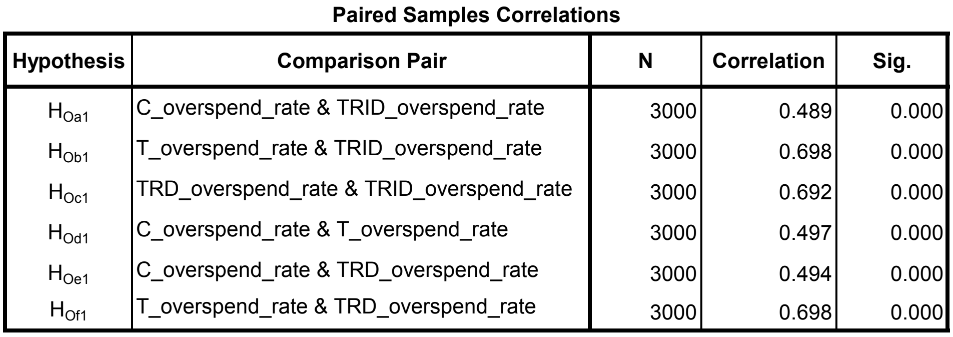
B.1.2 Hypotheses Results for Overspend Ratios



Paired Samples Statistics

Hypothesis	Comparison Pair	Mean	N	Std. Deviation	Std. Error Mean
H_{Oa1}	C_overspend_ratio	0.306	3000	0.125	0.002
	TRID_overspend_ratio	0.249	3000	0.090	0.002
H_{Ob1}	T_overspend_ratio	0.253	3000	0.091	0.002
	TRID_overspend_ratio	0.249	3000	0.090	0.002
H_{Oc1}	TRD_overspend_ratio	0.250	3000	0.089	0.002
	TRID_overspend_ratio	0.249	3000	0.090	0.002
H_{Od1}	C_overspend_ratio	0.306	3000	0.125	0.002
	T_overspend_ratio	0.253	3000	0.091	0.002
H_{Oe1}	C_overspend_ratio	0.306	3000	0.125	0.002
	TRD_overspend_ratio	0.250	3000	0.089	0.002
H_{Of1}	T_overspend_ratio	0.253	3000	0.091	0.002
	TRD_overspend_ratio	0.250	3000	0.089	0.002

Table B.13: Overspend Hypotheses Results for Behaviour Profile A (Statistics)



Paired Samples Correlations

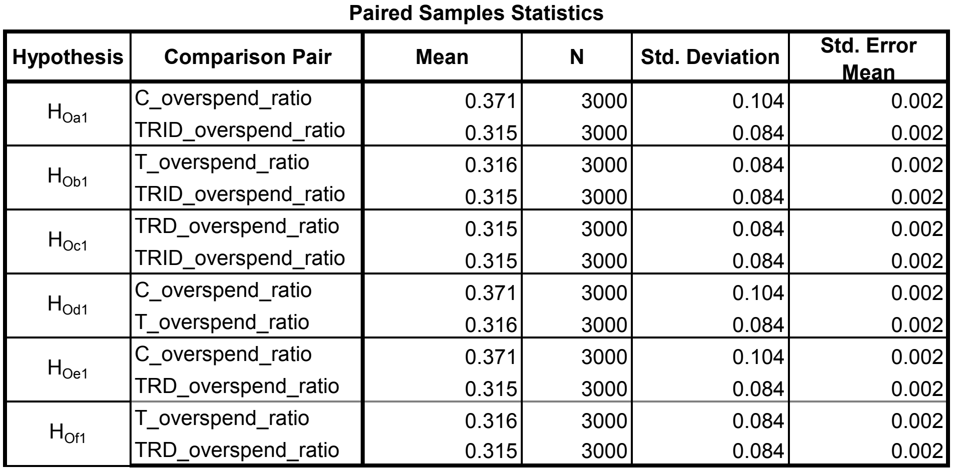
Hypothesis	Comparison Pair	N	Correlation	Sig.
H_{Oa1}	C_overspend_rate & TRID_overspend_rate	3000	0.489	0.000
H_{Ob1}	T_overspend_rate & TRID_overspend_rate	3000	0.698	0.000
H_{Oc1}	TRD_overspend_rate & TRID_overspend_rate	3000	0.692	0.000
H_{Od1}	C_overspend_rate & T_overspend_rate	3000	0.497	0.000
H_{Oe1}	C_overspend_rate & TRD_overspend_rate	3000	0.494	0.000
H_{Of1}	T_overspend_rate & TRD_overspend_rate	3000	0.698	0.000

Table B.14: Overspend Hypotheses Results for Behaviour Profile A (Correlations)

Paired Samples Test

Hypothesis	Mean Difference	Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
					Lower	Upper			
H_{Oa1}	C_overspend_rate - TRID_overspend_rate	0.057	0.112	0.002	0.053	0.061	27.888	2999	0.000
H_{Ob1}	T_overspend_rate - TRID_overspend_rate	0.004	0.070	0.001	0.002	0.007	3.309	2999	0.001
H_{Oc1}	TRD_overspend_rate - TRID_overspend_rate	0.001	0.070	0.001	-0.001	0.004	0.986	2999	0.324
H_{Od1}	C_overspend_rate - T_overspend_rate	0.053	0.112	0.002	0.049	0.057	25.962	2999	0.000
H_{Oe1}	C_overspend_rate - TRD_overspend_rate	0.056	0.112	0.002	0.052	0.060	27.432	2999	0.000
H_{Of1}	T_overspend_rate - TRD_overspend_rate	0.003	0.070	0.001	0.000	0.005	2.334	2999	0.020

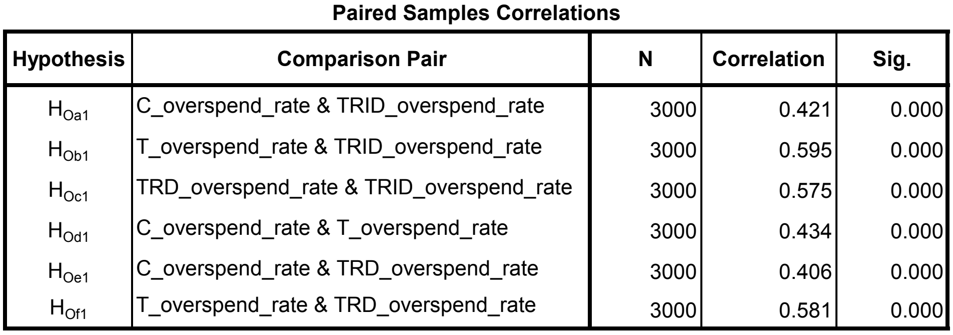
Table B.15: Overspend Hypotheses Results for Behaviour Profile A (Tests)



Paired Samples Statistics

Hypothesis	Comparison Pair	Mean	N	Std. Deviation	Std. Error Mean
H_{Oa1}	C_overspend_ratio	0.371	3000	0.104	0.002
	TRID_overspend_ratio	0.315	3000	0.084	0.002
H_{Ob1}	T_overspend_ratio	0.316	3000	0.084	0.002
	TRID_overspend_ratio	0.315	3000	0.084	0.002
H_{Oc1}	TRD_overspend_ratio	0.315	3000	0.084	0.002
	TRID_overspend_ratio	0.315	3000	0.084	0.002
H_{Od1}	C_overspend_ratio	0.371	3000	0.104	0.002
	T_overspend_ratio	0.316	3000	0.084	0.002
H_{Oe1}	C_overspend_ratio	0.371	3000	0.104	0.002
	TRD_overspend_ratio	0.315	3000	0.084	0.002
H_{Of1}	T_overspend_ratio	0.316	3000	0.084	0.002
	TRD_overspend_ratio	0.315	3000	0.084	0.002

Table B.16: Overspend Hypotheses Results for Behaviour Profile B (Statistics)



Paired Samples Correlations

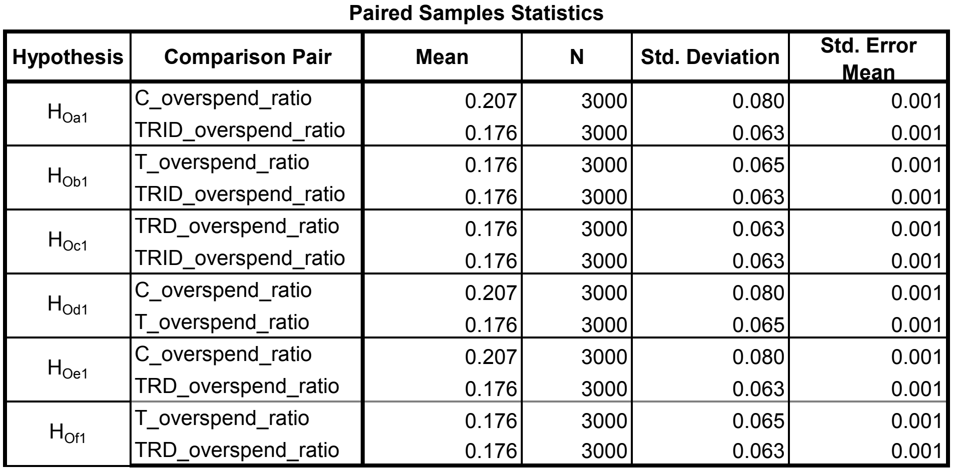
Hypothesis	Comparison Pair	N	Correlation	Sig.
H_{Oa1}	C_overspend_rate & TRID_overspend_rate	3000	0.421	0.000
H_{Ob1}	T_overspend_rate & TRID_overspend_rate	3000	0.595	0.000
H_{Oc1}	TRD_overspend_rate & TRID_overspend_rate	3000	0.575	0.000
H_{Od1}	C_overspend_rate & T_overspend_rate	3000	0.434	0.000
H_{Oe1}	C_overspend_rate & TRD_overspend_rate	3000	0.406	0.000
H_{Of1}	T_overspend_rate & TRD_overspend_rate	3000	0.581	0.000

Table B.17: Overspend Hypotheses Results for Behaviour Profile B (Correlations)

Paired Samples Test

		Paired Differences							
					95% Confidence Interval of the Difference				
Hypothesis	**Mean Difference**	**Mean**	**Std. Deviation**	**Std. Error Mean**	**Lower**	**Upper**	**t**	**df**	**Sig. (2-tailed)**
H_{Oa1}	C_overspend_rate - TRID_overspend_rate	0.055	0.103	0.002	0.052	0.059	29.396	2999	0.000
H_{Ob1}	T_overspend_rate - TRID_overspend_rate	0.001	0.076	0.001	-0.002	0.003	0.436	2999	0.663
H_{Oc1}	TRD_overspend_rate - TRID_overspend_rate	0.000	0.077	0.001	-0.003	0.002	-0.320	2999	0.749
H_{Od1}	C_overspend_rate - T_overspend_rate	0.055	0.102	0.002	0.051	0.058	29.429	2999	0.000
H_{Oe1}	C_overspend_rate - TRD_overspend_rate	0.056	0.104	0.002	0.052	0.059	29.354	2999	0.000
H_{Of1}	T_overspend_rate - TRD_overspend_rate	0.001	0.077	0.001	-0.002	0.004	0.753	2999	0.451

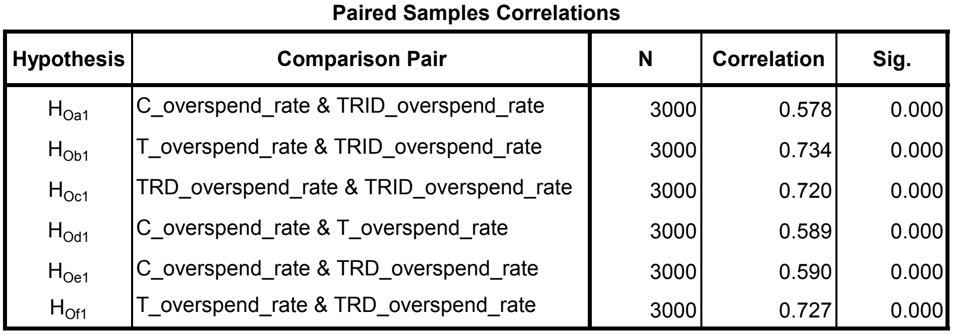
Table B.18: Overspend Hypotheses Results for Behaviour Profile B (Tests)



Paired Samples Statistics

Hypothesis	Comparison Pair	Mean	N	Std. Deviation	Std. Error Mean
H_{Oa1}	C_overspend_ratio	0.207	3000	0.080	0.001
	TRID_overspend_ratio	0.176	3000	0.063	0.001
H_{Ob1}	T_overspend_ratio	0.176	3000	0.065	0.001
	TRID_overspend_ratio	0.176	3000	0.063	0.001
H_{Oc1}	TRD_overspend_ratio	0.176	3000	0.063	0.001
	TRID_overspend_ratio	0.176	3000	0.063	0.001
H_{Od1}	C_overspend_ratio	0.207	3000	0.080	0.001
	T_overspend_ratio	0.176	3000	0.065	0.001
H_{Oe1}	C_overspend_ratio	0.207	3000	0.080	0.001
	TRD_overspend_ratio	0.176	3000	0.063	0.001
H_{Of1}	T_overspend_ratio	0.176	3000	0.065	0.001
	TRD_overspend_ratio	0.176	3000	0.063	0.001

Table B.19: Overspend Hypotheses Results for Behaviour Profile C (Statistics)



Paired Samples Correlations

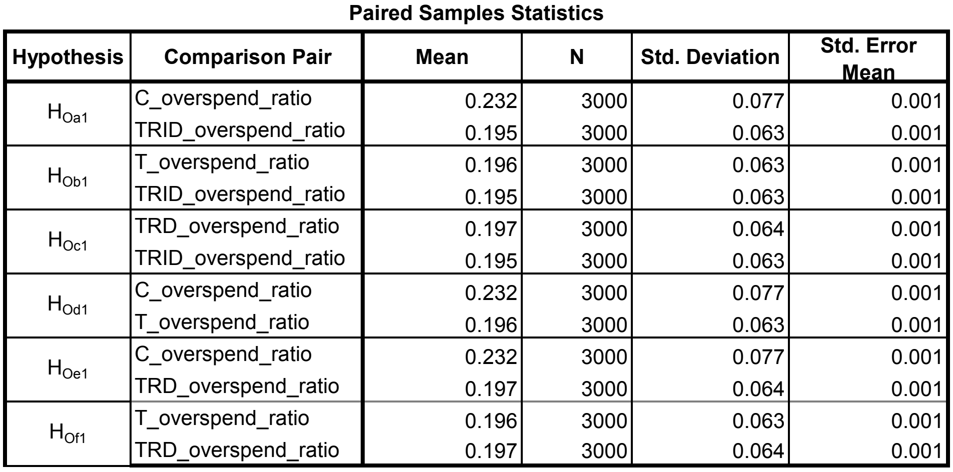
Hypothesis	Comparison Pair	N	Correlation	Sig.
H_{Oa1}	C_overspend_rate & TRID_overspend_rate	3000	0.578	0.000
H_{Ob1}	T_overspend_rate & TRID_overspend_rate	3000	0.734	0.000
H_{Oc1}	TRD_overspend_rate & TRID_overspend_rate	3000	0.720	0.000
H_{Od1}	C_overspend_rate & T_overspend_rate	3000	0.589	0.000
H_{Oe1}	C_overspend_rate & TRD_overspend_rate	3000	0.590	0.000
H_{Of1}	T_overspend_rate & TRD_overspend_rate	3000	0.727	0.000

Table B.20: Overspend Hypotheses Results for Behaviour Profile C (Correlations)

Paired Samples Test

Hypothesis	Mean Difference	Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
					Lower	Upper			
H_{Oa1}	C_overspend_rate - TRID_overspend_rate	0.031	0.068	0.001	0.029	0.033	25.093	2999	0.000
H_{Ob1}	T_overspend_rate - TRID_overspend_rate	0.001	0.047	0.001	-0.001	0.002	0.859	2999	0.390
H_{Oc1}	TRD_overspend_rate - TRID_overspend_rate	0.001	0.047	0.001	-0.001	0.002	0.661	2999	0.509
H_{Od1}	C_overspend_rate - T_overspend_rate	0.030	0.067	0.001	0.028	0.033	24.680	2999	0.000
H_{Oe1}	C_overspend_rate - TRD_overspend_rate	0.030	0.067	0.001	0.028	0.033	25.003	2999	0.000
H_{Of1}	T_overspend_rate - TRD_overspend_rate	0.000	0.047	0.001	-0.002	0.002	0.190	2999	0.850

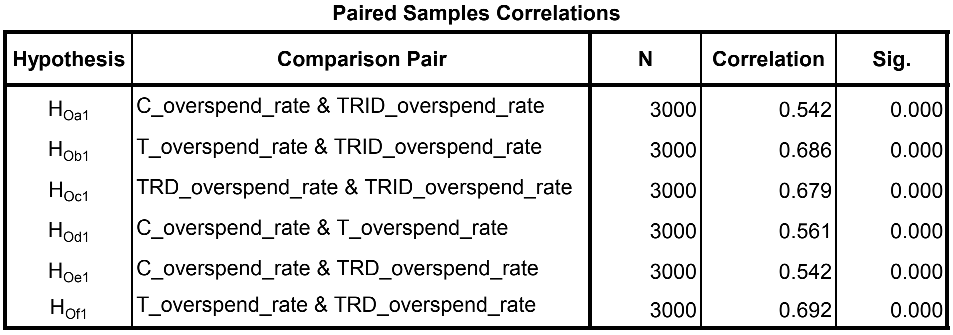
Table B.21: Overspend Hypotheses Results for Behaviour Profile C (Tests)



Paired Samples Statistics

Hypothesis	Comparison Pair	Mean	N	Std. Deviation	Std. Error Mean
H_{Oa1}	C_overspend_ratio	0.232	3000	0.077	0.001
	TRID_overspend_ratio	0.195	3000	0.063	0.001
H_{Ob1}	T_overspend_ratio	0.196	3000	0.063	0.001
	TRID_overspend_ratio	0.195	3000	0.063	0.001
H_{Oc1}	TRD_overspend_ratio	0.197	3000	0.064	0.001
	TRID_overspend_ratio	0.195	3000	0.063	0.001
H_{Od1}	C_overspend_ratio	0.232	3000	0.077	0.001
	T_overspend_ratio	0.196	3000	0.063	0.001
H_{Oe1}	C_overspend_ratio	0.232	3000	0.077	0.001
	TRD_overspend_ratio	0.197	3000	0.064	0.001
H_{Of1}	T_overspend_ratio	0.196	3000	0.063	0.001
	TRD_overspend_ratio	0.197	3000	0.064	0.001

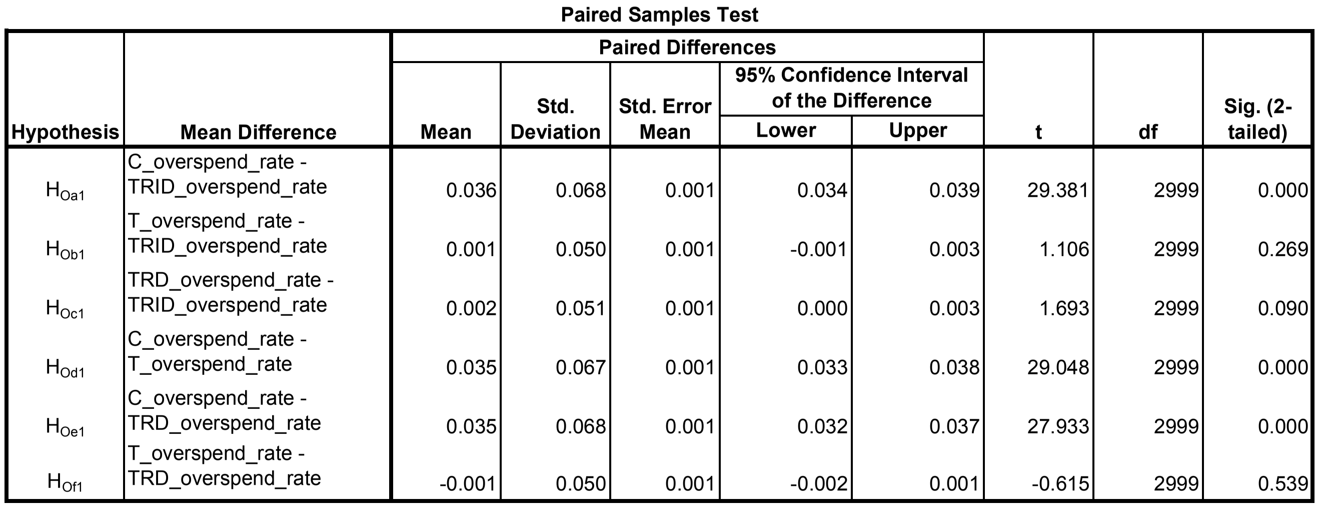
Table B.22: Overspend Hypotheses Results for Behaviour Profile D (Statistics)



Paired Samples Correlations

Hypothesis	Comparison Pair	N	Correlation	Sig.
H_{Oa1}	C_overspend_rate & TRID_overspend_rate	3000	0.542	0.000
H_{Ob1}	T_overspend_rate & TRID_overspend_rate	3000	0.686	0.000
H_{Oc1}	TRD_overspend_rate & TRID_overspend_rate	3000	0.679	0.000
H_{Od1}	C_overspend_rate & T_overspend_rate	3000	0.561	0.000
H_{Oe1}	C_overspend_rate & TRD_overspend_rate	3000	0.542	0.000
H_{Of1}	T_overspend_rate & TRD_overspend_rate	3000	0.692	0.000

Table B.23: Overspend Hypotheses Results for Behaviour Profile D (Correlations)

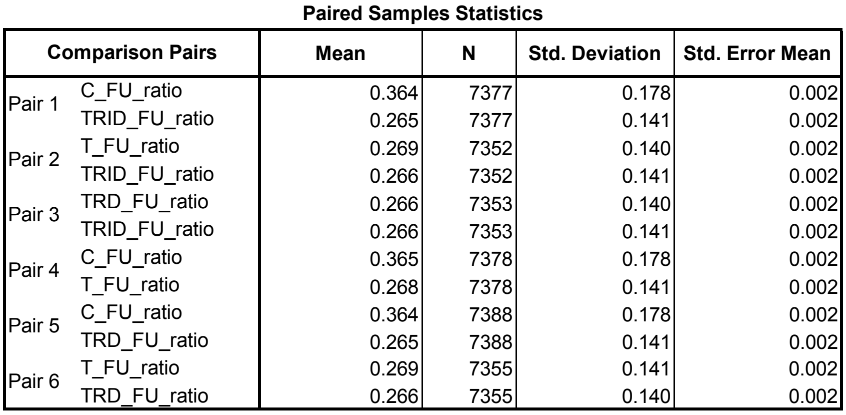


Paired Samples Test

Hypothesis	Mean Difference	Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
					Lower	Upper			
H_{Oa1}	C_overspend_rate - TRID_overspend_rate	0.036	0.068	0.001	0.034	0.039	29.381	2999	0.000
H_{Ob1}	T_overspend_rate - TRID_overspend_rate	0.001	0.050	0.001	-0.001	0.003	1.106	2999	0.269
H_{Oc1}	TRD_overspend_rate - TRID_overspend_rate	0.002	0.051	0.001	0.000	0.003	1.693	2999	0.090
H_{Od1}	C_overspend_rate - T_overspend_rate	0.035	0.067	0.001	0.033	0.038	29.048	2999	0.000
H_{Oe1}	C_overspend_rate - TRD_overspend_rate	0.035	0.068	0.001	0.032	0.037	27.933	2999	0.000
H_{Of1}	T_overspend_rate - TRD_overspend_rate	-0.001	0.050	0.001	-0.002	0.001	-0.615	2999	0.539

Table B.24: Overspend Hypotheses Results for Behaviour Profile D (Tests)

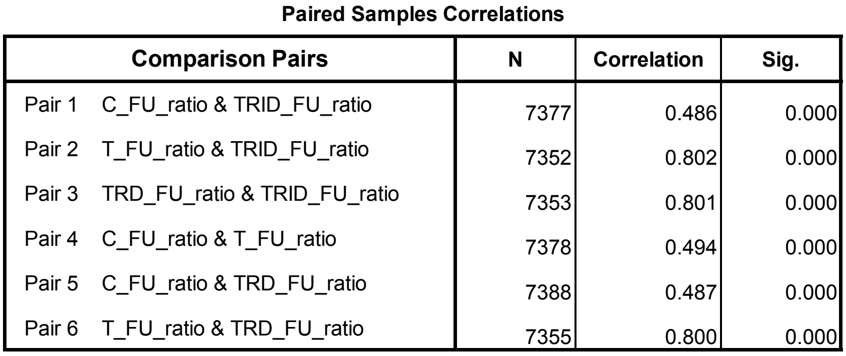
B.2 Paired t-Tests on Failed Task Ratios for Population Sizes 10 to 100



Paired Samples Statistics

Comparison Pairs		Mean	N	Std. Deviation	Std. Error Mean
Pair 1	C_FU_ratio	0.364	7377	0.178	0.002
	TRID_FU_ratio	0.265	7377	0.141	0.002
Pair 2	T_FU_ratio	0.269	7352	0.140	0.002
	TRID_FU_ratio	0.266	7352	0.141	0.002
Pair 3	TRD_FU_ratio	0.266	7353	0.140	0.002
	TRID_FU_ratio	0.266	7353	0.141	0.002
Pair 4	C_FU_ratio	0.365	7378	0.178	0.002
	T_FU_ratio	0.268	7378	0.141	0.002
Pair 5	C_FU_ratio	0.364	7388	0.178	0.002
	TRD_FU_ratio	0.265	7388	0.141	0.002
Pair 6	T_FU_ratio	0.269	7355	0.141	0.002
	TRD_FU_ratio	0.266	7355	0.140	0.002

Table B.25: Hypotheses Results for Behaviour Profile A, Population 1, Population 2, Population 3, Population Sizes 10 to 100 (Statistics)



Paired Samples Correlations

Comparison Pairs		N	Correlation	Sig.
Pair 1	C_FU_ratio & TRID_FU_ratio	7377	0.486	0.000
Pair 2	T_FU_ratio & TRID_FU_ratio	7352	0.802	0.000
Pair 3	TRD_FU_ratio & TRID_FU_ratio	7353	0.801	0.000
Pair 4	C_FU_ratio & T_FU_ratio	7378	0.494	0.000
Pair 5	C_FU_ratio & TRD_FU_ratio	7388	0.487	0.000
Pair 6	T_FU_ratio & TRD_FU_ratio	7355	0.800	0.000

Table B.26: Hypotheses Results for Behaviour Profile A, Population 1, Population 2, Population 3, Population Sizes 10 to 100 (Correlations)

Paired Samples Test

Mean Difference		Paired Differences					t	df	Sig. (2-tailed)
					95% Confidence Interval of the Difference				
		Mean	Std. Deviation	Std. Error Mean	Lower	Upper			
Pair 1	C_FU_rate - TRID_FU_rate	0.099	0.165	0.002	0.096	0.103	51.725	7376	0.000
Pair 2	T_FU_rate - TRID_FU_rate	0.003	0.088	0.001	0.001	0.005	2.714	7351	0.007
Pair 3	TRD_FU_rate - TRID_FU_rate	0.000	0.088	0.001	-0.002	0.002	-0.108	7352	0.914
Pair 4	C_FU_rate - T_FU_rate	0.097	0.164	0.002	0.093	0.100	50.693	7377	0.000
Pair 5	C_FU_rate - TRD_FU_rate	0.100	0.165	0.002	0.096	0.104	52.142	7387	0.000
Pair 6	T_FU_rate - TRD_FU_rate	0.003	0.089	0.001	0.001	0.005	2.816	7354	0.005

Table B.27: Hypotheses Results for Behaviour Profile A, Population 1, Population 2, Population 3, Population Sizes 10 to 100 (Test)

Appendix C

Collusion Detection Source Files

C.1 Interaction History File Example

Agent interaction history file and collusion detection information for population configuration (PopulationSize=10, HighTrust=0%, AverageTrust=0%, LowTrust=100%).

```
DS      18      s1      a1      a3      a4      252      648       -1
DS      18      s1      a1      a8      a4      272      528       -1
DA      18      a1      a2      a4      80      320      -1
DA      18      a1      a5      a7      16      254      -1
DS      18      s1      a1      a6      a7      178      632       -1
DS      18      s1      a1      a6      a8      72      198      -1
DA      18      a1      a2      a10      210      490      0
DS      18      s1      a1      a8      a10      32      48      0
A       18      s1      a1      a10      1
DS      38      s1      a1      a4      a7      140      260       -1
DS      38      s1      a1      a8      a7      105      375       -1
DS      38      s1      a1      a10      a7      20      60      -1
A       38      s1      a1      a4      1
DS      52      s1      a1      a4      a3      41      59      0
A       52      s1      a1      a3      1
DS      59      s1      a1      a4      a6      50      150      -1
DS      59      s1      a1      a3      a6      328      472       -1
IA      59      a1      a9      a2      a6      67      183      -1
DS      59      s1      a1      a4      a8      154      546       0
DA      59      a1      a5      a8      75      225      0
A       59      s1      a1      a8      0
DA      78      a1      a9      a3      50      130      -1
DA      78      a1      a5      a6      21      129      0
DS      78      s1      a1      a6      a7      296      334       -1
```

```
DA      78      a1      a5      a7      93      207      -1
DS      78      s1      a1      a8      a7      230      490      -1
DS      78      s1      a1      a7      a10      6       13      -1
A       78      s1      a1      a6      1
A       95      s1      a1      a10      0
IS      112      s1      a1      a9      a7      a8      9      41      -1
DS      112      s1      a1      a6      a10      225      675      -1
A       112      s1      a1      a7      0
DS      126      s1      a1      a10      a3      0      20      1
DS      126      s1      a1      a6      a3      118      242      1
DS      126      s1      a1      a8      a6      153      87      -1
DS      126      s1      a1      a10      a6      4      16      -1
DA      126      a1      a5      a7      36      264      -1
DA      126      a1      a2      a7      69      231      -1
DA      126      a1      a9      a7      50      370      -1
DA      126      a1      a9      a10      39      321      -1
DA      126      a1      a2      a10      119      581      -1
DS      126      s1      a1      a6      a10      233      397      -1
A       126      s1      a1      a3      0
DS      143      s1      a1      a6      a3      180      540      1
DA      143      a1      a2      a6      56      144      -1
DS      143      s1      a1      a8      a6      184      616      -1
IS      143      s1      a1      a9      a7      a6      18      82      -1
DS      143      s1      a1      a6      a10      42      48      -1
DS      143      s1      a1      a8      a10      23      57      -1
DS      143      s1      a1      a3      a10      189      711      -1
A       143      s1      a1      a3      0
DS      150      s1      a1      a3      a6      135      765      -1
DS      150      s1      a1      a7      a6      0      10      -1
DS      150      s1      a1      a8      a7      134      506      -1
A       150      s1      a1      a3      0
DS      157      s1      a1      a3      a10      80      420      -1
DA      157      a1      a5      a10      16      104      -1
DA      157      a1      a9      a10      113      427      -1
A       157      s1      a1      a7      1
A       168      s1      a1      a4      0
DS      187      s1      a1      a3      a6      140      360      -1
DA      187      a1      a5      a6      55      185      -1
DA      187      a1      a5      a7      9      51      1
A       187      s1      a1      a7      1
DS      194      s1      a1      a10      a3      1      59      -1
DA      194      a1      a9      a3      33      207      -1
DA      194      a1      a2      a3      210      790      -1
DA      194      a1      a2      a7      252      648      1
DS      194      s1      a1      a8      a7      216      264      1
DA      194      a1      a5      a7      19      221      1
DS      194      s1      a1      a6      a8      171      729      -1
DS      194      s1      a1      a3      a8      234      666      -1
A       194      s1      a1      a7      0
DA      200      a1      a5      a3      33      207      -1
DS      200      s1      a1      a10      a3      3      37      -1
DS      200      s1      a1      a7      a3      4      26      -1
```

```
DA      200     a1      a2      a7      140      560      1
DS      200     s1      a1      a4      a8       108      292      -1
A       200     s1      a1      a7      1
DA      219     a1      a9      a3      75       285      -1
IS      219     s1      a1      a9      a3       a6       200      600      -1
DS      219     s1      a1      a7      a6       23       47       -1
DS      219     s1      a1      a4      a6       19       81       -1
DA      219     a1      a5      a7      9       111      -1
DS      219     s1      a1      a6      a7       239      391      -1
DA      219     a1      a2      a7      238      462      -1
DS      219     s1      a1      a6      a10      184      266      -1
DS      219     s1      a1      a8      a10      105      135      -1
DA      219     a1      a2      a10      110      890      -1
A       219     s1      a1      a8      1
DS      233     s1      a1      a3      a4       155      345      1
DS      233     s1      a1      a6      a4       178      362      1
DA      233     a1      a5      a10      6       54      -1
A       233     s1      a1      a4      1
DA      252     a1      a2      a3      297      603      -1
DA      252     a1      a9      a3      43       197      -1
DS      252     s1      a1      a7      a3       2       18       -1
A       252     s1      a1      a8      1
DA      263     a1      a9      a3      40       200      1
DS      263     s1      a1      a4      a3       88       112      1
DS      263     s1      a1      a8      a3       91       149      1
DS      263     s1      a1      a4      a7       144      456      -1
DA      263     a1      a9      a7      126      474      -1
DS      263     s1      a1      a7      a10      6        23       -1
A       263     s1      a1      a3      0
DA      270     a1      a5      a6      55       185      -1
DA      270     a1      a2      a6      100      300      -1
A       270     s1      a1      a7      1
DS      279     s1      a1      a10      a3       6       54       -1
DS      279     s1      a1      a6      a3       52       128      -1
DA      279     a1      a9      a3      52       428      -1
DA      279     a1      a5      a8      9       81       -1
DS      279     s1      a1      a10      a8       3       27       -1
A       279     s1      a1      a4      0
DA      299     a1      a9      a3      10       170      -1
DS      299     s1      a1      a10      a3       1       89       -1
DS      299     s1      a1      a7      a4       2       48       -1
DS      299     s1      a1      a4      a6       80       920      -1
DS      299     s1      a1      a7      a6       5       35       -1
DA      299     a1      a2      a6      103      347      -1
DS      299     s1      a1      a6      a7       117      333      -1
DA      299     a1      a2      a7      161      539      -1
DS      299     s1      a1      a4      a8       80       120      1
DA      299     a1      a5      a8      2       208      1
DS      299     s1      a1      a10      a8       1       9       1
DA      299     a1      a5      a10      39       171      -1
DA      299     a1      a9      a10      162      438      -1
DS      299     s1      a1      a6      a10      86       274      -1
```

```
A         299        s1        a1        a8        0
DS        305        s1        a1        a7        a4        2        28        -1
DA        305        a1        a5        a4        48        222        -1
DS        305        s1        a1        a10        a6        4        66        1
DS        305        s1        a1        a3        a6        160        840        1
DS        305        s1        a1        a6        a10         210        600        -1
DS        305        s1        a1        a7        a10        8        8        -1
A         305        s1        a1        a6        0
DS        315        s1        a1        a7        a4        17        73        -1
DA        315        a1        a5        a4        21        159        -1
DS        315        s1        a1        a6        a4        75        285        -1
DS        315        s1        a1        a8        a6        14        66        1
A         315        s1        a1        a6        1
DS        333        s1        a1        a7        a3        8        92        -1
DS        333        s1        a1        a6        a3        26        64        -1
DA        333        a1        a5        a3        27        183        -1
IA        333        a1        a9        a5        a8        10        260        -1
DA        333        a1        a2        a8        70        430        -1
DS        333        s1        a1        a7        a8        7        93        -1
DS        333        s1        a1        a8        a10        259       461         -1
DS        333        s1        a1        a4        a10        28        172        -1
A         333        s1        a1        a6        0
DA        343        a1        a5        a4        16        164        -1
DS        343        s1        a1        a8        a4        190        370        -1
DS        343        s1        a1        a10        a6        2        78        1
A         343        s1        a1        a6        0
DS        350        s1        a1        a10        a3        10        90        1
DS        350        s1        a1        a8        a7        28        52        -1
DS        350        s1        a1        a3        a7        248        552        -1
DA        350        a1        a5        a7        42        258        -1
A         350        s1        a1        a3        0
DS        365        s1        a1        a6        a4        23        67        -1
DS        365        s1        a1        a7        a4        0        30        -1
DS        365        s1        a1        a8        a6        151        409        1
DS        365        s1        a1        a7        a6        3        47        1
DS        365        s1        a1        a3        a6        48        352        1
DS        365        s1        a1        a4        a7        48        352        -1
DS        365        s1        a1        a3        a7        102        198        -1
DA        365        a1        a5        a8        75        225        -1
A         365        s1        a1        a6        1
DS        375        s1        a1        a8        a3        201        279        -1
DS        375        s1        a1        a10        a3        8        52        -1
DS        375        s1        a1        a10        a4        8        92        -1
DA        375        a1        a9        a4        21        159        -1
A         375        s1        a1        a6        0
DS        391        s1        a1        a8        a4        40        200        -1
DS        391        s1        a1        a10        a4        3        37        -1
DA        391        a1        a5        a4        6        24        -1
IS        391        s1        a1        a9        a3        a8        54        546        -1
DA        391        a1        a2        a8        59        291        -1
DS        391        s1        a1        a7        a10        12        22         -1
DA        391        a1        a5        a10        6        24        -1
DA    391    a1    a2    a10    27    73    -1
A     391    s1    a1    a7     0
DS    408    s1    a1    a6     a3    309   411   -1
DS    408    s1    a1    a8     a3    24    56    -1
DA    408    a1    a9    a3     120   480   -1
DA    408    a1    a5    a7     14    166   -1
DS    408    s1    a1    a6     a7    171   729   -1
DA    408    a1    a2    a10    108   292   -1
DS    408    s1    a1    a8     a10   73    247   -1
A     408    s1    a1    a4     0
DA    421    a1    a9    a4     43    317   -1
DS    421    s1    a1    a8     a4    36    44    -1
DS    421    s1    a1    a10    a4    2     18    -1
DS    421    s1    a1    a3     a8    52    148   -1
DS    421    s1    a1    a6     a10   324   576   -1
DS    421    s1    a1    a7     a10   0     5     -1
A     421    s1    a1    a6     1
DA    441    a1    a9    a7     97    443   -1
A     441    s1    a1    a10    0
DA    460    a1    a5    a4     5     55    -1
DA    460    a1    a2    a4     216   684   -1
DS    460    s1    a1    a3     a4    150   350   -1
A     460    s1    a1    a8     1
DS    476    s1    a1    a8     a3    132   268   -1
DS    476    s1    a1    a4     a3    168   532   -1
DS    476    s1    a1    a8     a7    184   376   -1
DS    476    s1    a1    a4     a7    132   168   -1
DA    476    a1    a2    a7     52    348   -1
DS    476    s1    a1    a6     a8    201   519   1
IS    476    s1    a1    a9     a3    a8    215   285   1
A     476    s1    a1    a8     0
A     485    s1    a1    a10    0
A     490    s1    a1    a7     0
DA    504    a1    a5    a7     10    260   -1
DS    504    s1    a1    a3     a7    24    76    -1
DS    504    s1    a1    a10    a7    0     30    -1
DS    504    s1    a1    a6     a10   81    369   1
DA    504    a1    a2    a10    200   600   1
DS    504    s1    a1    a4     a10   126   474   1
A     504    s1    a1    a10    0
DS    523    s1    a1    a8     a4    79    161   -1
DA    523    a1    a5    a4     10    140   -1
DS    523    s1    a1    a4     a6    216   384   -1
A     523    s1    a1    a8     0
DA    542    a1    a5    a4     16    194   -1
DS    542    s1    a1    a8     a4    108   292   -1
DS    542    s1    a1    a6     a4    91    449   -1
DS    542    s1    a1    a10    a6    0     10    -1
DA    542    a1    a2    a6     60    140   -1
A     542    s1    a1    a7     0
DA    552    a1    a5    a3     8     112   -1
DA    552    a1    a9    a3     9     231   -1
```

```
DS        552        s1        a1        a7        a3        14        86        -1
DS        552        s1        a1        a7        a6        11        49        -1
DA        552        a1        a5        a6        9       141         -1
DA        552        a1        a5        a8        19        41        1
DS        552        s1        a1        a7        a8        1        19        1
A         552        s1        a1        a8        1
DS        565        s1        a1        a8        a3        86        234        -1
DS        565        s1        a1        a4        a3        180        720        -1
DA        565        a1        a2        a7        164        236        -1
DS        565        s1        a1        a3        a7        189        711        -1
DS        565        s1        a1        a4        a7        35        65        -1
DS        565        s1        a1        a6        a8        25        155        -1
DS        565        s1        a1        a4        a8        312        488        -1
DS        565        s1        a1        a10        a8        16        64        -1
A         565        s1        a1        a6        0
DS        583        s1        a1        a3        a4        203        497        -1
DS        583        s1        a1        a8        a4        168        392        -1
DS        583        s1        a1        a6        a4        11        79        -1
DA        583        a1        a2        a6        45        205        -1
DA        583        a1        a5        a6        5       265         -1
DS        583        s1        a1        a3        a6        120        480        -1
IA        583        a1        a9        a2        a8        32        368        -1
DS        583        s1        a1        a7        a8        5        75        -1
DS        583        s1        a1        a8        a10        50        670        -1
DA        583        a1        a9        a10        30        570        -1
A         583        s1        a1        a3        1
DA        600        a1        a2        a3        160        240        -1
DS        600        s1        a1        a10        a4        3        47        -1
DS        600        s1        a1        a6        a4        194        166        -1
DA        600        a1        a9        a4        8       112         -1
A         600        s1        a1        a8        0
DS        620        s1        a1        a6        a3        104        256        -1
DS        620        s1        a1        a8        a7        20        60        1
DS        620        s1        a1        a4        a7        102        198        1
DS        620        s1        a1        a6        a7        72        108        1
A         620        s1        a1        a7        0
DS        639        s1        a1        a6        a3        226        404        -1
DS        639        s1        a1        a4        a3        98        602        -1
DS        639        s1        a1        a10        a4        13        47        -1
DS        639        s1        a1        a4        a6        104        296        -1
DS        639        s1        a1        a8        a7        224        416        -1
DA        639        a1        a9        a10        24        276        -1
DA        639        a1        a5        a10        10        140        -1
A         639        s1        a1        a8        1
DS        656        s1        a1        a4        a3        195        305        -1
DA        656        a1        a2        a4        72        828        -1
DS        656        s1        a1        a10        a4        5        45        -1
DS        656        s1        a1        a8        a4        74        166        -1
DA        656        a1        a2        a6        8       392         -1
IS        656        s1        a1        a9        a7        a6        13        67        -1
DS        656        s1        a1        a3        a6        20        380        -1
DA        656        a1        a2        a10        24        576        -1
```

```
A       656     s1      a1      a7      1
DA      674     a1      a2      a3      200      600      -1
DS      674     s1      a1      a8      a4      96      384      -1
DS      674     s1      a1      a6      a7      105      165      -1
A       674     s1      a1      a6      0
DS      686     s1      a1      a10      a3      9      61      -1
DS      686     s1      a1      a7      a3      2      68      -1
DA      686     a1      a2      a3      90      210      -1
DA      686     a1      a5      a10      3      27      1
A       686     s1      a1      a10      0
DS      706     s1      a1      a10      a7      0      10      -1
DS      706     s1      a1      a4      a7      84      516      -1
DS      706     s1      a1      a3      a7      63      237      -1
DS      706     s1      a1      a4      a8      74      126      1
DA      706     a1      a2      a8      56      294      1
DS      706     s1      a1      a6      a8      27      63      1
A       706     s1      a1      a8      1
A       712     s1      a1      a10      0
DA      731     a1      a5      a4      9      81      -1
DS      731     s1      a1      a6      a4      129      231      -1
DS      731     s1      a1      a4      a7      155      345      -1
DS      731     s1      a1      a6      a7      252      378      -1
DS      731     s1      a1      a7      a10      8      8      -1
DA      731     a1      a9      a10      19      41      -1
DA      731     a1      a5      a10      3      27      -1
A       731     s1      a1      a3      0
DA      749     a1      a9      a3      14      106      -1
DS      749     s1      a1      a10      a3      3      47      -1
DS      749     s1      a1      a8      a4      31      209      1
DS      749     s1      a1      a10      a6      7      33      -1
DS      749     s1      a1      a3      a10      66      534      -1
A       749     s1      a1      a4      0
DS      756     s1      a1      a10      a4      3      77      -1
DA      756     a1      a9      a4      52      428      -1
DS      756     s1      a1      a3      a4      13      87      -1
A       756     s1      a1      a8      0
DS      763     s1      a1      a8      a3      230      490      1
DA      763     a1      a9      a3      11      49      1
DS      763     s1      a1      a4      a7      68      332      -1
DA      763     a1      a9      a7      84      516      -1
DA      763     a1      a2      a7      15      85      -1
DA      763     a1      a9      a10      30      150      -1
DS      763     s1      a1      a6      a10      36      144      -1
DS      763     s1      a1      a7      a10      4      14      -1
A       763     s1      a1      a3      0
DA      771     a1      a2      a10      56      744      -1
DS      771     s1      a1      a7      a10      20      30      -1
DS      771     s1      a1      a4      a10      272      528      -1
A       771     s1      a1      a3      1
DA      789     a1      a5      a3      10      50      1
DA      789     a1      a2      a3      48      352      1
A       789     s1      a1      a3      1
```

```
DA    805    a1    a9    a4    7      53     -1
A     805    s1    a1    a3    0
DA    818    a1    a5    a3    29     61     -1
DA    818    a1    a2    a3    110    390    -1
DS    818    s1    a1    a7    a3     9      91     -1
DS    818    s1    a1    a7    a4     8      82     -1
DS    818    s1    a1    a10   a4     19     81     -1
IS    818    s1    a1    a9    a8     a6     30     50     -1
DA    818    a1    a2    a6    60     140    -1
DA    818    a1    a5    a6    19     161    -1
DA    818    a1    a5    a7    46     134    -1
DS    818    s1    a1    a4    a10    210    290    1
A     818    s1    a1    a10   0
DS    826    s1    a1    a10   a8     3      67     -1
IA    826    a1    a9    a5    a8     3      27     -1
DA    826    a1    a5    a8    6      84     -1
DA    826    a1    a2    a10   258    342    -1
DA    826    a1    a5    a10   10     80     -1
DS    826    s1    a1    a4    a10    273    427    -1
A     826    s1    a1    a6    0
DA    832    a1    a5    a4    40     230    -1
IA    832    a1    a9    a2    a8     4      96     1
A     832    s1    a1    a8    1
DA    843    a1    a5    a4    37     233    -1
DS    843    s1    a1    a6    a8     36     144    -1
DS    843    s1    a1    a4    a8     27     273    -1
DS    843    s1    a1    a4    a10    33     267    -1
A     843    s1    a1    a3    1
DS    863    s1    a1    a6    a4     220    410    -1
DA    863    a1    a2    a4    126    574    -1
DA    863    a1    a9    a4    37     383    -1
DA    863    a1    a2    a6    64     136    -1
DS    863    s1    a1    a7    a6     11     69     -1
DA    863    a1    a5    a10   3      87     -1
DS    863    s1    a1    a6    a10    130    320    -1
A     863    s1    a1    a7    0
DS    869    s1    a1    a3    a10    96     504    -1
DA    869    a1    a9    a10   67     413    -1
A     869    s1    a1    a3    1
DA    884    a1    a2    a3    105    195    -1
DS    884    s1    a1    a6    a3     25     65     -1
DS    884    s1    a1    a3    a6     84     216    1
DS    884    s1    a1    a3    a10    57     243    -1
DA    884    a1    a2    a10   39     261    -1
DS    884    s1    a1    a4    a10    294    406    -1
A     884    s1    a1    a6    1
DS    889    s1    a1    a10   a8     17     73     1
DS    889    s1    a1    a8    a10    108    292    -1
DS    889    s1    a1    a6    a10    186    624    -1
A     889    s1    a1    a8    1
DS    899    s1    a1    a10   a6     5      75     -1
DS    899    s1    a1    a4    a6     204    396    -1
DS        899       s1        a1        a8        a6        140        180        -1
DS        899       s1        a1        a7        a10        4        9        1
A         899       s1        a1        a10        1
DA        906       a1        a2        a3        390        610        -1
DA        906       a1        a5        a3        27        123        -1
DS        906       s1        a1        a4        a6        56        144        -1
DA        906       a1        a5        a6        14        166        -1
DA        906       a1        a5        a7        24        246        -1
DA        906       a1        a2        a7        220        780        -1
DS        906       s1        a1        a8        a10        153        487        -1
A         906       s1        a1        a8        0
DS        915       s1        a1        a4        a10        93        207        1
DS        915       s1        a1        a8        a10        81        159        1
A         915       s1        a1        a10        1
DS        930       s1        a1        a8        a3        129        591        -1
DS        930       s1        a1        a10        a3        2        38        -1
DS        930       s1        a1        a3        a10        550        450        -1
DA        930       a1        a5        a10        32        88        -1
DA        930       a1        a9        a10        79        341        -1
A         930       s1        a1        a6        1
DA        947       a1        a5        a4        2        118        -1
DS        947       s1        a1        a8        a4        156        404        -1
DS        947       s1        a1        a3        a4        48        352        -1
DS        947       s1        a1        a6        a8        43        137        -1
DS        947       s1        a1        a10        a8        5        55        -1
DS        947       s1        a1        a4        a8        140        360        -1
A         947       s1        a1        a6        0
DA        965       a1        a5        a10        45        135        -1
DS        965       s1        a1        a7        a10        2        24        -1
DS        965       s1        a1        a8        a10        33        207        -1
A         965       s1        a1        a4        1
DS        973       s1        a1        a3        a4        112        588        -1
DS        973       s1        a1        a3        a10        160        840        1
DA        973       a1        a5        a10        35        235        1
A         973       s1        a1        a10        0
DA        980       a1        a9        a7        24        96        -1
DS        980       s1        a1        a10        a7        2        38        -1
DA        980       a1        a2        a7        77        623        -1
A         980       s1        a1        a8        1
DS        989       s1        a1        a4        a3        189        711        -1
DA        989       a1        a9        a3        0        300        -1
DS        989       s1        a1        a8        a3        128        272        -1
DA        989       a1        a2        a4        64        736        -1
DS        989       s1        a1        a10        a4        3        27        -1
DA        989       a1        a5        a4        6        24        -1
DS        989       s1        a1        a3        a6        14        86        1
DA        989       a1        a5        a6        16        164        1
DS        989       s1        a1        a8        a7        44        116        -1
DS        989       s1        a1        a10        a7        3        37        -1
DA        989       a1        a9        a10        12        108        -1
A         989       s1        a1        a6        1
DA        1008        a1        a5        a7        5        25        -1
```

```
DS        1008        s1        a1        a4        a7        70        130        -1
DS        1008        s1        a1        a6        a7        214         416        -1
A         1008        s1        a1        a3        1
DS        1019        s1        a1        a4        a3        78        222        -1
DS        1019        s1        a1        a3        a4        84        216        -1
DS        1019        s1        a1        a8        a4        137         183        -1
DA        1019        a1        a5        a6        10        170        -1
DS        1019        s1        a1        a4        a6        44        156        -1
DS        1019        s1        a1        a3        a8        60        440        1
DS        1019        s1        a1        a7        a8        4        16        1
A         1019        s1        a1        a8        0
DS        1025        s1        a1        a7        a6        6        24        -1
DS        1025        s1        a1        a10        a8        22        68        -1
DA        1025        a1        a5        a8        10        110        -1
A         1025        s1        a1        a4        1
A         1037        s1        a1        a10        0
A         1057        s1        a1        a4        0
A         1067        s1        a1        a6        0
DS        1075        s1        a1        a10        a6        28        72        -1
DA        1075        a1        a2        a6        60        440        -1
DS        1075        s1        a1        a10        a7        7        73        1
DA        1075        a1        a5        a10        3        27        -1
A         1075        s1        a1        a7        1
DS        1082        s1        a1        a10        a4        4        46        1
DS        1082        s1        a1        a6        a4        119         511        1
DA        1082        a1        a5        a6        1        29        -1
IA        1082        a1        a9        a2        a6        23        77        -1
DA        1082        a1        a2        a6        3        47        -1
DS        1082        s1        a1        a6        a8        104         256        -1
A         1082        s1        a1        a4        0
DS        1097        s1        a1        a6        a3        223         497        -1
DA        1097        a1        a2        a4        48        552        1
DA        1097        a1        a9        a4        93        207        1
DS        1097        s1        a1        a3        a4        182         518        1
DS        1097        s1        a1        a8        a7        80        320        -1
DA        1097        a1        a5        a8        45        195        -1
DA        1097        a1        a2        a8        17        33        -1
IS        1097        s1        a1        a9        a7        a8        12        18        -1
A         1097        s1        a1        a4        0
A         1113        s1        a1        a7        1
DS        1128        s1        a1        a6        a4        372         438        -1
DS        1128        s1        a1        a8        a4        216         504        -1
DS        1128        s1        a1        a3        a4        161         539        -1
IS        1128        s1        a1        a9        a7        a8        18        72        -1
A         1128        s1        a1        a6        0
DS        1134        s1        a1        a10        a8        4        36        -1
DS        1134        s1        a1        a3        a8        26        174        -1
A         1134        s1        a1        a10        0
DS        1150        s1        a1        a7        a3        9        41        -1
DA        1150        a1        a5        a3        3        177        -1
DS        1150        s1        a1        a6        a3        167         373        -1
DS        1150        s1        a1        a10        a6        7        53        -1
DS        1150        s1        a1        a4        a6        90        210        -1
A         1150        s1        a1        a10        1
DS        1166        s1        a1        a7        a3        1        9        -1
DS        1166        s1        a1        a8        a3        48        112        -1
DS        1166        s1        a1        a10        a3        3        17        -1
DA        1166        a1        a2        a4        5        495        -1
DS        1166        s1        a1        a8        a4        40        760        -1
DS        1166        s1        a1        a10        a6        3        27        -1
DS        1166        s1        a1        a3        a10        400        600        -1
A         1166        s1        a1        a7        1
DA        1171        a1        a9        a3        147        273        -1
DA        1171        a1        a5        a3        27        183        -1
DS        1171        s1        a1        a4        a3        231        569        -1
DS        1171        s1        a1        a7        a6        1        9        -1
DS        1171        s1        a1        a4        a6        250        250        -1
DS        1171        s1        a1        a8        a6        128        192        -1
DS        1171        s1        a1        a3        a7        170        830        -1
DA        1171        a1        a9        a7        62        418        -1
DS        1171        s1        a1        a4        a7        130        370        -1
DS        1171        s1        a1        a8        a10        44        276        1
A         1171        s1        a1        a10        1
DS        1181        s1        a1        a4        a3        252        448        -1
DA        1181        a1        a5        a6        54        156        -1
DS        1181        s1        a1        a10        a7        9        81        1
DS        1181        s1        a1        a4        a7        240        760        1
DS        1181        s1        a1        a8        a10        240        320        -1
DA        1181        a1        a5        a10        3        177        -1
DS        1181        s1        a1        a3        a10        48        152        -1
A         1181        s1        a1        a7        1
DA        1189        a1        a2        a3        387        513        -1
DS        1189        s1        a1        a6        a3        170        640        -1
DS        1189        s1        a1        a7        a3        12        78        -1
DA        1189        a1        a9        a4        67        173        1
DA        1189        a1        a5        a7        1        29        -1
DA        1189        a1        a9        a7        58        362        -1
DS        1189        s1        a1        a3        a7        115        385        -1
DS        1189        s1        a1        a4        a10        360        540        -1
DA        1189        a1        a5        a10        4        56        -1
A         1189        s1        a1        a4        0
DS        1195        s1        a1        a6        a3        162        738        -1
DS        1195        s1        a1        a8        a3        100        380        -1
DS        1195        s1        a1        a7        a6        3        27        1
DS        1195        s1        a1        a4        a6        36        64        1
IS        1195        s1        a1        a9        a8        a6        136        264        1
A         1195        s1        a1        a6        1
DS        1200        s1        a1        a8        a3        136        264        -1
A         1200        s1        a1        a8        0
DA        1214        a1        a2        a4        93        207        -1
IS        1214        s1        a1        a9        a3        a8        160        340        -1
DA        1214        a1        a5        a8        3        27        -1
DA        1214        a1        a5        a10        40        230        1
DS        1214        s1        a1        a3        a10        145        355        1
```

```
DA      1214     a1      a2      a10      25      75      1
A       1214     s1      a1      a10      0
IS      1223     s1      a1      a9      a3      a8      110      890      -1
DA      1223     a1      a5      a8      12      78      -1
A       1223     s1      a1      a6      1
DA      1242     a1      a2      a6      47      203      -1
DS      1242     s1      a1      a3      a6      49      651      -1
DS      1242     s1      a1      a8      a10      44      356      -1
A       1242     s1      a1      a4      0
DS      1254     s1      a1      a7      a3      2      8      -1
DA      1254     a1      a5      a3      9      51      -1
DS      1254     s1      a1      a8      a3      396      324      -1
A       1254     s1      a1      a7      0
DS      1261     s1      a1      a10      a6      2      18      -1
DS      1261     s1      a1      a7      a8      4      36      -1
DS      1261     s1      a1      a10      a8      9      61      -1
A       1261     s1      a1      a4      0
DS      1271     s1      a1      a3      a6      222      378      -1
DA      1271     a1      a5      a6      4      146      -1
DS      1271     s1      a1      a7      a6      0      20      -1
DS      1271     s1      a1      a6      a8      43      227      1
DA      1271     a1      a5      a8      3      297      1
A       1271     s1      a1      a8      1
DS      1285     s1      a1      a6      a3      19      161      -1
DA      1285     a1      a9      a3      142      278      -1
DS      1285     s1      a1      a8      a3      235      325      -1
DS      1285     s1      a1      a7      a6      1      59      -1
DS      1285     s1      a1      a3      a6      120      380      -1
DS      1285     s1      a1      a10      a7      2      18      -1
DS      1285     s1      a1      a4      a8      99      201      1
DS      1285     s1      a1      a8      a10      105      375      -1
A       1285     s1      a1      a8      1
DA      1296     a1      a2      a4      46      154      -1
IS      1296     s1      a1      a9      a3      a8      276      324      -1
DS      1296     s1      a1      a3      a8      104      696      -1
DS      1296     s1      a1      a8      a10      316      404      -1
DS      1296     s1      a1      a6      a10      124      416      -1
DA      1296     a1      a9      a10      14      166      -1
A       1296     s1      a1      a3      1
DA      1307     a1      a5      a7      37      143      -1
A       1307     s1      a1      a3      1
DS      1320     s1      a1      a4      a3      69      231      -1
DS      1320     s1      a1      a8      a3      338      382      -1
DS      1320     s1      a1      a3      a4      45      455      -1
DA      1320     a1      a2      a4      25      75      -1
A       1320     s1      a1      a6      1
DA      1328     a1      a2      a4      76      324      -1
DS      1328     s1      a1      a7      a6      8      52      1
DA      1328     a1      a9      a7      39      141      -1
DS      1328     s1      a1      a10      a7      2      8      -1
DA      1328     a1      a2      a7      115      285      -1
DS      1328     s1      a1      a3      a8      240      760      -1
DS    1328    s1    a1    a10    a8     0      10     -1
A     1328    s1    a1    a6     0
DS    1334    s1    a1    a4     a6     84     116     -1
DS    1334    s1    a1    a7     a6     7     43     -1
DA    1334    a1    a9    a7     135     405     1
DA    1334    a1    a2    a10    154     546     -1
DS    1334    s1    a1    a8     a10     237     483     -1
DS    1334    s1    a1    a7     a10     2     29     -1
A     1334    s1    a1    a7     1
DS    1350    s1    a1    a4     a6     280     520     -1
DS    1350    s1    a1    a8     a6     165     555     -1
DS    1350    s1    a1    a7     a6     11     89     -1
A     1350    s1    a1    a3     1
DS    1368    s1    a1    a3     a6     80     320     -1
DA    1368    a1    a2    a6     81     369     -1
DA    1368    a1    a5    a6     37     233     -1
DA    1368    a1    a5    a7     18     102     -1
DS    1368    s1    a1    a8     a7     128     272     -1
DS    1368    s1    a1    a3     a7     208     592     -1
A     1368    s1    a1    a10     0
DS    1385    s1    a1    a8     a4     179     461     -1
DA    1385    a1    a5    a4     10     170     -1
DS    1385    s1    a1    a6     a4     54     216     -1
IA    1385    a1    a9    a5     a6     8     262     1
DS    1385    s1    a1    a7     a6     2     18     1
DA    1385    a1    a5    a6     14     166     1
DA    1385    a1    a2    a7     26     74     -1
A     1385    s1    a1    a6     1
DS    1392    s1    a1    a4     a3     190     810     -1
DA    1392    a1    a2    a3     150     850     -1
A     1392    s1    a1    a4     0
DS    1399    s1    a1    a4     a8     330     670     1
DA    1399    a1    a2    a8     87     263     1
A     1399    s1    a1    a8     0
DS    1404    s1    a1    a10     a3     7     43     -1
DS    1404    s1    a1    a7     a10     16     11     -1
DA    1404    a1    a2    a10     30     170     -1
DA    1404    a1    a5    a10     15     45     -1
A     1404    s1    a1    a8     0
DS    1421    s1    a1    a8     a3     112     288     1
DA    1421    a1    a2    a3     70     630     1
DS    1421    s1    a1    a3     a4     60     540     -1
DS    1421    s1    a1    a7     a4     4     36     -1
DA    1421    a1    a2    a6     55     195     -1
DA    1421    a1    a2    a10     162     738     -1
DS    1421    s1    a1    a7     a10     30     37     -1
DS    1421    s1    a1    a8     a10     67     173     -1
A     1421    s1    a1    a3     0
DS    1428    s1    a1    a6     a4     16     74     -1
DS    1428    s1    a1    a7     a4     3     27     -1
DS    1428    s1    a1    a10     a7     2     48     -1
DS    1428    s1    a1    a10     a8     4     36     -1
```

```
DS	1428	s1	a1	a3	a8	85	415	-1
DA	1428	a1	a2	a10	44	356	-1
A	1428	s1	a1	a6	0
DS	1442	s1	a1	a8	a6	84	156	-1
DS	1442	s1	a1	a6	a7	157	293	-1
DS	1442	s1	a1	a10	a7	3	67	-1
DA	1442	a1	a5	a8	0	30	1
IA	1442	a1	a9	a5	a8	7	83	1
DS	1442	s1	a1	a3	a8	390	610	1
A	1442	s1	a1	a8	1
DA	1462	a1	a2	a4	50	450	-1
DS	1462	s1	a1	a6	a4	66	114	-1
DS	1462	s1	a1	a4	a7	258	342	1
A	1462	s1	a1	a7	1
DS	1481	s1	a1	a4	a3	297	603	-1
DS	1481	s1	a1	a7	a3	6	54	-1
DS	1481	s1	a1	a7	a4	3	37	-1
DA	1481	a1	a9	a4	10	110	-1
DS	1481	s1	a1	a8	a4	32	48	-1
DA	1481	a1	a9	a7	3	357	-1
DS	1481	s1	a1	a3	a7	180	720	-1
DS	1481	s1	a1	a6	a8	27	63	-1
DS	1481	s1	a1	a3	a8	90	510	-1
DS	1481	s1	a1	a3	a10	26	74	1
DA	1481	a1	a5	a10	9	81	1
DA	1481	a1	a9	a10	21	519	1
A	1481	s1	a1	a10	1
DS	1493	s1	a1	a7	a8	7	63	-1
DS	1493	s1	a1	a3	a8	55	445	-1
DS	1493	s1	a1	a3	a10	42	258	-1
DA	1493	a1	a2	a10	175	525	-1
DS	1493	s1	a1	a8	a10	192	448	-1
A	1493	s1	a1	a6	0
DA	1505	a1	a2	a3	81	819	-1
DA	1505	a1	a9	a3	67	413	-1
DS	1505	s1	a1	a7	a3	13	47	-1
DS	1505	s1	a1	a4	a10	66	234	-1
DA	1505	a1	a9	a10	36	564	-1
A	1505	s1	a1	a7	1
DS	1511	s1	a1	a10	a4	0	20	-1
DA	1511	a1	a2	a7	150	350	-1
DS	1511	s1	a1	a4	a8	168	432	-1
DA	1511	a1	a5	a10	19	221	-1
DA	1511	a1	a9	a10	10	50	-1
A	1511	s1	a1	a3	0
DA	1525	a1	a2	a4	35	465	-1
DS	1525	s1	a1	a3	a6	30	270	-1
DS	1525	s1	a1	a7	a6	1	49	-1
A	1525	s1	a1	a8	1
DS	1545	s1	a1	a10	a4	12	28	-1
DS	1545	s1	a1	a7	a4	7	63	-1
DS	1545	s1	a1	a8	a6	150	170	-1
```

```
DA      1545     s1       a5       a6       1        149      -1
DS      1545     s1       a1       a8       a7        187      293      1
A       1545     s1       a1       a7       1
DA      1563     a1       a5       a4       56        154      -1
DS      1563     s1       a1       a6       a4        201      519      -1
DS      1563     s1       a1       a10       a8        18       62      1
A       1563     s1       a1       a8       1
DA      1580     a1       a9       a4       12        228      1
DS      1580     s1       a1       a6       a4        72       378      1
DS      1580     s1       a1       a8       a4        139      341      1
DS      1580     s1       a1       a10       a6        10       80      -1
IS      1580     s1       a1       a9       a4        a6       245      255       -1
DS      1580     s1       a1       a8       a6        288      432      -1
DS      1580     s1       a1       a10       a8        13       67      -1
DS      1580     s1       a1       a7       a8       1        49       -1
DS      1580     s1       a1       a7       a10        18       25      -1
DA      1580     a1       a2       a10       168        532      -1
DA      1580     a1       a5       a10       4        26       -1
A       1580     s1       a1       a4       0
DA      1595     a1       a9       a3       4        56       -1
DS      1595     s1       a1       a10       a7        18       42      -1
A       1595     s1       a1       a6       0
DA      1605     a1       a2       a7       90        810      -1
DS      1605     s1       a1       a10       a7        10       80      -1
A       1605     s1       a1       a6       0
DA      1619     a1       a2       a7       200        800      -1
DS      1619     s1       a1       a8       a7        72       328      -1
DS      1619     s1       a1       a6       a7        32       148      -1
DA      1619     a1       a2       a8       170        330      -1
DA      1619     a1       a9       a10       33        207      -1
DS      1619     s1       a1       a6       a10        423      477      -1
DA      1619     a1       a2       a10       57        243      -1
A       1619     s1       a1       a6       0
DS      1633     s1       a1       a10       a3        5       95      1
DA      1633     a1       a9       a3       36        204      1
DS      1633     s1       a1       a8       a4        91       149      -1
DS      1633     s1       a1       a3       a7        280      520      -1
A       1633     s1       a1       a3       0
DS      1640     s1       a1       a4       a8        188      212      -1
DS      1640     s1       a1       a3       a8        21       79      -1
A       1640     s1       a1       a4       1
DA      1658     a1       a2       a8       69        231      -1
DS      1658     s1       a1       a3       a8        122      178      -1
DS      1658     s1       a1       a6       a8        296      334      -1
A       1658     s1       a1       a10       0
DA      1671     a1       a5       a3       5        25       -1
DS      1671     s1       a1       a10       a6        4       16      -1
DS      1671     s1       a1       a7       a6       0        50       -1
DA      1671     a1       a2       a6       9        41       -1
DS      1671     s1       a1       a3       a8        38       162      -1
IA      1671     a1       a9       a2       a8        76       324      -1
A       1671     s1       a1       a7       1
```

```
DS    1687    s1    a1    a3    a6    21    79    -1
IA    1687    a1    a9    a2    a6    57    193    -1
DA    1687    a1    a2    a6    63    237    -1
DA    1687    a1    a2    a8    91    259    1
IA    1687    a1    a9    a2    a8    32    168    1
DS    1687    s1    a1    a10    a8    15    85    1
A     1687    s1    a1    a8    1
A     1699    s1    a1    a6    0
DA    1717    a1    a2    a6    65    185    1
DS    1717    s1    a1    a10    a6    1    9    1
IS    1717    s1    a1    a9    a4    a6    200    600    1
DS    1717    s1    a1    a3    a7    216    684    -1
A     1717    s1    a1    a6    1
DS    1733    s1    a1    a8    a4    121    199    -1
DS    1733    s1    a1    a10    a4    13    67    -1
DA    1733    a1    a9    a4    96    324    -1
DS    1733    s1    a1    a3    a10    210    790    1
DA    1733    a1    a5    a10    3    117    1
DS    1733    s1    a1    a8    a10    121    199    1
A     1733    s1    a1    a10    1
DS    1752    s1    a1    a8    a3    22    58    -1
IA    1752    a1    a9    a5    a8    33    267    -1
DS    1752    s1    a1    a4    a8    168    532    -1
DS    1752    s1    a1    a7    a8    14    76    -1
A     1752    s1    a1    a10    0
DS    1761    s1    a1    a7    a3    5    15    -1
DA    1761    a1    a5    a3    60    240    -1
DS    1761    s1    a1    a10    a6    3    27    -1
A     1761    s1    a1    a8    1
DA    1774    a1    a5    a3    54    246    1
DS    1774    s1    a1    a4    a8    23    77    -1
DS    1774    s1    a1    a6    a8    405    495    -1
A     1774    s1    a1    a3    1
DS    1779    s1    a1    a6    a7    351    549    -1
DS    1779    s1    a1    a8    a7    172    468    -1
DS    1779    s1    a1    a3    a8    210    490    -1
A     1779    s1    a1    a3    0
DS    1795    s1    a1    a10    a6    13    87    -1
IS    1795    s1    a1    a9    a8    a6    62    178    -1
DS    1795    s1    a1    a4    a6    80    420    -1
A     1795    s1    a1    a8    1
DA    1806    a1    a2    a3    104    296    -1
DS    1806    s1    a1    a7    a4    2    38    -1
DA    1806    a1    a2    a4    108    792    -1
DS    1806    s1    a1    a4    a6    108    192    -1
DA    1806    a1    a2    a7    87    213    -1
DS    1806    s1    a1    a4    a7    243    657    -1
DS    1806    s1    a1    a10    a7    18    42    -1
DA    1806    a1    a5    a10    42    168    -1
DA    1806    a1    a2    a10    135    765    -1
A     1806    s1    a1    a8    0
DS    1816    s1    a1    a8    a3    201    359    1
```

```
DA        1816        a1        a5        a3        4        56        1
DS        1816        s1        a1        a6        a3        52        128        1
DS        1816        s1        a1        a8        a6        73        87        -1
DA        1816        a1        a5        a8        1        89       -1
DS        1816        s1        a1        a10        a8        0        10        -1
DS        1816        s1        a1        a6        a8        129         411       -1
A         1816        s1        a1        a3        1
DS        1830        s1        a1        a3        a6        45        855        1
DS        1830        s1        a1        a4        a7        330        670        -1
DS        1830        s1        a1        a7        a8        17        83        -1
A         1830        s1        a1        a6        0
DA        1843        a1        a2        a3        24        76        -1
DA        1843        a1        a9        a3        18        102        -1
DS        1843        s1        a1        a10        a4        9        81        -1
DS        1843        s1        a1        a4        a6        50        150        -1
DS        1843        s1        a1        a10        a7        6        44        1
DS        1843        s1        a1        a6        a7        118        242       1
DS        1843        s1        a1        a8        a7        128        192       1
DS        1843        s1        a1        a6        a8        50        670        -1
DA        1843        a1        a5        a8        27        243        -1
A         1843        s1        a1        a7        0
DA        1863        a1        a9        a3        70        470        -1
DS        1863        s1        a1        a6        a4        189        351       1
DS        1863        s1        a1        a10        a4        16        44        1
DS        1863        s1        a1        a8        a10        56        344        -1
DS        1863        s1        a1        a3        a10        75        225        -1
DS        1863        s1        a1        a6        a10        135        315        -1
A         1863        s1        a1        a4        0
DA        1872        a1        a2        a4        105        595        -1
DS        1872        s1        a1        a6        a4        34        56        -1
DS        1872        s1        a1        a10        a4        0        40        -1
A         1872        s1        a1        a6        1
DS        1879        s1        a1        a7        a3        3        27        1
DA        1879        a1        a9        a3        126        174        1
DS        1879        s1        a1        a8        a3        153        487       1
DA        1879        a1        a5        a4        1        59       -1
DS        1879        s1        a1        a7        a6        6        24        -1
DS        1879        s1        a1        a8        a6        257        303        -1
DA        1879        a1        a5        a8        46        134       -1
IS        1879        s1        a1        a9        a6        a8        216        504        -1
A         1879        s1        a1        a3        0
A         1888        s1        a1        a10        1
DS        1896        s1        a1        a6        a3        208        512        -1
DA        1896        a1        a9        a3        62        418       -1
DA        1896        a1        a2        a3        130        370        -1
DS        1896        s1        a1        a10        a4        3        57        -1
A         1896        s1        a1        a6        0
DA        1914        a1        a2        a4        174        426        -1
DS        1914        s1        a1        a7        a4        1        79        -1
DA        1914        a1        a5        a7        14        166        -1
IS        1914        s1        a1        a9        a7        a8        4        76        -1
DS        1914        s1        a1        a4        a8        144        456        -1
```

```
DA        1914        a1        a5        a8        0        30        -1
A         1914        s1        a1        a10        0
DS        1931        s1        a1        a10        a3        9        31        -1
DS        1931        s1        a1        a6        a3        43        497        -1
DS        1931        s1        a1        a4        a10        384        416        -1
DA        1931        a1        a2        a10        144        756        -1
A         1931        s1        a1        a7        0
DS        1938        s1        a1        a7        a6        14        86        -1
IS        1938        s1        a1        a9        a3        a8        85        415        -1
A         1938        s1        a1        a4        0
DS        1954        s1        a1        a3        a4        38        162        -1
DA        1954        a1        a2        a4        18        582        -1
DS        1954        s1        a1        a6        a7        201        519        -1
DA        1954        a1        a2        a7        220        280        -1
DS        1954        s1        a1        a3        a8        108        492        1
A         1954        s1        a1        a8        1
DA        1959        a1        a5        a6        29        241        -1
DS        1959        s1        a1        a4        a7        182        518        -1
DS        1959        s1        a1        a8        a7        64        256        -1
DS        1959        s1        a1        a4        a10        189        511        -1
DA        1959        a1        a5        a10        29        241        -1
DS        1959        s1        a1        a8        a10        40        200        -1
A         1959        s1        a1        a8        0
DA        1974        a1        a2        a8        95        405        -1
DS        1974        s1        a1        a3        a8        39        61        -1
DS        1974        s1        a1        a6        a8        99        171        -1
A         1974        s1        a1        a4        1
DS        1980        s1        a1        a10        a4        2        28        -1
DS        1980        s1        a1        a8        a4        224        416        -1
DA        1980        a1        a2        a4        170        830        -1
IS        1980        s1        a1        a9        a7        a6        7        73        1
DS        1980        s1        a1        a7        a6        9        61        1
DS        1980        s1        a1        a3        a6        390        610        1
DS        1980        s1        a1        a4        a8        140        560        -1
DA        1980        a1        a2        a8        96        204        -1
IA        1980        a1        a9        a5        a8        58        152        -1
A         1980        s1        a1        a6        0
DS        1993        s1        a1        a4        a3        33        67        1
DA        1993        a1        a2        a3        216        584        1
DA        1993        a1        a5        a6        30        150        -1
DS        1993        s1        a1        a6        a8        210        600        -1
DS        1993        s1        a1        a7        a8        6        74        -1
DS        1993        s1        a1        a6        a10        105        705        -1
DS        1993        s1        a1        a4        a10        72        128        -1
A         1993        s1        a1        a3        0
DA        1999        a1        a2        a3        198        702        -1
DA        1999        a1        a9        a3        96        204        -1
DS        1999        s1        a1        a4        a3        440        560        -1
DS        1999        s1        a1        a3        a8        36        64        -1
A         1999        s1        a1        a6        0
IS        2019        s1        a1        a9        a4        a8        308        392        -1
DS        2019        s1        a1        a6        a8        162        198        -1
```

```
A         2019        s1        a1        a4        0
Actual collusion
a10        a7        PTW
Detectable PTW collusion
a10        a7
Detected PTW collusion
Precision
1.0
Recall
0.0
```

C.2 Collusion Detection Results File Example

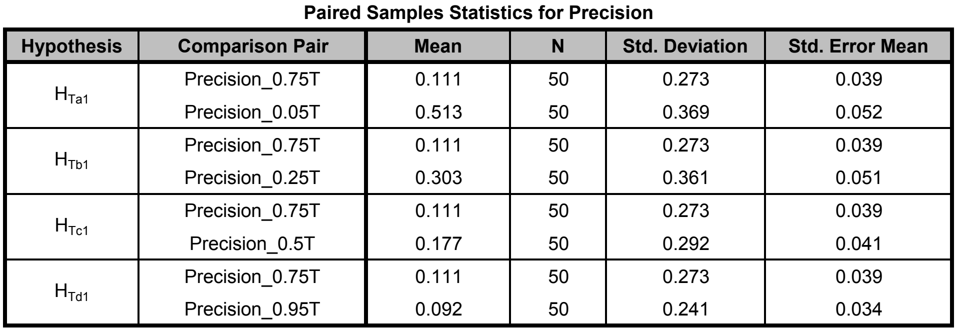
For an agent population of size 10, 132 population configurations have been generated and for which collusion detection has been applied from the point of view of one evaluating agent. The first two lines of the file is the header row.

```
population_size,trust_high,trust_ave,trust_low,service_count,collusive_pairs_count,
aveTruePositive,aveFalsePositive,aveFalseNegative,avePrecision,aveRecall,runCount
10,0,0,100,2,1,0.1,0.1,0.9,0.9,0.1,10
10,0,10,90,2,1,0.3,1.2,0.7,0.5666666666666667,0.3,10
10,0,20,80,2,1,0.0,0.8,1.0,0.4,0.0,10
10,0,30,70,2,1,0.1,3.6,0.9,0.10909090909090909,0.1,10
10,0,40,60,2,1,0.1,3.4,0.9,0.12,0.1,10
10,0,50,50,2,1,0.1,4.8,0.9,0.21000000000000002,0.1,10
10,0,60,40,2,1,0.2,3.4,0.8,0.33111111111111113,0.2,10
10,0,70,30,2,1,0.1,3.9,0.9,0.2111111111111111,0.1,10
10,0,80,20,2,1,0.4,5.0,0.6,0.25928571428571423,0.4,10
10,0,90,10,2,1,0.2,2.6,0.8,0.3,0.2,10
10,0,100,0,2,1,0.0,2.1,1.0,0.6,0.0,10
10,10,0,90,2,1,0.0,2.3,1.0,0.5,0.0,10
10,10,10,80,2,1,0.2,5.5,0.8,0.1325,0.2,10
10,10,20,70,2,1,0.2,3.3,0.8,0.27,0.2,10
10,10,30,60,2,1,0.4,4.0,0.6,0.16588235294117648,0.4,10
10,10,40,50,2,1,0.1,5.8,0.9,0.10909090909090909,0.1,10
10,10,50,40,2,1,0.2,3.6,0.8,0.16999999999999998,0.2,10
10,10,60,30,2,1,0.4,7.2,0.6,0.1403943278943279,0.4,10
10,10,70,20,2,1,0.4,4.9,0.6,0.2796703296703297,0.4,10
10,10,80,10,2,1,0.4,8.6,0.6,0.06337662337662338,0.4,10
10,10,90,0,2,1,0.5,6.0,0.5,0.17678571428571427,0.5,10
10,20,0,80,2,1,0.2,5.6,0.8,0.11526315789473685,0.2,10
10,20,10,70,2,1,0.2,4.3,0.8,0.0375,0.2,10
10,20,20,60,2,1,0.0,7.7,1.0,0.0,0.0,10
10,20,30,50,2,1,0.1,6.9,0.9,0.01,0.1,10
10,20,40,40,2,1,0.3,7.8,0.7,0.12969827586206897,0.3,10
10,20,50,30,2,1,0.1,4.1,0.9,0.011111111111111111,0.1,10
10,20,60,20,2,1,0.3,5.6,0.7,0.16428571428571428,0.3,10
```

```
10,20,70,10,2,1,0.2,5.8,0.8,0.11101190476190477,0.2,10
10,20,80,0,2,1,0.4,7.7,0.6,0.1754312251216276,0.4,10
10,30,0,70,2,1,0.0,9.6,1.0,0.0,0.0,10
10,30,10,60,2,1,0.3,7.6,0.7,0.04236111111111111,0.3,10
10,30,20,50,2,1,0.1,5.5,0.9,0.11428571428571428,0.1,10
10,30,30,40,2,1,0.4,9.0,0.6,0.029066576698155645,0.4,10
10,30,40,30,2,1,0.1,7.1,0.9,0.004545454545454545,0.1,10
10,30,50,20,2,1,0.2,9.7,0.8,0.014814814814814814,0.2,10
10,30,60,10,2,1,0.3,8.3,0.7,0.0313960113960114,0.3,10
10,30,70,0,2,1,0.2,5.2,0.8,0.06999999999999999,0.2,10
10,40,0,60,2,1,0.5,11.7,0.5,0.0710989010989011,0.5,10
10,40,10,50,2,1,0.2,9.3,0.8,0.11114285714285714,0.2,10
10,40,20,40,2,1,0.1,11.6,0.9,0.004545454545454545,0.1,10
10,40,30,30,2,1,0.4,11.6,0.6,0.027122518286311386,0.4,10
10,40,40,20,2,1,0.2,10.0,0.8,0.0625,0.2,10
10,40,50,10,2,1,0.2,9.3,0.8,0.10883458646616542,0.2,10
10,40,60,0,2,1,0.1,5.4,0.9,0.20909090909090908,0.1,10
10,50,0,50,2,1,0.3,14.5,0.7,0.02013888888888889,0.3,10
10,50,10,40,2,1,0.0,5.8,1.0,0.0,0.0,10
10,50,20,30,2,1,0.3,14.0,0.7,0.033333333333333334,0.3,10
10,50,30,20,2,1,0.2,12.0,0.8,0.12954545454545455,0.2,10
10,50,40,10,2,1,0.1,8.8,0.9,0.10714285714285714,0.1,10
10,50,50,0,2,1,0.2,9.6,0.8,0.14166666666666666,0.2,10
10,60,0,40,2,1,0.3,17.7,0.7,0.015049019607843139,0.3,10
10,60,10,30,2,1,0.4,13.5,0.6,0.02626573617952928,0.4,10
10,60,20,20,2,1,0.1,11.0,0.9,0.0125,0.1,10
10,60,30,10,2,1,0.3,7.7,0.7,0.029548872180451123,0.3,10
10,60,40,0,2,1,0.3,11.1,0.7,0.032454212454212455,0.3,10
10,70,0,30,2,1,0.4,14.6,0.6,0.025302840434419382,0.4,10
10,70,10,20,2,1,0.3,11.7,0.7,0.02786037491919845,0.3,10
10,70,20,10,2,1,0.1,12.2,0.9,0.0040,0.1,10
10,70,30,0,2,1,0.2,8.9,0.8,0.12678571428571428,0.2,10
10,80,0,20,2,1,0.3,12.1,0.7,0.11611061352440663,0.3,10
10,80,10,10,2,1,0.3,10.2,0.7,0.03042780748663102,0.3,10
10,80,20,0,2,1,0.3,8.1,0.7,0.029761904761904757,0.3,10
10,90,0,10,2,1,0.2,5.5,0.8,0.36,0.2,10
10,90,10,0,2,1,0.4,10.6,0.6,0.12879901960784312,0.4,10
10,100,0,0,2,1,0.3,9.7,0.7,0.125,0.3,10
10,0,0,100,2,0,0.0,0.1,0.0,0.9,0.0,10
10,0,10,90,2,0,0.0,0.2,0.0,0.8,0.0,10
10,0,20,80,2,0,0.0,3.9,0.0,0.2,0.0,10
10,0,30,70,2,0,0.0,2.1,0.0,0.6,0.0,10
10,0,40,60,2,0,0.0,5.4,0.0,0.1,0.0,10
10,0,50,50,2,0,0.0,3.1,0.0,0.3,0.0,10
10,0,60,40,2,0,0.0,5.1,0.0,0.3,0.0,10
10,0,70,30,2,0,0.0,3.9,0.0,0.2,0.0,10
10,0,80,20,2,0,0.0,5.7,0.0,0.0,0.0,10
10,0,90,10,2,0,0.0,1.3,0.0,0.5,0.0,10
10,0,100,0,2,0,0.0,3.7,0.0,0.1,0.0,10
10,10,0,90,2,0,0.0,2.1,0.0,0.5,0.0,10
10,10,10,80,2,0,0.0,5.6,0.0,0.1,0.0,10
10,10,20,70,2,0,0.0,4.2,0.0,0.2,0.0,10
```

```
10,10,30,60,2,0,0.0,6.1,0.0,0.1,0.0,10
10,10,40,50,2,0,0.0,6.0,0.0,0.1,0.0,10
10,10,50,40,2,0,0.0,5.2,0.0,0.1,0.0,10
10,10,60,30,2,0,0.0,7.6,0.0,0.0,0.0,10
10,10,70,20,2,0,0.0,4.5,0.0,0.1,0.0,10
10,10,80,10,2,0,0.0,2.5,0.0,0.3,0.0,10
10,10,90,0,2,0,0.0,3.6,0.0,0.3,0.0,10
10,20,0,80,2,0,0.0,7.9,0.0,0.0,0.0,10
10,20,10,70,2,0,0.0,5.5,0.0,0.2,0.0,10
10,20,20,60,2,0,0.0,8.2,0.0,0.0,0.0,10
10,20,30,50,2,0,0.0,6.0,0.0,0.1,0.0,10
10,20,40,40,2,0,0.0,8.3,0.0,0.1,0.0,10
10,20,50,30,2,0,0.0,7.5,0.0,0.1,0.0,10
10,20,60,20,2,0,0.0,4.7,0.0,0.0,0.0,10
10,20,70,10,2,0,0.0,4.4,0.0,0.1,0.0,10
10,20,80,0,2,0,0.0,4.7,0.0,0.0,0.0,10
10,30,0,70,2,0,0.0,6.1,0.0,0.1,0.0,10
10,30,10,60,2,0,0.0,6.5,0.0,0.2,0.0,10
10,30,20,50,2,0,0.0,6.9,0.0,0.1,0.0,10
10,30,30,40,2,0,0.0,8.8,0.0,0.1,0.0,10
10,30,40,30,2,0,0.0,6.8,0.0,0.1,0.0,10
10,30,50,20,2,0,0.0,7.2,0.0,0.0,0.0,10
10,30,60,10,2,0,0.0,9.1,0.0,0.0,0.0,10
10,30,70,0,2,0,0.0,8.0,0.0,0.1,0.0,10
10,40,0,60,2,0,0.0,10.3,0.0,0.1,0.0,10
10,40,10,50,2,0,0.0,8.8,0.0,0.0,0.0,10
10,40,20,40,2,0,0.0,12.8,0.0,0.0,0.0,10
10,40,30,30,2,0,0.0,6.0,0.0,0.2,0.0,10
10,40,40,20,2,0,0.0,9.6,0.0,0.0,0.0,10
10,40,50,10,2,0,0.0,7.1,0.0,0.1,0.0,10
10,40,60,0,2,0,0.0,9.8,0.0,0.0,0.0,10
10,50,0,50,2,0,0.0,9.0,0.0,0.1,0.0,10
10,50,10,40,2,0,0.0,11.2,0.0,0.0,0.0,10
10,50,20,30,2,0,0.0,9.3,0.0,0.0,0.0,10
10,50,30,20,2,0,0.0,12.1,0.0,0.0,0.0,10
10,50,40,10,2,0,0.0,7.2,0.0,0.1,0.0,10
10,50,50,0,2,0,0.0,11.2,0.0,0.1,0.0,10
10,60,0,40,2,0,0.0,13.8,0.0,0.0,0.0,10
10,60,10,30,2,0,0.0,9.0,0.0,0.0,0.0,10
10,60,20,20,2,0,0.0,11.2,0.0,0.1,0.0,10
10,60,30,10,2,0,0.0,11.0,0.0,0.1,0.0,10
10,60,40,0,2,0,0.0,10.4,0.0,0.0,0.0,10
10,70,0,30,2,0,0.0,12.8,0.0,0.0,0.0,10
10,70,10,20,2,0,0.0,7.1,0.0,0.1,0.0,10
10,70,20,10,2,0,0.0,10.7,0.0,0.1,0.0,10
10,70,30,0,2,0,0.0,11.6,0.0,0.0,0.0,10
10,80,0,20,2,0,0.0,15.3,0.0,0.0,0.0,10
10,80,10,10,2,0,0.0,13.5,0.0,0.1,0.0,10
10,80,20,0,2,0,0.0,11.4,0.0,0.0,0.0,10
10,90,0,10,2,0,0.0,12.3,0.0,0.0,0.0,10
10,90,10,0,2,0,0.0,12.9,0.0,0.0,0.0,10
10,100,0,0,2,0,0.0,9.4,0.0,0.0,0.0,10
```

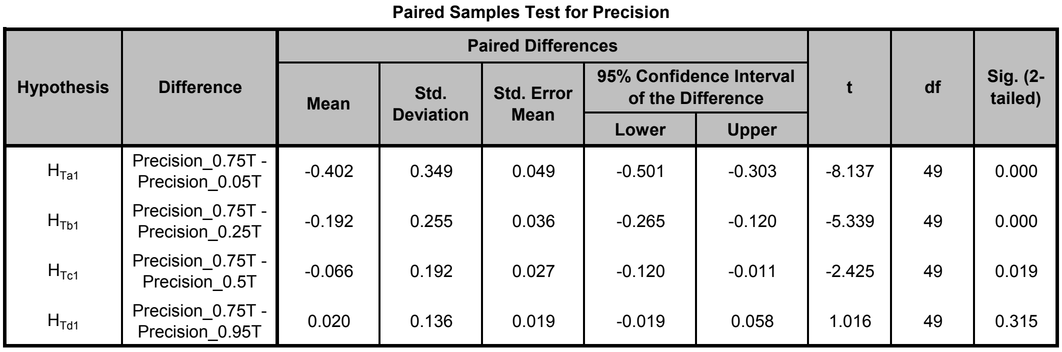
C.3 Paired t-Tests on Cosine Similarity Thresholds for Differences in Precision



Paired Samples Statistics for Precision

Hypothesis	Comparison Pair	Mean	N	Std. Deviation	Std. Error Mean
H_{Ta1}	Precision_0.75T	0.111	50	0.273	0.039
	Precision_0.05T	0.513	50	0.369	0.052
H_{Tb1}	Precision_0.75T	0.111	50	0.273	0.039
	Precision_0.25T	0.303	50	0.361	0.051
H_{Tc1}	Precision_0.75T	0.111	50	0.273	0.039
	Precision_0.5T	0.177	50	0.292	0.041
H_{Td1}	Precision_0.75T	0.111	50	0.273	0.039
	Precision_0.95T	0.092	50	0.241	0.034

Table C.1: Hypotheses Results for Precision (Statistics)

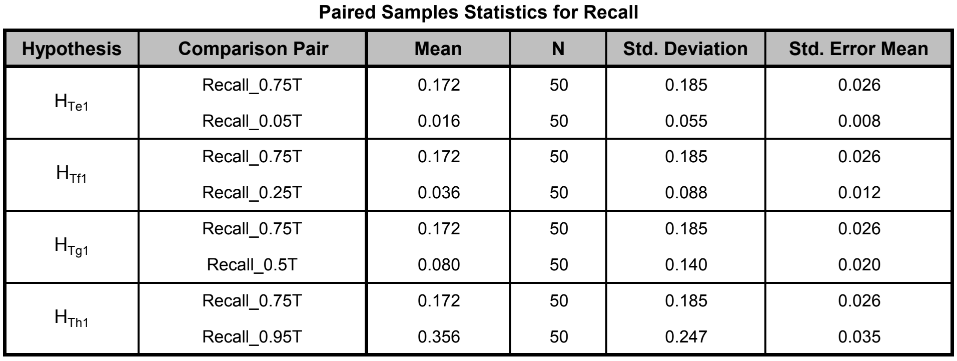


Paired Samples Test for Precision

Hypothesis	Difference	Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
					Lower	Upper			
H_{Ta1}	Precision_0.75T - Precision_0.05T	-0.402	0.349	0.049	-0.501	-0.303	-8.137	49	0.000
H_{Tb1}	Precision_0.75T - Precision_0.25T	-0.192	0.255	0.036	-0.265	-0.120	-5.339	49	0.000
H_{Tc1}	Precision_0.75T - Precision_0.5T	-0.066	0.192	0.027	-0.120	-0.011	-2.425	49	0.019
H_{Td1}	Precision_0.75T - Precision_0.95T	0.020	0.136	0.019	-0.019	0.058	1.016	49	0.315

Table C.2: Hypotheses Results for Precision (t-Test)

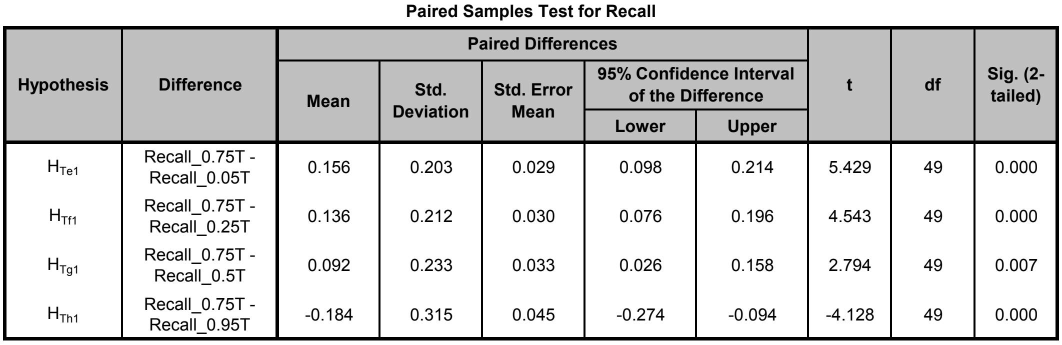
C.4 Paired t-Tests on Cosine Similarity Thresholds for Differences in Recall



Paired Samples Statistics for Recall

Hypothesis	Comparison Pair	Mean	N	Std. Deviation	Std. Error Mean
H_{Te1}	Recall_0.75T	0.172	50	0.185	0.026
	Recall_0.05T	0.016	50	0.055	0.008
H_{Tf1}	Recall_0.75T	0.172	50	0.185	0.026
	Recall_0.25T	0.036	50	0.088	0.012
H_{Tg1}	Recall_0.75T	0.172	50	0.185	0.026
	Recall_0.5T	0.080	50	0.140	0.020
H_{Th1}	Recall_0.75T	0.172	50	0.185	0.026
	Recall_0.95T	0.356	50	0.247	0.035

Table C.3: Hypotheses Results for Recall (Statistics)



Paired Samples Test for Recall

Hypothesis	Difference	Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
					Lower	Upper			
H_{Te1}	Recall_0.75T - Recall_0.05T	0.156	0.203	0.029	0.098	0.214	5.429	49	0.000
H_{Tf1}	Recall_0.75T - Recall_0.25T	0.136	0.212	0.030	0.076	0.196	4.543	49	0.000
H_{Tg1}	Recall_0.75T - Recall_0.5T	0.092	0.233	0.033	0.026	0.158	2.794	49	0.007
H_{Th1}	Recall_0.75T - Recall_0.95T	-0.184	0.315	0.045	-0.274	-0.094	-4.128	49	0.000

Table C.4: Hypotheses Results for Recall (*t*-Test)

Appendix D

Glossary of Notations

The following notations are used throughout this thesis:

Category	Notation	Description
Agent:		An autonomous computational entity.
	a_e	An evaluator assessing agent behaviour.
	a_p	A service provider that can execute particular tasks.
	a_t	A target agent is a potential service provider, being assessed by the evaluator.
	a_r	A recommender (witness) agent which can give recommendations to the requester.
	$a_{r'}$	An intermediate witness.
	$a_{r''}$	With respect to an intermediate witness, it refers to a further witness along the recommendation chain.
	a_{r^d}	A direct recommender.

Category	Notation	Description
Agent:		
	a_{r^i}	An indirect recommender.
	$a_{r^{i'}}$	A further indirect recommender along the recommender chain.
Confidence:		The amount of certainty.
	ST_c	The confidence in the situational trust value.
	RT_c	The confidence in the recommendation trust value.
Count:		A tally.
	$count^+$	The number of positive interactions.
	$count^-$	The number of negative interactions.
	$count^s$	The number of service types.
	$count_{total}$	The total number of interactions.
Dimension:	d	A service characteristic.
	d_{type}	The service characteristic denoted by $type$.
Disposition:		An indication of an agent behaviour's behaviour.
	$disposition_{pass}$	Behaviour indication as a result of a success.
	$disposition_{fail}$	Behaviour indication as a result of a failure.
History:		A record of past events.
	H_{i_s}	A history of past service interactions.
	H_{i_r}	A history of recommendations received.
Interaction:		A transaction event between two agents.
	i_s	An interaction about the execution of a service.

Category	Notation	Description
Interaction:		
	i_r	An interaction relating to a recommendation request.
	i_{r^d}	A direct recommendation interaction.
	i_{r^i}	An indirect recommendation interaction.
Performance value:	PV	The result of the calculation of an agent's trustworthiness from trust and reputation.
Recommendation:	r	The opinion of a witness.
	r^d	A direct recommendation.
	r^i	An indirect recommendation.
Service type:	s	The type of service offered by a provider, or needed by an evaluator.
Time:	t	A point in time.
	t_{curr}	The current time.
Trust:		The assessment of the likelihood that an agent will fulfil its commitments.
	ST	Situational trust - trust in a specific context.
	GT	General trust - the overall trustworthiness of an agent.

Category	Notation	Description
Trust:		
	RT	Recommendation trust - the trustworthiness of a witness in giving opinions.
	$initialT$	Initial trust - the trust value that an agent has in another, even without having interacted with it.
Witness reputation:	WR	Trust information originating from third parties.
	max_{WI}	The maximum number of interactions that the witnesses have used when giving recommendations.
	$total_{WI}$	The total number of interactions actually used in that recommendation.
Weight:	ω	The amount of influence an entity has.
	$\omega_{H_{i_s}}$	Recency weight applied to the interaction history.
	ω_{WR}	The influence of witness reputation.
	ω_{RT_c}	The influence of the confidence in the recommendation trust.
	ω_{WRR}	The weight of of witness reputation relevance.
	μ_f	The weight of the performance evaluation factors.

Appendix E

Abbreviations

MAS	multi-agent systems
MCDA	multiple criteria decision analysis
MDT	multidimensional trust
P2P	peer-to-peer
PD	Prisoners' Dilemma
PTW	Persistent Target-Witness collusion type
TW	Target-Witness collusion
WW	Witness-Witness collusion

References

[1] A. Abdul-Rahman. *A Framework for Decentralised Trust Reasoning*. PhD thesis, Department of Computer Science, University College London, 2005.

[2] A. Abdul-Rahman and S. Hailes. Supporting trust in virtual communities. In *Proceedings of the 33rd Hawaii International Conference on System Sciences (HICSS 2000)*, page 6007. IEEE Computer Society, 2000.

[3] P. Asch and J. J. Seneca. Characteristics of collusive firms. *The Journal of Industrial Economics*, 23(3):223–237, 1975.

[4] R. Axelrod and W. D. Hamilton. The evolution of cooperation. *Science Magazine*, 211(4489), 1981.

[5] R. Baeza-Yates, C. Castillo, and V. López. Pagerank increase under different collusion topologies. In *Workshop on Adversarial Information Retrieval on the Web*, 2007.

[6] R. A. Belecheanu, S. Munroe, M. Luck, T. Payne, T. Miller, P. McBurney, and M. Pěchouček. Commercial applications of agents: Lessons, experiences and challenges. In *Proceedings of the 5th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2006)*, 2006.

[7] F. Brandt. A verifiable, bidder-resolved auction protocol. In *Proceedings of the 5th Workshop on Deception Fraud and Trust In Agent Societies*, pages 18–25, Bologna, Italy, 2002.

[8] S. Braynov and T. Sandholm. Contracting with uncertain level of trust. In *Proceedings of the 1st ACM conference on Electronic commerce*. ACM Press, 1999.

[9] S. Braynov and T. Sandholm. Trust revelation in multiagent interaction. In *Proceedings of CHI 2002 Workshop on The Philosophy and Design of Socially Adept Technologies*, 2002.

[10] J. J. Brown and P. H. Reingen. Social ties and word-of-mouth referral behavior. *Journal of Consumer Behaviour*, 14(3):350–362, 1987.

[11] V. Buskens. Social networks and the effect of reputation on cooperation. ISCORE Paper No. 42, Utrecht University, 1998.

[12] C. Castelfranchi and R. Falcone. Principles of trust for MAS: Cognitive anatomy, social importance, and quantification. In *Proceedings of the International Conference of Multi-Agent Systems (ICMAS 1998)*, pages 72–79, 1998.

[13] C. Castelfranchi and R. Falcone. Socio-cognitive theory of trust. Deliverable report D1, ALFEBIITE, 2001.

[14] P.-A. Chirita, W. Nejdl, and C. Zamfir. Preventing shilling attacks in online recommender systems. In *Proceedings of the 7th Annual ACM International Workshop on Web Information and Data Management (WIDM 2005)*, pages 67–74. Association for Computing Machinery, 2005.

[15] European Commission. Competition: Antitrust overview. Available at `htt://`

ec.europa.eu/competition/antitrust/overview_en.html[accessed February 2011].

[16] F. Dabek, M. F. Kaashoek, D. Karger, R. Morris, and I. Stoica. Wide-area cooperative storage with CFS. In *Proceedings of the 18th ACM Symposium on Operating Systems Principles (SOSP 2001)*, pages 202–215, New York, NY, USA, 2001. ACM.

[17] K. S. Decker. *Environment Centered Analysis and Design of Coordination Mechanisms*. PhD thesis, Department of Computer Science, University of Massachusetts, May 1995.

[18] C. Dellarocas. Immunizing online reputation reporting systems against unfair ratings and discriminatory behavior. In *Proceedings of the 2nd ACM conference on Electronic commerce*. ACM Press, 2000.

[19] J. R. Douceur. The Sybil attack. In *Revised Papers from the 1st International Workshop on Peer-to-Peer Systems (IPTPS 2001)*, pages 251–260, London, UK, 2002. Springer-Verlag.

[20] Y. Du, Y. Shi, and X. Zhao. Using spam farm to boost PageRank. In *Proceedings of the 3rd International Workshop on Adversarial Information Retrieval on the Web (AIRWeb 2007)*, pages 29–36, New York, NY, USA, 2007. ACM.

[21] D. Emerson and S. Piramuthu. Agent-based framework for dynamic supply chain configuration. In *Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS 2004)*, page 70168.1, Washington, DC, USA, 2004. IEEE Computer Society.

[22] B. Esfandiari and S. Chandrasekharan. On how agents make friends: Mechanisms for trust acquisition. In *Proceedings of the 4^{th} Workshop on Deception, Fraud and Trust in Agent Societies*, 2001.

[23] W. N. Evans and I. N. Kessides. Living by the "Golden Rule": Multimarket Contact in the U.S. Airline Industry. *The Quarterly Journal of Economics*, 109(2):341–366, 1994.

[24] R. Falcone and C. Castelfranchi. The socio-cognitive dynamics of trust: Does trust create trust? In *Proceedings of the workshop on Deception, Fraud, and Trust in Agent Societies held during the Autonomous Agents Conference: Trust in Cyber-societies, Integrating the Human and Artificial Perspectives*. Springer-Verlag, 2000.

[25] R. Falcone and C. Castelfranchi. Social trust: A cognitive approach. In Cristiano Castelfranchi and Yao-Hua Tan, editors, *Trust and Deception in Virtual Societies*, pages 55–90. Kluwer Academic Publishers, Netherlands, 2001.

[26] R. Falcone, G. Pezzulo, and C. Castelfranchi. A fuzzy approach to a belief-based trust computation. In R. Falcone, S. Barber, L. Korba, and M. Singh, editors, *Trust, Reputation and Security: Theories and Practice*, volume 2631 of *Lecture Notes in Artificial Intelligence*. Springer-Verlag, 2003.

[27] M. Fasli. On agent technology for e-commerce: trust, security and legal issues. *The Knowledge Engineering Review*, 22(1):3–35, 2007.

[28] G. W. Flake, S. Lawrence, C. L. Giles, and F. M. Coetzee. Self-organization and identification of web communities. *IEEE Computer*, 35(3):66–71, 2002.

[29] S. Franklin and A. Graesser. Is it an agent, or just a program?: A taxonomy for autonomous agents. In J.P. Müller, M.J. Wooldridge, and N.R. Jennings, editors, *Intelligent Agents III – Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages, Lecture Notes in Artificial intelligence*, volume 1193, pages 21–35. Springer-Verlag, 1997.

[30] E. Friedman and P. Resnick. The social cost of cheap pseudonyms. *Journal of Economics and Management Strategy*, 10(2):173–199, 2001.

[31] D. Gambetta. Can we trust trust? In *Trust: Making and Breaking of Cooperative Relations*, pages 213–237. Department of Sociology, University of Oxford, 2000.

[32] D. Gambetta, editor. *Trust: Making and Breaking of Cooperative Relations*. Department of Sociology, University of Oxford, 2000.

[33] K. C. Gowda and G. Krishna. Agglomerative clustering using the concept of mutual nearest neighbourhood. *Pattern Recognition*, 10(2):105–112, 1978.

[34] E. Gray, J.-M. Seigneur, Y. Chen, and C. Jensen. Trust propagation in small worlds. In *Proceedings of the 1st International Conference on Trust Management (iTrust 2003)*. Springer-Verlag Berlin, 2003.

[35] N. Griffiths. Task delegation using experience-based multi-dimensional trust. In *Proceedings of the 4th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2005)*, pages 489–496, New York, NY, USA, 2005. ACM Press.

[36] N. Griffiths. Enhancing peer-to-peer collaboration using trust. *International Journal of Expert systems with Applications*, 31(4):849–858, 2006.

[37] N. Griffiths. A fuzzy approach to reasoning with trust, distrust and insufficient trust. In *Cooperative Information Agents X*, volume 4149 of *Lecture Notes in Computer Science*, pages 360–374. Springer-Verlag, 2006.

[38] N. Griffiths, K.-M. Chao, and M. Younas. Fuzzy trust for peer-to-peer systems. In *Proceedings of P2P Data and Knowledge Sharing Workshop (P2P/DAKS 2006), at the 26th International Conference on Distributed Computing Systems (ICDCS 2006)*, 2006.

[39] N. Griffiths and M. Luck. Cooperative plan selection through trust. In *Proceedings of Multi-Agent System Engineering: Proceedings of the 9th European Workshop on Modelling Autonomous Agents in a Multi-Agent World*, volume 1647 of *Lecture Notes in Artificial Intelligence*, pages 162–174, 1999.

[40] S. P. Hargreaves-Heap and Y. Varoufakis. *Game Theory: A Critical Introduction*. Routledge, London, UK, 1997.

[41] J. Harrington. Detecting cartels. In P. Buccirossi, editor, *Handbook in Antitrust Economics*. MIT Press, 2008.

[42] E. Hartuv and R. Shamir. A clustering algorithm based on graph connectivity. *Information Processing Letters*, 76(4–6):175–181, 2000.

[43] K.E. Heinrich. *Similarity Measures*, chapter 3, pages 27–38. World Scientific Publishing, Singapore, 2006.

[44] David Hull. Using statistical testing in the evaluation of retrieval experiments. In *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 329–338, New York, USA, 1993. ACM.

[45] T. D. Huynh. *Trust and Reputation in Open Multi-Agent Systems*. PhD thesis, Electronics and Computer Science, University of Southampton, June 2006.

[46] T. D. Huynh, N. R. Jennings, and N. Shadbolt. Developing an integrated trust and reputation model for open multi-agent systems. In *Proceedings of the 7^{th} International Workshop on Trust in Agent Societies*, pages 65–74, New York, USA, 2004.

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

[48] D. Isern, D. Sánchez, and A. Moreno. Agents applied in health care: A review. *International Journal of Medical Informatics*, 79(3):145–166, 2010.

[49] M. Ivaldi, B. Jullien, P. Rey, P. Seabright, and J. Tirole. The economics of tacit collusion. IDEI Working Papers 186, Institut d'Économie Industrielle (IDEI), Toulouse, March 2003.

[50] R. A. Jarvis and E. A. Patrick. Clustering using a similarity measure based on shared near neighbors. *IEEE Transactions on Computers*, C-22(11):1025–1034, 1973.

[51] A. Jøsang, R. Ismail, and C. Boyd. A survey of trust and reputation systems for online service provision. *Decision Support Systems*, 43(2):618–644, 2007.

[52] R. Jurca. *Truthful Reputation Mechanisms for Online Systems*. Phd thesis, 3955, Ecole Polytechnique Fédérale de Lausanne, 2007.

[53] R. Jurca and B. Faltings. Collusion-resistant, incentive-compatible feedback pay-

ments. In *Proceedings of the 8th ACM conference on Electronic commerce (EC 2007)*, pages 200–209, New York, NY, USA, 2007. ACM.

[54] 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 2003)*, pages 640–651, New York, NY, USA, 2003. ACM.

[55] H. Kautz, B. Selman, and A. Milewski. Agent amplified communication. In *Proceedings of the 13th National Conference on Artificial Intelligence (AAAI 1996)*, 1996.

[56] H. Kautz, B. Selman, and M. Shah. Referral Web: Combining social networks and collaborative filtering. *Communications of the ACM*, 40(3), 1997.

[57] J. Kleinberg and S. Lawrence. The structure of the Web. *Science*, 294(5548):1849 – 1850, 2001.

[58] B. Kosko. Fuzzy cognitive maps. *International Journal Man-Machine Studies*, 24, 1986.

[59] S. Kuhn. Prisoner's dilemma. The Stanford Encyclopedia of Philosophy (Fall 2003 Edition), Edward N. Zalta, editor, available at `http://plato.stanford.edu/archives/fall2003/entries/prisoner-dilemma/`, 2003.

[60] Z. Kunda. The case for motivated reasoning. *Psychological Bulletin*, 108(3):480–498, 1990.

[61] S. K. Lam, D. Frankowski, and J. Riedl. Do you trust your recommendations? An exploration of security and privacy issues in recommender systems. In *Proceedings*

of the International Conference on Emerging Trends in Information and Communication Security (ETRICS 2006), volume 3995 of *Lecture Notes in Computer Science*, pages 14–29. Springer-Verlag, 2006.

[62] S. K. Lam and J. Riedl. Shilling recommender systems for fun and profit. In *Proceedings of the Proceedings of the 13th International Conference on World Wide Web (WWW 2004)*, pages 393–402. Association for Computing Machinery, 2004.

[63] P. Leitẽo. Agent-based distributed manufacturing control: A state-of-the-art survey. *Engineering Applications of Artificial Intelligence*, 22(7):979–991, 2009. Distributed Control of Production Systems.

[64] J. Leskovec and E. Horvitz. Planetary-scale views on a large instant-messaging network. In *Proceedings of the 17th International World Wide Web Conference (WWW 2008)*. ACM, 2008.

[65] L. Li. *Supply Chain Management: Concepts, Techniques and Practices Enhancing the Value Through Collaboration*. World Scientific, 2007.

[66] Q. Lian, Z. Zhang, M. Yang, B. Y. Zhao, Y. Dai, and X. Li. An empirical study of collusion behavior in the Maze P2P file-sharing system. In *Proceedings of the 27th International Conference on Distributed Computing Systems (ICDCS 2007)*, page 56. IEEE Computer Society, 2007.

[67] Y. Liao and V. R. Vemuri. Use of K-Nearest Neighbor classifier for intrusion detection. *Computers & Security*, 21(5):439–448, 2002.

[68] D. Liben-Nowell and J. Kleinberg. The link prediction problem for social networks.

In *Proceedings of the 12th International Conference on Information and Knowledge Management*, pages 556–559, 2003.

[69] S. N. Lim Choi Keung and N. Griffiths. Indirect recommendations for improved trust assessment. In *Proceedings of 11th International Workshop on Trust in Agent Societies*, pages 33–40. IFAAMAS, 2008.

[70] S. N. Lim Choi Keung and N. Griffiths. Towards improved partner selection using recommendations and trust. In R. Falcone et al., editors, *Trust in Agent Societies (TRUST 2008)*, volume 5396 of *Lecture Notes in Computer Science*, pages 43–64. Springer-Verlag Berlin Heidelberg, 2008.

[71] S. N. Lim Choi Keung and N. Griffiths. Using recency and relevance to assess trust and reputation. In *Proceedings of AISB 2008 Symposium on Behaviour Regulation in Multi-Agent Systems*, volume 4, pages 13–18. The Society for the Study of Artificial Intelligence and Simulation of Behaviour, 2008.

[72] S. N. Lim Choi Keung and N. Griffiths. Building a trust-based social agent network. In R. Falcone, S. Barber, J. Sabater-Mir, and M. Singh, editors, *Proceedings of the 12th International Workshop on Trust in Agent Societies*, pages 68–79, 2009.

[73] S. N. Lim Choi Keung and N. Griffiths. Networking for multi-agents: Beyond a local view. In K. Decker, J. Sichman, C. Sierra, and C. Castelfranchi, editors, *Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2009)*, pages 1315–1316. IFAAMAS, 2009.

[74] S. N. Lim Choi Keung and N. Griffiths. Trust and reputation. In N. Griffiths and K.-M. Chao, editors, *Agent-Based Service-Oriented Computing*, Advanced Information and Knowledge Processing, pages 189–224. Springer London, 2010.

[75] A. R. Lomuscio, M. Wooldridge, and N. R. Jennings. A classification scheme for negotiation in electronic commerce. In F. Dignum and C. Sierra, editors, *Agent-Mediated Electronic Commerce: A European AgentLink Perspective*, pages 19–33. Springer-Verlag, 2001.

[76] M. Luck and M. d'Inverno. A formal framework for agency and autonomy. In *Proceedings of the 1^{st} International Conference on Multi-Agent Systems*, pages 254–260. AAAI Press / MIT Press, 1995.

[77] M. Luck, P. McBurney, and C. Preist. A manifesto for agent technology: Towards next generation computing. *Journal of Autonomous Agents and Multi-Agent Systems*, 9(3):203–252, 2004.

[78] M. Luck, P. McBurney, O. Shehory, and S. Wilmott. *Agent Technology: Computing as Interaction (A Roadmap for Agent Based Computing)*. AgentLink, 2005.

[79] N. Luhmann. *Trust and power*. Wiley, Chichester, 1979.

[80] N. Luhmann. Familiarity, confidence, trust: Problems and alternatives. In D. Gambetta, editor, *Trust*, pages 94–107. Blackwell, 1990.

[81] S. Marsh. *Formalising Trust as a Computational Concept*. PhD thesis, Department of Computer Science, University of Stirling, 1994.

[82] S. Marsh. Optimism and pessimism in trust. In H. Geffner, editor, *Proceedings of IV Ibero-American Conference on Artificial Intelligence (IBERAMIA 1994)*, pages 286–297. Addison-Wesley, 1994.

[83] N. Mezzetti. A socially inspired reputation model. In *Proceedings of the 1^{st} European PKI Workshop (EuroPKI 2004)*. Springer-Verlag, 2004.

[84] S. Milgram. The small world problem. *Psychology Today*, 2:60–67, 1967.

[85] M. Montaner, B. López, and J. LLuís de la Rosa. A taxonomy of recommender agents on the internet. *Artificial Intelligence Review*, 19(4):285–330, 2003.

[86] D. C. Montgomery and G. C. Runger. *Applied Statistics and Probability for Engineers*. John Wiley & Sons, New York, USA, 2003.

[87] A. Moreno and J. L. Nealon, editors. *Applications of software agent technology in the health care domain*. Birkhäuser Verlag, 2000.

[88] M. Motta. *Competition Policy: Theory and Practice*. Cambridge University Press, 2004.

[89] L. Mui, M. Mohtashemi, and A. Halberstadt. A computational model of trust and reputation for e-businesses. In *Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS 2002)*, volume 7, page 188, Washington, DC, USA, 2002. IEEE Computer Society.

[90] G. Muller and L. Vercouter. Decentralized monitoring of agent communications with a reputation model. In *Trusting Agents for Trusting Electronic Societies*, volume 3577 of *Lecture Notes in Artificial Intelligence*, pages 144–161. Springer Berlin Heidelberg, 2005.

[91] G. Muller and L. Vercouter. L.I.A.R.: Achieving social control in open and decentralised multi-agent systems. Technical Report 2008-700-001, École Nationale Supérieure des Mines de Saint-Étienne, 2008.

[92] United States Department of Justice. Price fixing, bid rigging, and market allocation schemes: What they are and what to look for. Electronic document, available

at http://www.usdoj.gov/atr/public/guidelines/211578.htm, accessed July 2009.

[93] David L. Olson and Dursun Delen. *Performance Evaluation for Predictive Modeling*, pages 137–147. Springer Berlin Heidelberg, 2008.

[94] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank citation ranking: Bringing order to the web. Sidl-wp-1999-0120, Stanford University, 1999.

[95] G. K. Palshikar and M. M. Apte. Collusion set detection using graph clustering. *Data Mining and Knowledge Discovery*, 16(2):135–164, 2008.

[96] M. Paolucci and R. Sacile. *Agent-based manufacturing and control systems: New agile manufacturing solutions for achieving peak performance*. CRC Press, 2005.

[97] Cambridge University Press. Cambridge Dictionaries Online. Available online at http://dictionary.cambridge.org.

[98] P. Raghavan. Social networks: From the Web to the enterprise. *IEEE Internet Computing*, 6(1):91–94, 2002.

[99] S. D. Ramchurn, D. Huynh, and N. R. Jennings. Trust in multi-agent systems. *The Knowledge Engineering Review*, 19(1):1–25, 2004.

[100] S. D. Ramchurn and N. Jennings. Trust in agent-based software. In *Trust and Crime in Information Societies*, pages 165–204. Elgar Publishing, 2005.

[101] R. Rees. Tacit collusion. *Oxford Review of Economic Policy*, 9(2):27–40, 1993.

[102] M. Rehák and M. Pěchouček. Trust modeling with context representation and generalized identities. In *Proceedings of the 11th International Workshop on Coop-*

erative Information Agents (CIA 2007), volume 4676 of *Lecture Notes in Computer Science*, pages 298–312. Springer, 2007.

[103] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, New Jersey, USA, second edition, 2003.

[104] J. Sabater. *Trust and Reputation in Agent Societies*. PhD thesis, Universitat Aùtonoma de Barcelona, 2003.

[105] J. Sabater. Evaluating the ReGreT system. *Applied Artificial Intelligence*, 18, 2004.

[106] J. Sabater and C. Sierra. A reputation model for gregarious societies. In *Proceedings of the 4th Workshop on Deception Fraud and Trust in Agent Societies*, pages 61–70, Montreal, Canada, 2001.

[107] J. Sabater and C. Sierra. REGRET: reputation in gregarious societies. In *Proceedings of the 5th International Conference on Autonomous Agents*, pages 194–195, Montreal, Canada, 2002. ACM Press.

[108] J. Sabater and C. Sierra. Review on computational trust and reputation models. *Artificial Intelligence Review*, 24:33–60, 2005.

[109] S. E. Schaeffer. Graph clustering. *Computer Science Review*, 1(1):27–64, 2007.

[110] M. Schillo. *Trust and Deceit in Multi-agent Systems*. PhD thesis, Department of Computer Science, Saarland University, Germany, 1999.

[111] M. Schillo, P. Funk, and M. Rovatsos. Using trust for detecting deceitful agents in artificial societies. *Applied Artificial Intelligence, Special Issue on Trust, Deception, and Fraud in Agent Societies*, 14(8):825–848, 2000.

[112] J. Scott. *Social Network Analysis: a handbook*. Sage Publications, London, UK, 2001.

[113] S. Sen. Reciprocity: A foundational principle for promoting cooperative behavior among self-interested agents. In Victor Lesser, editor, *Proceedings of the 1stt International Conference on Multiagent Systems*, pages 322–329. MIT Press, 1996.

[114] S. Sen and P. S. Dutta. The evolution and stability of cooperative traits. In C. Caltelfranchi and L. Johnson, editors, *Proceedings of the 1st International Joint Conference on Autonomous Agents and Multiagent Systems*, volume 3, pages 1114–1120. ACM Press, 2002.

[115] G. Shafer. The Dempster-Shafer theory. In S. C. Shapiro, editor, *Encyclopedia of Artificial Intelligence*, pages 330–331. Wiley, 1992.

[116] A. Sharma, A. K. Pujari, and K. K. Paliwal. Intrusion detection using text processing techniques with a kernel based similarity measure. *Computers & Security*, 26(7–8):488–495, 2007.

[117] J. Smed, T. Knuutila, and H. Hakonen. Can we prevent collusion in multiplayer online games? In *Proceedings of the 9th Scandinavian Conference on Artificial Intelligence (SCAI 2006)*, 2006.

[118] J. Smed, T. Knuutila, and H. Hakonen. Towards swift and accurate collusion detection. In *Proceedings of the 8th International Conference on Intelligent Games and Simulation (Game-On 2007)*, pages 103–107, 2007.

[119] M. Srivatsa and L. Liu. Securing decentralized reputation management using TrustGuard. *Journal of Parallel and Distributed Computing*, 66(9):1217–1232, 2006.

[120] E. Staab and T. Engel. Formalizing excusableness of failures in multi-agent systems. In *Proceedings of the 10th Pacific Rim International Conference on Multi-Agents (PRIMA 2007)*. Springer, 2007.

[121] E. Staab and T. Engel. Collusion detection for grid computing. In *Proceedings of the 9th IEEE/ACM International Symposium on Cluster Computing and the Grid*, pages 412–419. IEEE Computer Society, 2009.

[122] P. Stone and M. Veloso. Multiagent systems: A survey from a machine learning perspective. *Autonomous Robots*, 8(3):345–383, 2000.

[123] G. Swamynathan, B. Y. Zhao, and K. C. Almeroth. Exploring the feasibility of proactive reputations. *Concurrency and Computation: Practice and Experience*, 20:155–166, 2008.

[124] P.-N. Tan, M. Steinbach, and V. Kumar. *Introduction to Data Mining*, pages 65–84. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2005.

[125] W. T. L. Teacy, J. Patel, N. 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.

[126] J. Travers and S. Milgram. An experimental study of the small world problem. *Sociometry*, 32(4):425–443, 1969.

[127] I. Čače and J. J. Bryson. Agent based modelling of communication costs: Why information can be free. In C. Lyon, C. L. Nehaniv, and A. Cangelosi, editors, *Emergence and Evolution of Linguistic Communication*, pages 305–322. Springer, London, 2007.

[128] J. von Newmann and O. Morgenstern. *Theory of Games and Economic Behaviour*. Princeton University Press, 60th anniversary edition edition, 2004.

[129] R. Vragov. Implicit consumer collusion in auctions on the internet. In *Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS 2005) – Track 7*, page 174.3, Washington, DC, USA, 2005. IEEE Computer Society.

[130] F. E. Walter, S. Battiston, and F. Schweitzer. A model of a trust-based recommendation system on a social network. *Journal of Autonomous Agents and Multi-Agent Systems*, 16(1):57–74, 2008.

[131] J.-C. Wang and C.-C. Chiu. Detecting online auction inflated-reputation behaviors using social network analysis. In *Annual Conference of the North American Association for Computational Social and Organizational Science (NAACSOS 2005)*, 2005.

[132] J.-C. Wang and C.-C. Chiu. Recommending trusted online auction sellers using social network analysis. *Expert Systems with Applications*, 34(3):1666–1679, 2008.

[133] B. Whaley. Detecting deception: A bibliography of counterdeception across time, cultures, and disciplines. Technical report, Office of the Director of National Intelligence, March 2006.

[134] M. Witkowski, A. Artikis, and J. Pitt. Experiments in building experiential trust in a society of objective-trust based agents. In R. Falcone, M. Singh, and Y. H. Tan, editors, *Trust in Cyber Societies*, volume 2246 of *Lecture Notes in Artificial Intelligence*, pages 111–132. Springer-Verlag, 2001.

[135] M. Witkowski and J. Pitt. Objective trust-based agents: Trust and trustworthiness in a multi-agent trading society. In *Proceedings of the 4th International Conference on MultiAgent Systems (ICMAS 2000)*, 2000.

[136] M. Wooldridge and N. R. Jennings. Intelligent agents: Theory and practice. *The Knowledge Engineering Review*, 10(2):115–152, 1995.

[137] D. J. Wu and Y. Sun. The emergence of trust in multi-agent bidding: A computational approach. In *Proceedings of the 34th Annual Hawaii International Conference on System Sciences (HICSS 2001)*, volume 1, page 1041, Washington DC, USA, 2001. IEEE Computer Society.

[138] Nong Ye, editor. *The Handbook of Data Mining*. Lawrence Erlbaum Associates, New Jersey, USA, 2003.

[139] B. Yu and M. P. Singh. An evidential model of reputation management. In C. Caltelfranchi and L. Johnson, editors, *Proceedings of the 1st International Joint Conference on Autonomous Agents and Multi-Agent Systems*, volume 1, pages 295–300. ACM Press, 2002.

[140] B. Yu and M. P. Singh. Detecting deception in reputation management. In *Proceedings of the 2nd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2003)*, pages 73–80. ACM, 2003.

[141] B. Yu and M. P. Singh. Searching social networks. In *Proceedings of the 2nd International Joint Conference on Autonomous Agents and MultiAgent Systems (AAMAS 2003)*, pages 65–72. ACM Press, 2003.

[142] G. Zacharia and P. Maes. Trust management through reputation mechanisms. *Applied Artificial Intelligence*, 14(9):881–907, 2000.

[143] G. Zacharia, A. Moukas, and P. Maes. Collaborative reputation mechanisms in electronic marketplaces. In *Proceedings of the 32nd Annual Hawaii International Conference on System Sciences*, volume 8, page 8026. IEEE Computer Society, 1999.

[144] L. Zadeh. Fuzzy sets. *Information and Control*, 8, 1965.