University of Fort Hare
Together in Excellence

A Scalable Trust Management Model for the Internet of Things (IoT) Environment

A Thesis Submitted in Fulfilment of The
Requirements for the Degree
Of

DOCTOR OF PHILOSOPHY
IN
COMPUTER SCIENCE

By

Caroline Gurajena

Supervisor
Prof Khulumani Sibanda

Co-Supervisor
Prof Mamello Thinyane

July 2018

Abstract

The Internet of Things (IoT) environment involves the interaction of numerous ‘things’. These ‘things’ are embedded with different kinds of technologies such as RFID technology, NFC technology and sensors. This causes IoT to bring into play many security risks apart from the ones that already exist in the current Internet, for example, embedded RFID tags can be easily triggered and send their content which could be private information. IoT also introduces internal attacks such as on-off attack, bad mouthing and white washing attacks into networks that already exist. These internal attacks cannot be solved by hard security mechanisms such as cryptographic algorithms and firewalls because they guarantee total trust. This eliminates uncertainty which should always be available where trust exist. That is, hard security mechanisms enable IoT ‘things’ to either trust another ‘thing’ completely or not and this makes them unsuitable for the IoT environment. When objects in any network are communicating, there is some element of uncertainty. Also, hard security mechanisms such as public key cryptography cause communication overheard in the already resource-constrained IoT devices and these conventional cryptography methods cannot deal with internal attacks. This brings about the need for a middleware that includes functions that will manage trust, privacy and security issues of all data exchange, communications and network connections.

For IoT to be successful, the provision of trust, security and privacy measures are essential. Trust management may enhance the adoption and security measures in IoT. Trust helps in identifying trustworthy ‘things’ in the network and give ‘things’ in the network the ability to reason in all aspects concerning trust in the environment. Trust can be administered through a trust management model.

This research notes that most of the trust models that have been proposed fail to address scalability challenges and lack suitable computation methods. It is on that premise that this research focuses on developing a suitable trust model for the IoT environment. The research also introduces new ways of creating relationships in IoT. This enables the creation of new cooperation opportunities in the environment. In overall, this research aimed to design and develop a generic trust and authority delegation model for the heterogonous IoT environment that is scalable and generalized to cater for the heterogeneous IoT environment. This research was conducted in three phases. The first phase reviewed literature in order to identify outstanding issues in IoT trust management and also identify the suitable computational method. This provided a critical analysis of different computational methods highlighting their advantages and limitations.

In the second phase of the research, the proposed trust model was designed and tested. In the last phase, the feasibility of the proposed model was evaluated. The proposed model is based on fuzzy logic. Fuzzy logic was selected for trust computation because it is able to imitate the process of the human mind through the use of linguistic variables and it can handle uncertainty. The proposed model was tested in a simulated environment. The simulation results showed that the proposed model can identify selfish and malicious entities effectively. The results also showed that the model was able to deal with different types of behaviours of entities. The testing proved that the proposed trust model can support decision making in IoT based on trust. The results from the evaluation show that this research ameliorates the design and development of trust management solutions for the IoT environment.

DECLARATION

I, Caroline Gurajena, do hereby declare that the work titled "A Scalable Trust Management Model for the Internet of Things (IoT) Environment" is my own work and that, to the best of my knowledge, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any degree or diploma of the University or any other institution of higher learning except where due acknowledgement has been made in the text.

I hereby also declare that I am fully aware of the University of Fort Hare's policy on plagiarism and research ethics, and I have taken every necessary measure to comply with the regulations.

Signature.. 17/092018 …………………………..

Caroline Gurajena Date

ACKNOWLEDGEMENTS

I would like to take this opportunity to show my appreciation:

- Thank you so much, God, for the strength, wisdom, and guidance you provided for me throughout this journey.
- To my parents Mr and Mrs Hahn, I am forever grateful for all that you do for me. Nyari and Letwin thank you so much for your support. Without forgetting the rest of my family, thank you so much guys, you are my support system.
- To my friends Edmore and Kudzanai, Thank you so much for all your support. I am sorry for all the time you had to sit and listen to me rumble, but I think it was all worth it in the end. Eddie thank you so much for the constructive criticism and proofreading. To Gamuchirai, Doiline, Vongai and Itai, thank you so much for your support and prayers. All of you guys are the best, I couldn't have asked for better friends.
- To my supervisors, Prof M. Thinyane and Prof K. Sibanda, thank so much for your invaluable encouragement, support and guidance. I would not have done this without you.
- To the Computer Science department staff at UFH, that you so much for your support and contribution. May God bless you.
- A special thank you also goes to Mr M.S. Scott and Prof B. Makamba; I greatly appreciate your assistance. A special thank you also goes to Prof T. Mushoriwa and family, thank you for your hospitality and support.

DEDICATION

To the blessed memory of:

my grandmother Monica Nyashanu who taught me that I could be anything if I make the right decisions,

my sister Constance Mapira who believed in me all the way

and my uncle Thulani Mushonga who showed unconditional love to everyone he encountered.

May your Memories be eternal.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

ATMS – Agent-based Trust Management System

CoG – Centre of Gravity

FIS – Fuzzy Inference System

IoT – Internet of Things

NFC – Near Field Communication

QoS – Quality of Service

RFID – Radio Frequency Identification

SIoT – Social Internet of Things

TMS – Trust Management System

TTP – Trusted Third Party

WSN – Wireless Sensor Network

1

Introduction

1.1 Introduction

Advances in the network coverage and different communication technologies have led to many opportunities for network use. One of these opportunities is the Internet of Things. The term Internet of Things (IoT) was coined in 1999 by Ashton as an idea for Radio Frequency Identification (RFID) (Ashton, 1999). However, over the years the idea of IoT has grown to include other technologies and the concept has become even more complex. This research adopted the definition of IoT by (Vermesan et al., 2011) who defined IoT as: "*a dynamic global network infrastructure with self-configuring capabilities based on standard and interoperable communication protocols where physical and virtual "things" have identities, physical attributes, and virtual personalities, use intelligent interfaces, and are seamlessly integrated into the information network*". This definition highlights the importance of the information network in IoT. The information network enables the virtual 'things' to interact with each other.

The term IoT is "*composed of two terms: internet and things*" (Mäkinen, 2015). The term 'Internet' focuses on the "*network-oriented vision of IoT*" and the term 'things' focuses on the generic objects that form the IoT network (Atzori, Iera, & Morabito, 2010). These generic objects are heterogeneous. Because of this, IoT is also highly affected by interoperability which leads to a third term (interoperability). Figure 1-1 shows the three main visions that different researchers have been focusing on over the past few years which come together in IoT. Figure 1-1 shows the importance of interoperability in IoT. Interoperability caters for heterogeneous 'things' in the IoT network by adding disparate 'things' to connect and communicate seamlessly.

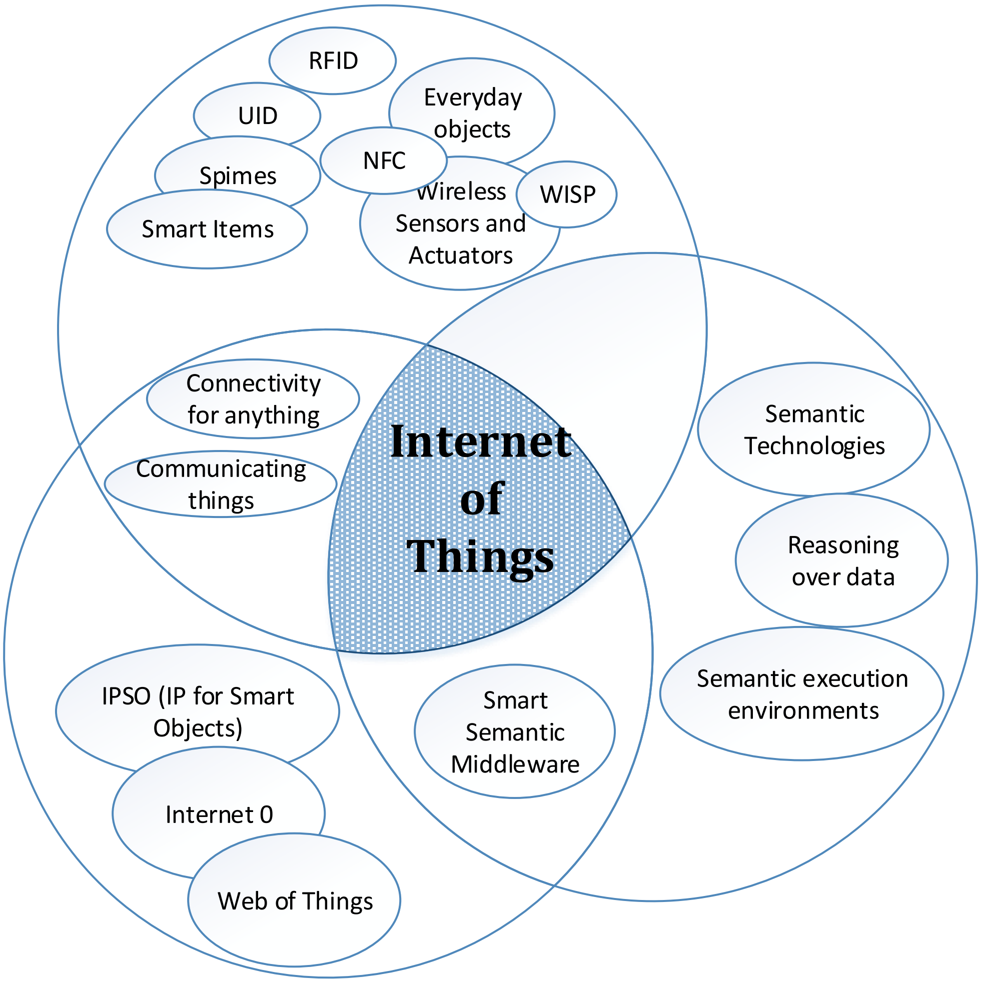


Figure 1-1: IoT paradigm due to the convergence of different visions (Atzori et al., 2010)

The IoT brings about a world where all objects and devices are uniquely identified and connected to the internet. Technological developments have been underway in the last few years for such a network. Many frameworks have already been proposed for the IoT. The building blocks of the IoT include Wireless Sensor Networks (WSNs), RFID systems and heterogeneous connectivity to the Internet (Oteafy & Hassanein, 2012).

The advantages of technologies like RFID tags in the IoT environment are that they are small and they do not depend on battery power. Their disadvantage is that they can only be used for object and location identification due to their limited resources. RFID personify some of the 'things' found in IoT which have limited resources. The IoT vision requires context-aware services. This makes sensors and other devices with more functionality to be more favourable in other instances of the network. There are also other technologies that can also be used besides these two such as Near Field Communication (NFC) and Bluetooth.

The vision of IoT includes minimal human intervention on the cooperation and communication of the 'things'. This means that the IoT objects should support self-* functions which include:

- Self-healing
- Self-configuration
- Self-adaptation
- Self-organization

These self-* functions causes the IoT to come into play with its own complex security requirements besides those already found on the Internet. This is mainly because of the added heterogeneity, autonomy and resource capability inconsistencies of the 'things' in the network (Saied, Olivereau, Zeghlache, & Laurent, 2013).

Apart from these 'things' being heterogeneous and autonomous, they are also mobile. Unlike the Internet, the IoT network is not constant, 'things' join and leave the network constantly. Each 'thing' in the IoT should be able to securely connect and collaborate with other 'things' on the network. This machine to machine communication in IoT brings into play new forms of attacks as well. While a lot of approaches have been designed which include end-to-end encryption, token-based access control and others, it was recognised that a trust management model is very necessary. This research proposes a solution to some of the IoT security and privacy issues such as internal attacks through a trust management model.

1.2 IoT Security and Privacy Issues

Security is one of the outstanding issues in IoT (Mc Kelvey, Kevin, & Subaginy, 2014). Security and privacy requirements play a crucial role in the IoT because of the amount of data that will be available. *Figure 1-2* shows issues that are still outstanding in the IoT environment.

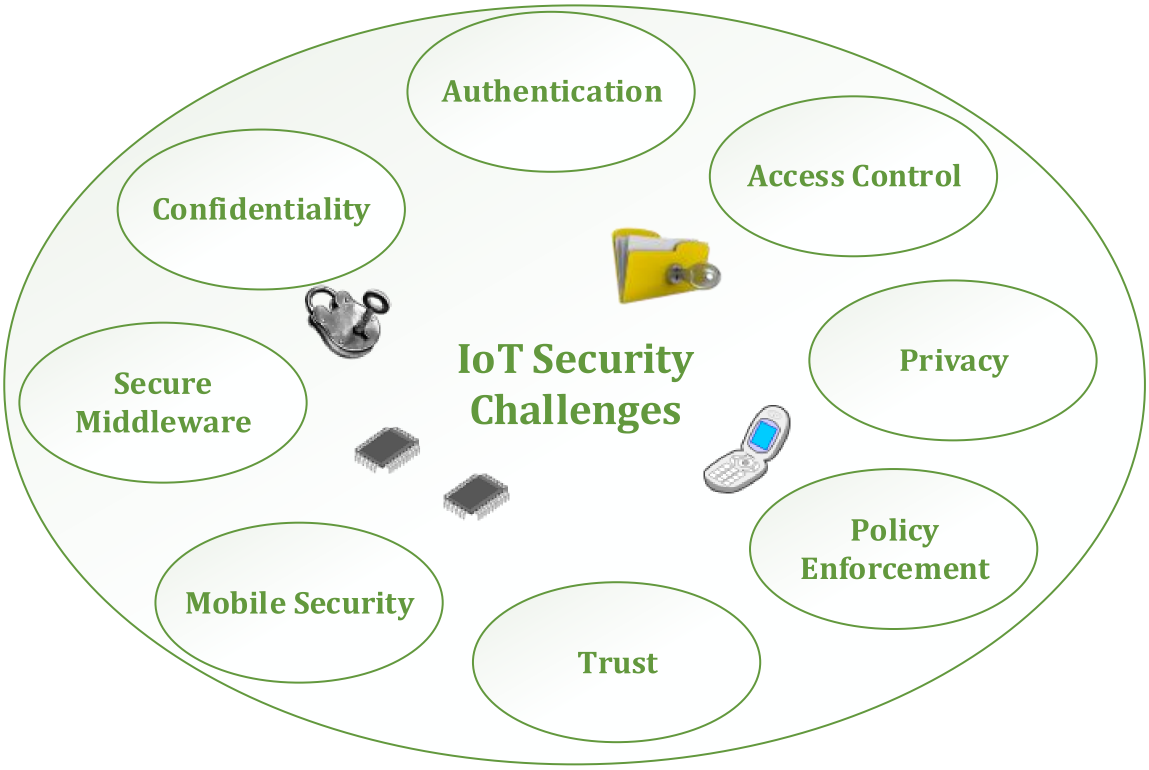


Figure 1-2: IoT Security Issues (Sicari, Rizzardi, Grieco, & Coen-Porisini, 2015)

The security and privacy requirements for the IoT environment include:

- Authentication
- Confidentiality
- Integrity
- Access control
- Privacy
- Trust

"*Traditional security countermeasures cannot be directly applied to IoT technologies due to the different standards and communication stacks involved*" (Sicari et al., 2015). The IoT requires different kinds of security measures in the form of security and trust management to prevent malfunctions and attacks. The security and trust management solutions should support scalability. Scalability issues have risen due to the number of connected devices in the IoT; hence there is a need for a flexible infrastructure that will be capable of handling scalability. According to Sicari et al., (2015), if not properly handled the IoT systems have the potential to magnify current security threats due to its high level of heterogeneity and scalability.

1.3 Research Background and Motivation

The last few years have seen a great increase in the use of the IoT. We are moving into an era where everyone and every device that is embedded with any form of network technology will be connected to the internet. This brings into play a massive network of heterogeneous 'things' that are capable of communicating and collaborating with each other. The IoT joins together our physical world with the digital world. The vision of IoT is to support billions of 'things'. All these 'things' in the IoT need to be intelligent and autonomous by supporting self-* functions and they also need to be able to communicate with each other. Each of these 'things' has a unique identity and is expected to be able to establish secure communication with other 'things' in the IoT environment. Figure 1-3 shows estimates of IoT 'things' made by International Data Corporation (IDC).

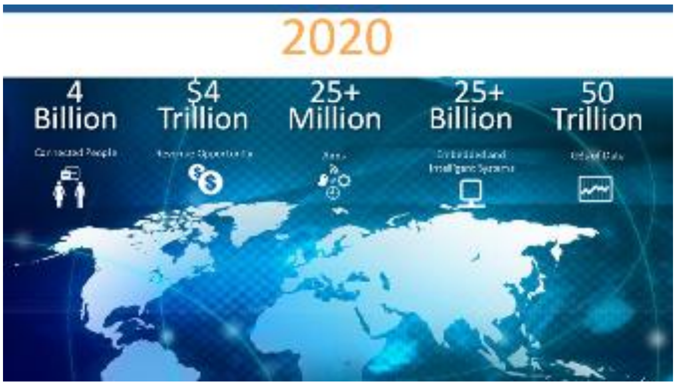


Figure 1-3: IoT 2020 Network Estimation

With this enormous internet connection comes also many challenges which include:

- Addressing device heterogeneity
- Interoperability issues with different communication technologies
- Cooperation, coordination and scalability beyond current systems
- Security and privacy issues

The IoT is prone to security attacks because the smart objects in the IoT environment are open and have limited resources (Alam, Chowdhury, & Noll, 2011). The amount of personal information that will be available in the IoT environment requires security in the form of protocols, frameworks and architectures not as an add-on feature. Certificate authorities and authorization servers can be used to validate the identity of a 'thing' in IoT but they cannot guarantee the trustworthiness of the 'thing' in the IoT environment.

Therefore, there is a need to put into place features which provide integrity, availability, confidentiality, and services (Bassi & Horn, 2008). The trust requirement for the IoT environment can be provided through trust management models. These trust management models should "*be able to define trust in a dynamic and collaborative environment*" (Roman, Najera, & Lopez, 2011). Trust can enable safe communication and collaboration among 'things'.

Users in the IoT environment need to be able to control their services and have tools that enable them to control all their interactions. This will help them to have a mental map of all their virtual surroundings. To enable this, there is a need to design and develop a common interoperable security framework that includes trust and have continuity capabilities with support for different policies (Pescatore, 2014). Trust management can be provided as one of the functionalities of such a framework. The framework needs to be created in such a way that it is not excessive in its monitoring and controlling of the network.

Most of the IoT 'things' have limited resources such as memory and power, and hence are not capable of providing security on their own, they require a trust model. Also, due to the heterogeneity of the IoT environment, a trust model goes a long way towards solving some of the IoT security issues. The model can identify malicious, selfish and compromised entities in the network (Umarani & Sundaram, 2013). Traditional methods such as lightweight cryptography, secure protocols, and privacy assurance are no longer enough for IoT (Pescatore, 2014). Hence there is a need to develop an IoT security model that support privacy by design for the IoT environment.

1.4 Problem Description

Trust management is imperative in mitigating internal attacks. The past few decades have seen the proposal and design of a number of trust management for the IoT. The first problem with these models is that most of them have been designed for specific scenarios such as WSN. The second is that there are few general trust model suitable for different kinds of application and devices in the IoT (Umarani & Sundaram, 2013). The third problem is that the computation methods used do not take into consideration uncertainty in computing trust. There is always an element of uncertainty in trust, hence an alternative approach is required for trust management that minimizes communication overheads in the already resource-constrained IoT devices. The aim of this research is to design and develop a generic trust and authority delegation model for the heterogonous IoT environment that will all the problems mentioned above.

1.5 Research Questions and Objectives

The aim of this research is to design and develop a trust and authority delegation model that manages trust and cooperation issues in the IoT environment while taking into consideration the resources available. The model is designed to be:

- Scalable
- Adaptive
- Fault-tolerant

Table 1-1 shows the summarized research questions and objectives.

Table 1-1: Research Question and Objective

Research Question	**Objective**	**Chapter**
What are the main trust properties suitable for IoT?	Identifying trust properties suitable for IoT	Chapter 3
How can trust be best managed in an IoT environment?	Identifying limitations of current models	Chapter 4
What are suitable trust computation methods for IoT?	Identifying computation methods suitable for IoT	Chapter 5
What are the trust relationships available in the IoT environment? How can new trust relationships be created in IoT?	Creating new trust relationships that are suitable for the IoT environment	Chapter 6
How can trust be managed for IoT things with limited resources in terms of power and computational capability?	Designing of an architecture that is suitable for the IoT environment	Chapter 6
What is the suitable propagation method for IoT?	Identifying trust propagation method suitable for IoT	Chapter 6
What is the value of distrust in the IoT environment?	Investigation the importance of distrust in IoT	Chapter 7

Does including distrust improve the trust model?		

The research also seeks to explore the concept of trust with the aim of identifying its main characteristics that need to be taken into consideration in the IoT environment.

1.6 Importance of Research

The current security methods, access control methods and trust methods cannot manage the general concept of trustworthiness in the IoT environment. There is a need for a generic scalable trust model suitable for heterogeneous IoT 'things'. This research seeks to develop such a model. The research takes into consideration the heterogeneity of the network in designing and developing the trust model. Resources and computational capabilities were also taken into consideration. This research aims to alleviate inside attacks which are common in the IoT environment. The trust model will equip IoT 'things' with the ability to establish trust and detect behaviour anomalies. The proposed model also seeks to ameliorate the design and development of trust management solutions for the IoT environment. The research seeks also to introduce new ways of creating relationships among IoT 'things'. This will enable the creation of new cooperation opportunities in the environment.

1.7 Research Outline

The rest of this research thesis is organised as follows:

- **Chapter 2**: describes in detail the research design and methodology. Research design elaborates on the research philosophy and paradigms considered in this research. The methodology explains the steps taken in carrying out the research.
- **Chapter 3**: discusses the concept of trust. It also outlines the properties of trust that needs to be taken into consideration when developing a trust model for IoT. The chapter concludes by discussing the relationship between trust and distrust.
- **Chapter 4**: discusses the current state of trust modelling in IoT. It highlights that work that has already been done and the outstanding issues.

- **Chapter 5**: highlights common computational methods that have been used in other trust models for computing trust. The chapter concludes by giving reasons for the choice of the computational method selected for this research.
- **Chapter 6**: gives details of the proposed model. The details include the architecture and components of the proposed model.
- **Chapter 7**: discusses the testing and evaluation of the proposed model. The chapter includes the evaluation of the results that were obtained.
- **Chapter 8**: concludes the thesis by discussing the contribution of the research and highlighting the future work.

1.8 Conclusion

For IoT to generate value, it is important for 'things' to interact and these interactions are done in uncertain circumstances due to the openness of the network. This challenge can be tackled by putting into place mechanisms that can enable the IoT devices to deal with this uncertainty. This can be done through trust management. This research focuses on trust management in the IoT environment. It proposes a trust management that will enable different entities in the environment to safely communicate and collaborate. The research proposes a trust evaluation model that will enable all the devices on the IoT environment to communicate and cooperate regardless of the amount of time that the device has been on the IoT network. The main aim of this research is to develop a generic, scalable and reconfigurable model suitable for the IoT environment. The research seeks to provide an accurate trust assessment method for IoT. This is achieved by proposing relationship types and a suitable computation method.

2

Research Design and Methodology

2.1 Introduction

Research is the systematic investigation of a problem using facts, theories and data analysis with the aim to solve a problem. The main aim of the research is to contribute to the board of knowledge through discovering hidden facts that have not yet been discovered. Research consists of "*defining and redefining problems, formulating hypotheses or suggested solutions; collecting, organising and evaluating data; making deductions and reaching conclusions; and at last, carefully testing the conclusions to determine whether they fit the formulating hypothesis*" (Woody, 1927).

With all this in mind, this research is done under computer science. Computer science research is difficult to carry out mainly because there are no set rules on how it should be done (Tedre, 2007). Apart from lack of clear rules for computer science; the perception, beliefs, and assumptions of the researcher(s) also affect how the research is conducted.

Research in computer science can be regarded as an invention. The main focus of computer science is to drive the computer to become more efficient, reliable, secure and faster. This also causes research in computer science to be difficult because the value of the research output depends mainly on the Computer Science community. However, in overall research in computer science is similar to research in all other branches of science Hassani (2017) depending on the view of the researcher(s).

The main aim of this research is to design an IoT trust management model. In order to provide a systematic way to design the IoT trust management model, this chapter outlines the research design and methodology. Trust is a social aspect which is complex. Since trust is a complex human element, it is important to describe the research philosophy that was considered. The human element of trust adds complexity through "free will". The aim of the research is also to bring a better understanding to the fundamental concepts of trust management and trust computation. The researcher's understanding of the relationship between knowledge and its development process and practical consideration determines the philosophy chosen by the researcher. The following sections describe the phases and methods that were used in this research.

The objective of this chapter is to give a brief description of the procedure taken to design the proposed trust model. This chapter is divided into major two sections: research design and research methodology. The research design expounds on the research assumptions and research philosophy stance taken by the researchers. The research methodology explains the methods and processes used in the research.

2.2 Research design

Research design outlines the procedure that will be taken in solving the identified problem. It is the conceptual structure that will be followed in carrying out the research. According to (Carroll & Swatman, 2000), *"[A]ll researchers interpret the world through some sort of conceptual lens formed by their beliefs, previous experiences, existing knowledge, assumptions about the world and theories about knowledge and how it is accrued. The researcher's conceptual lens acts as a filter: the importance placed on the huge range of observations made in the field (choosing to record or note some observations and not others, for example) is partly determined by this filter*" (Carroll & Swatman, 2000). Research consists of formulating a problem, designing a methodology for solving the problem and finally solving the problem. A research that lacks a methodology is incomplete. The research design describes the research methods followed to achieve the objectives of the research.

This research is an applied research. The research sought to provide a security solution for the IoT. The research design consists of a detailed description of the research methods, the concepts, assumptions, and beliefs of the researchers. Before discussing the research paradigm adopted in this research, it is imperative to first discuss the philosophy of computer science. Therefore, the following section highlights some of the major points that have been raised regarding the philosophy of computer science.

2.2.1 The Philosophy of Computer Science

Philosophy of computer science "*reflect on the concepts, aims, structure, and methodologies of computer science and its various fields*" (Brey & Søraker, 2009). (Brey & Søraker, 2009) argue that the titles "*philosophy of computing*" and "*philosophy of computer science*" limits the research area of computer science because the former implies that the focus is on computing systems while the latter implies that the focus is on the field rather than what the field produces. Since computer science has its foundation in other fields such as logic, mathematics, and biology, its conceptual questions are analogous to these fields. Some of the research done in computer science makes it seem as a branch of mathematics that seeks to design and develop algorithms that can be developed into computer systems using a programming language.

Since computer science utilizes different fields and combines theory (Dodig-crnkovic, 2002), design and practice, its research needs to have both a theoretical basis and an experimental design. Figure 2-1 shows relationships among different disciplines that affect computer science. The core of computer science is theoretical computer science (Brey & Søraker, 2009).

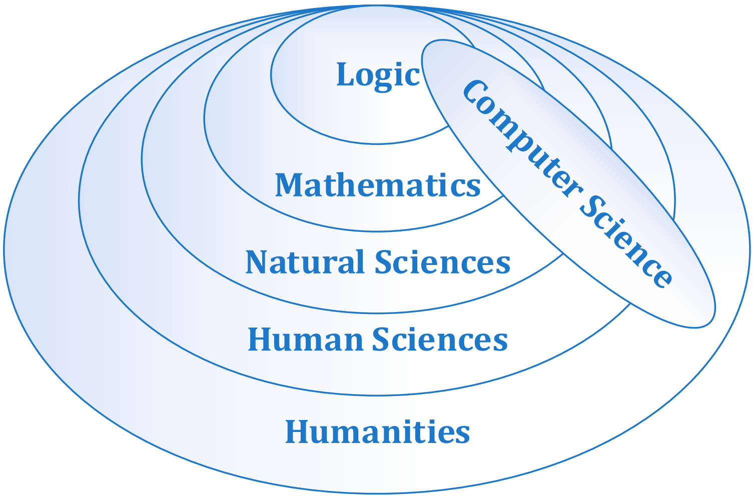


Figure 2-1: Computer Science Discipline

Unlike other science disciplines which are governed by natural laws, computer science is governed by man-made laws which can be proved. Apart from methods involved in system design and testing, the philosophy of computer science also focuses on ontological, methodological and computational issues (Turner & Angius, 2017).

According to (Turner & Angius, 2017) philosophy of computer science asks the following questions:

- "*What kinds of methods do computer scientists use to investigate computing*?"
- "*What is the subject matter of computer science*?"

(Tedre, 2007) highlighted that there was a lack of textbooks on the philosophy of computer science; this is still the case even now. A computer scientist needs to have knowledge of the different methods and approaches he/she uses in his/ her research. (Brey & Søraker, 2009) identified the following as the activities which should be considered for the philosophy of computer science:

- "*Analysis, interpretation, and clarification of central concepts in computer science and the relation between them".*
- *"Analysis, clarification, and evaluation of aims and key assumptions of computer science and its various subfields and the relations between them".*
- *"Analysis, clarification, and evaluation of the methods and methodologies of computer science and its various subfields".*
- *"Analysis of the scientific status of computer science and its relation to other academic fields".*
- *"Analysis of the role and meaning of computer science for society as a whole, as well as for particular human aims and enterprises".*

In his argument that computer science was not a science discipline but an engineering discipline, (Brooks, 1996) made the point that "*the scientist builds in order to study; the engineer studies in order to build*". Indeed, this is true for most of the research that is being done in computer science over the past few decades. Computer scientists are concerned with designing algorithms and software systems.

However, someone might argue that engineering disciplines build tangible objects. The abstract, mathematical and scientific part of computer science is undeniable, however, these are not the only parts of computer science. Social studies are not considered as a branch of computer science, however, some of their concepts are considered in computer science research such as trust. This affects research such as this one which includes social aspects. The following section highlights research paradigms with the aim of highlighting computer science research paradigms.

2.2.2 Research Paradigms

A research paradigm is "*a basic set of beliefs, views, values, and assumptions that guide action and include the researcher's epistemological, ontological and methodological premises*". Research Paradigm is made up of the ontology, epistemology, methodology, and methods. Both the research design and methodology are affected by the paradigm. The paradigm affects how knowledge is studied and interpreted by researchers. The paradigm helps in describing the motivation and expectations of the researcher. The research paradigm explains the assumptions and concepts of the researchers.

"*Doctoral research must have a sound theoretical or philosophical foundation and it must make an original contribution to a theoretical aspect of knowledge"* (Odejobi, 2012). Philosophy is the fundamental knowledge of research. According to (M Saunders, Lewis, & Thornhill, 2009), research philosophy "*relate to the development of knowledge and the nature of knowledge*". It describes the beliefs of the researchers on the research being carried out. Philosophy was also defined by (Rapaport, 2005) as "*the search for truth in any field by rational means (which might be limited to deductive logic, or might be extended to include empirical scientific investigation*)".

The research philosophy adopted in any research is greatly affected by epistemology and doxology. Philosophy focuses on reality, knowledge, and existence. The research philosophy adopted by the researcher determines the assumptions of how the researcher views the world and the assumptions will, in turn, determine the research methods and strategy (M Saunders et al., 2009). The methodology used in a research is affected by the research strategy.

The research onion proposed by (M Saunders et al., 2009) is useful in the design of the research as well as the methodology (Hassani, 2017). It enables the researcher to explain factors that affect the research such as social and psychological factors.

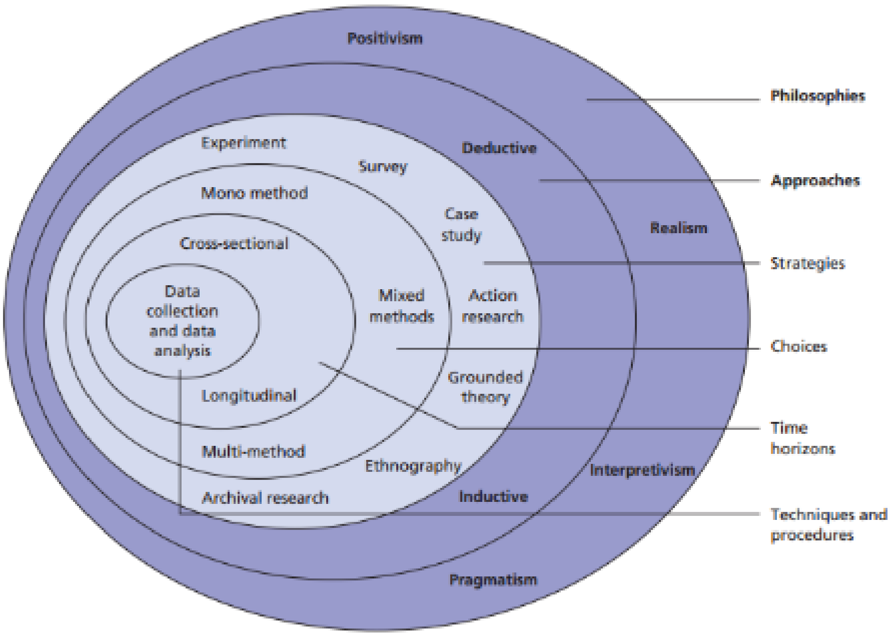


Figure 2-2: Research Onion (Mark Saunders, Lewis, & Thornhill, 2008)

Figure 2-2 shows four types of philosophies for research. Realism states that knowledge about objects can be obtained from the researchers' common sense. The positivism paradigm takes the view that "*knowledge is absolute and objective*" (Mora, Gelman, Steenkamp, & Raisinghani, 2012). According to the positivist paradigm, a "*single objective reality exists external to human beings*" (Mora et al., 2012). The positivist paradigm is similar to the scientific method in that knowledge is obtained through experiments. The positivism research is mainly done using quantitative methods and data is analysed using statistical methods.

Interpretivism paradigm "*aims to find new interpretations or underlying meanings and permits the accommodation of multiple correct approaches and findings, mediated by time, context and researcher*" (Mora et al., 2012). The findings of the Interpretivism research are subjective and depend on the researcher. Interpretivism paradigm is appropriate for research that is focused on human behaviour and social phenomena. Research in this paradigm is done using qualitative methods. Pragmatists "*believe that reality is constantly renegotiated, debated, interpreted, and therefore the best method to use is the one that solves the problem*". These are the common paradigms found across most disciplines. The following section discusses research paradigms that have been suggested for computer science and concludes by discussing the paradigm in which this research falls into.

2.2.3 Computer Science Research Paradigm

The researcher's perception of reality affects the research being done and determines the paradigm of the research. This can be viewed as an ontological issue and not as an epistemological issue. "*An ontology is a theory of what exists and how it exists, and an epistemology is a related theory of how we can come to know those things*". The nature of knowledge (epistemology) and the nature of reality (ontology) determines the framework of a research. Some researchers argue that the world can only be viewed as either objective (limited number of truth) and while others argue that it is subjective (various interpretations).

As mentioned earlier, computer science has different branches. These branches of computer science, have different ontological, epistemological and methodological views. This leads us to closely examine research paradigms that apply to computer science. Paradigms help in explaining other views that are not scientific views which were taken into consideration by the researcher. In 2012, (Denicolo & Becker, 2012) identified positivism and realism as research paradigms that can be applied to computer science (Denicolo & Becker, 2012). Some researchers refer to positivism as the 'scientific method'. "*For positivists, the goal of research is describing what we experience through observation and measurement in order to predict and control the forces that surround us*" (O'Leary, 2004). Generally, positivism refers to the collection of quantitative data using scientific methods. In quantitative research, information is obtained from analysis of numerical data. Quantitative methods are usually used in positivists and post-positivists research.

The following paradigms have been proposed for computer science by (Eden, 2007):

- **The rationalist paradigm** – commonly used by theoretical computer scientists. In this paradigm, computer science is regarded as a branch of mathematics. In this regard, computers are regarded as mathematical machines and programs are regarded as mathematical activities. This makes deductive reasoning of importance when researching programs.

- **The technocratic paradigm** – software engineers view computer science as a branch of engineering discipline. This paradigm focuses on branches of software engineering and argues that computer science lacks both theory and science. "*In* *line with the empiricist position in traditional philosophy, the technocratic paradigm holds that reliable, a posteriori knowledge about programs emanates only from experience, whereas certain, a priori 'knowledge' emanating from the deductive methods of theoretical computer science is either impractical or impossible in principle*" (Eden, 2007). The technocratic paradigm focuses on the providing tool that can be used in large programs to control complexity and provide reliability. The controlling of complexity may hinder the functionality of the program.

- **The scientific paradigm** – in this paradigm, computer science is viewed as a natural science. This view is usually taken by researchers in artificial intelligence. "*Since many programs are unpredictable, or even 'chaotic', the scientific paradigm holds that a priori knowledge emanating from deductive reasoning must be supplanted with a posteriori knowledge emanating from the empirical evidence by conducting scientific experiments. Since program-processes are temporal, non-physical, causal, metabolic, contingent upon a physical manifestation, and nonlinear entities, the scientific paradigm holds them to be on a par with mental processes*" (Eden, 2007).

Research in computer science is governed by either mathematical methods or methods of natural science (Eden, 2007) depending on what is being investigated. (Eden, 2007) went on to argue that regarding computer science as a branch of natural science implies that deductive and analytical methods of investigation must be included in its methods. In this regard, deductive reasoning plays similar roles in both computer science and in other branches of natural science. The debate about the discipline to which computer science belong has been going on for decades and this research is not going to add into it. In this research, the researchers are of the view that computer science falls into multiple disciplines depending on the research being done.

The view of this research is that computer science is a discipline of both mathematics and experimental science, and the research also acknowledge that it borrows from other fields as well. Taking into consideration the above-mentioned paradigms, this research falls under the scientific paradigm. This research to investigated different aspects that affect trust with the aim of proposing a model suitable for IoT.

The prior knowledge obtained from deductive reasoning about trust and computation methods will be used to build the model. The model will be tested in a simulated environment. The experiments done on the model will enable us to gain knowledge about how trust can be established and maintained in an IoT environment. The tests done on the model will be focused on identifying factors that affect trust, new ways of defining trust relationships and the best computational equations that can be applied in an IoT environment. These tests will enable us to either validate or invalidate the proposed model.

2.3 Research Methodology

The paradigm in which the research is based determines the methodological strategies that the researcher uses. According to (Hassani, 2017; Kothari, 2004), research methods are the approaches, procedures, techniques and guidelines utilized in carrying out research while research methodology is the systematic "*scientific approach that investigates, compares, contrasts and explains the different ways that a research could be conducted alongside different methods that could be used in these processes*" (Hassani, 2017). The method also defines the "*limits of computation and the computational paradigm*".

In order to make a scientific inquiry, the research methodology must be based on a theoretical basis and explains a scientific procedure. Methodology summarizes a formal scientific procedure taken in solving a problem. According to (Avison & Fitzgerald, 2006), a methodology is based on the philosophical perspective of the world. The methodology also includes the discussion of the assumptions that were made by the researcher.

Formulating a proper methodology is difficult in computer science because computer science is interdisciplinary (Hassani, 2017). The methodology used in the research is affected by whether the research is experimental or theoretical (Hassani, 2017). Methodologies in computer science can be categorized as modelling, experimental, simulation and theoretical. Figure 2-3 shows the relationships among methodologies that can be used in computer science.

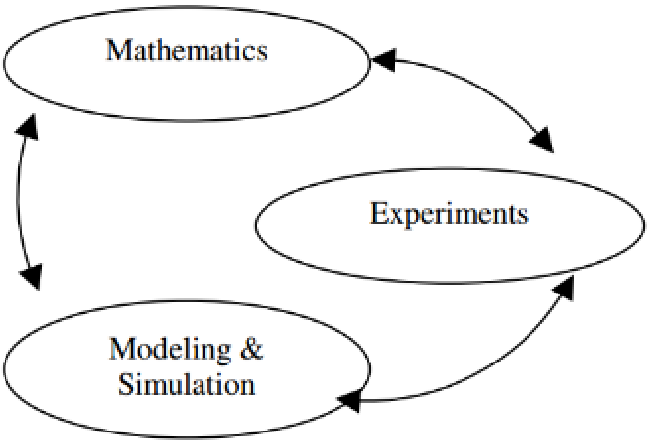


Figure 2-3: Relationship between experiment and simulation (Longman, 2003)

Modelling, however, is a combination of the other three methodologies. (Dodig-Crnkovic, 2002) defined modelling as "*a process that always occurs in science, in a sense that the phenomenon of interest must be simplified, in order to be studied*". The modelling method defines an abstract model which is less complex than the real system. The model can be used in carrying out experiments called simulations.

(Computer Science and Telecommunications Board National Research Council, 1994) define experimental computer science as "*the building of, or the experimentation with or on, nontrivial hardware or software systems*". Experimental science is composed of observation, hypothesis testing, and reproducibility (Feitelson, 2006). The experimental methodology is used in research that includes designing and developing a system. According to P. J. Denning (1981), the experimental method enables researchers to create experiments where they can extract results from real-world implementations. The experiments can be used to test either systems or theories. The results from this method should also be reproducible. The experimental method can be divided into two phases: exploratory and evaluation.

The simulation method is used for systems that are outside the experimental method scope. It is mostly used to carry out research on new inventions. Simulations are experiments that are carried out on an abstract model. Theoretical research conforms to traditions of mathematics and logic and the theoretical methodology follows the methodologies for building theories. The theoretical method is mostly used in the design and analysis of algorithms.

Before the description of the methodology used in this research, it is imperative to consider the scientific method. This consideration raised the following question: Does computer scientists require a scientific method? The following section addresses this question.

2.3.1 Scientific Method

Science is composed of models, measurements, predictions, and validations using the scientific method (Cerf, 2012). The scientific method improves the quality of research by guaranteeing a degree of objectivity. According to Francis Bacon, a scientific method is a process of forming hypotheses and verifying them through experimentation. The hypotheses can be transformed into models if the experiments are successful. Figure 2-4 shows a summarized version of the scientific method.

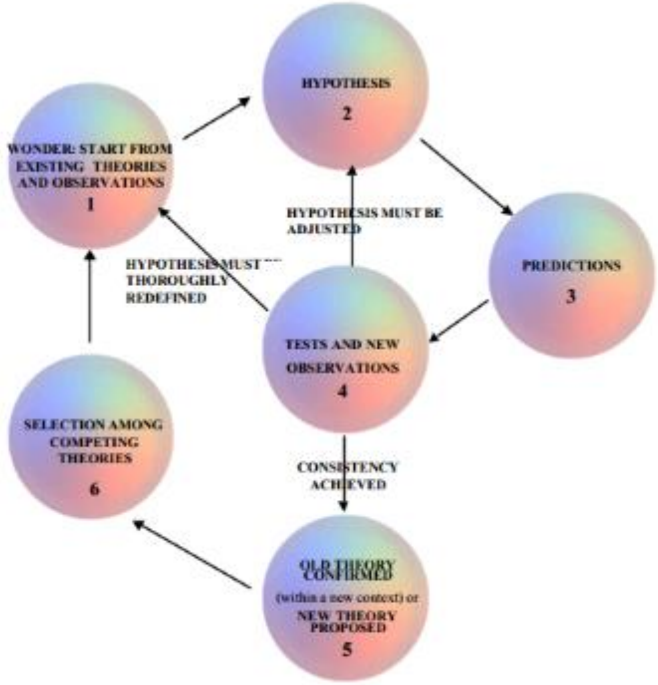


Figure 2-4: The Scientific Method (Dodig-Crnkovic, 2002)

Considering computer science as a science field makes the scientific method applicable to computer science. However, research in computer science is differentiated from other sciences by the artefact that is being studied (Dodig-Crnkovic, 2002). The scientific method is widely used in software engineering (Eden, 2007). The scientific method highlights three main activities:

- Observation
- Formulation
- Experimentation

This method is important in experimental and simulation research. The scientific method enables researches to be repeated and this enables results to be validated. Following a scientific method minimizes the bias of the researcher(s) carrying out the research. Computational models have roots in mathematics and are difficult to combine with social elements. This makes this research more difficult because it combines a computational method with a human element (trust). This makes research such as this one difficult to clearly categorize because it overlaps different fields. A report written by Peter J Denning et al., (1989) suggested computer science research falls in to any one of the following paradigms: theory, abstraction and design. This research falls loosely into two of these paradigms. (Eden, 2007) concluded that programs are at par with mental processes since this research models trust which is a human aspect, the researcher is of the same opinion.

Considering the above discussion and taking into consideration the computer science methodologies the conclusion that the scientific method is significant in computer science research. This conclusion is also supported directly by the experimental methodology which requires the results of the research to be reproducible. Therefore, the scientific method was considered in this research. In respect to computer science methodologies, this research is both an experimental and a simulation research. It is experimental because the research sought to ameliorate the design and development of trust management solutions. It falls under simulation because the model will be tested in a simulated environment. The following section gives details of the steps that were taken is conducting the research.

2.3.2 Research Process

This research was carried out in three phases as shown in Figure 2-5. Each of the phases is made of two activities. The activities for phase 1 and phase 2 were recursive.

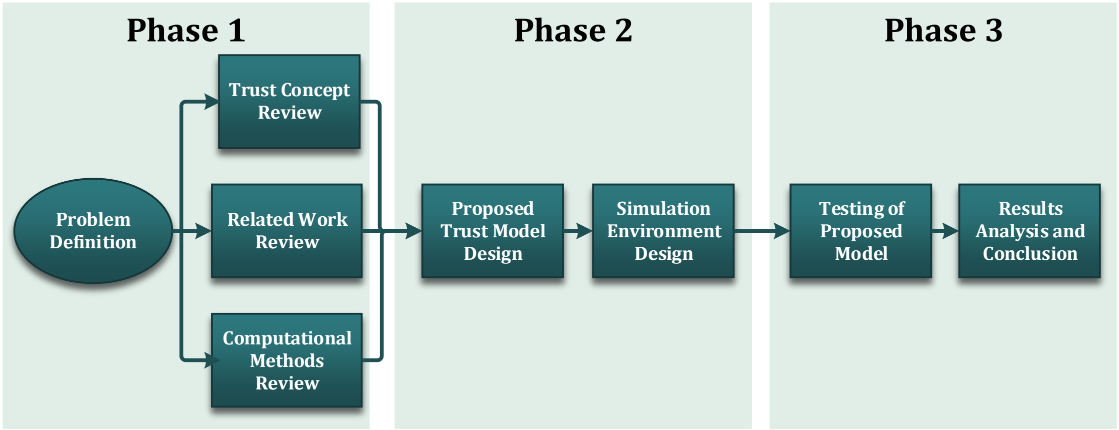

Phase 1
Phase 2
Phase 3
Problem Definition
Trust Concept Review
Related Work Review
Computational Methods Review
Proposed Trust Model Design
Simulation Environment Design
Testing of Proposed Model
Results Analysis and Conclusion

Figure 2-5: Research Process

2.3.2.1 Phase 1

Phase 1 consisted of problem definition and literature review. The main focus of the phase was to understand the research area and clearly identify the outstanding problems. Multiple kinds of literature on related work were reviewed in order to refine the problem statement. The literature on trust concepts was also reviewed. The review of trust concepts highlighted trust components that are important in trust modelling. Finally, different computation methods were reviewed. The main focus of this review was to identify the most suitable method for the IoT trust model.

2.3.2.2 Phase 2

Phase two was carried out in steps: Model Design and Simulation environment design. These two steps were repeated multiple times. The next two sections discuss these steps in more details.

2.3.2.2.1 Model Design

This is the first step of phase 2. Figure 2-6 shows the steps which were taken in designing the model. Before the design of the trust model, the following questions were taken into consideration:

- How is trust modelled?
- What factors should be taken into consideration when modelling trust?
- What factors should be neglected when modelling?
- What computation methods are appropriate when modelling trust?

These questions were addressed in phase 2.

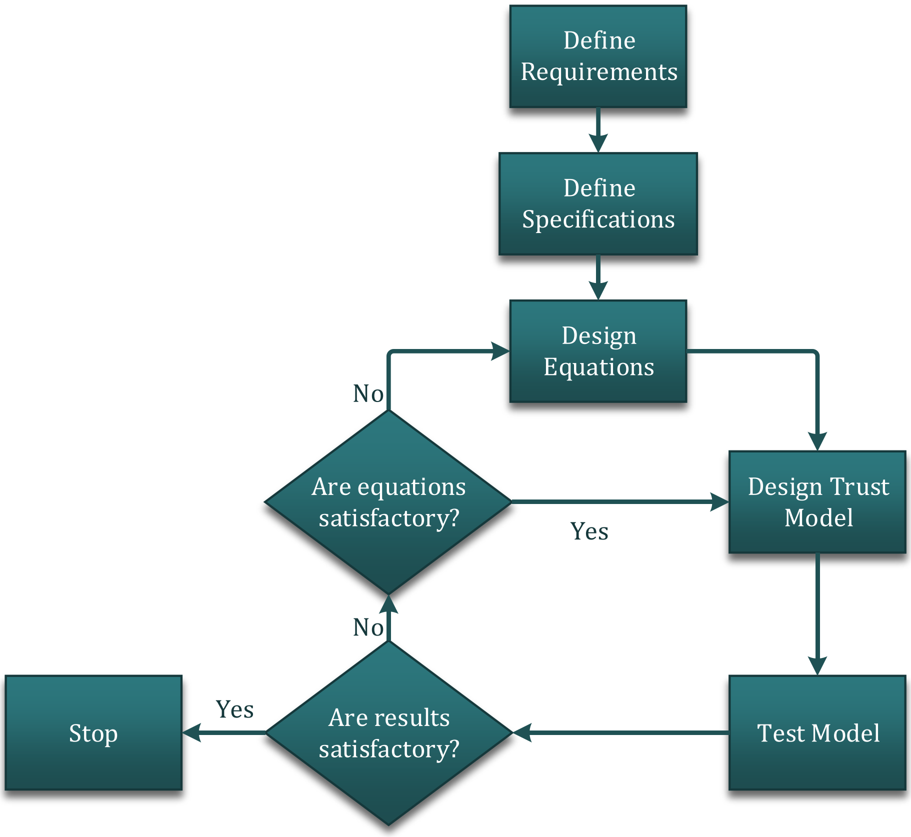


Figure 2-6: Model Design Process

The research follows the following steps in designing and testing the trust model:

- **Define requirements:** the requirements were specifically for an IoT environment. They included the requirements of the environment as well as the requirements for different IoT 'things'.
- **Define specifications:** these were the specifications of the trust model.
- **Equations Design:** The equations were grouped into three groups: first-hand equations, recommendations, and trust decay equations.
- **Design the trust model:** In this step, the trust model was designed and implemented. The model was divided into multiple components and was designed component by component.
- **Test model:** This testing was focussed on the performance of the model in identifying malicious entities and on the performance of the equations.
- **Analyse results:** After analysing the results of the testing, the equations were adjusted and the model was modified. The testing was repeated while making adjustments to the equations until the performance of the model met the benchmarks that were set.

2.3.2.2.2 *Simulation Environment Design*

The model was tested in a simulated environment. The designing of the simulation environment consisted of designing of an IoT environment that consisted of entities with varying capabilities and integrating the proposed model in the simulated environment. The details of the implementation of the simulated environment are covered in chapter 7. The simulated environment was implemented using the prototype model mainly because the process quickly provides a functional system. Figure 2-7 summarises the design and development of the simulation environment.

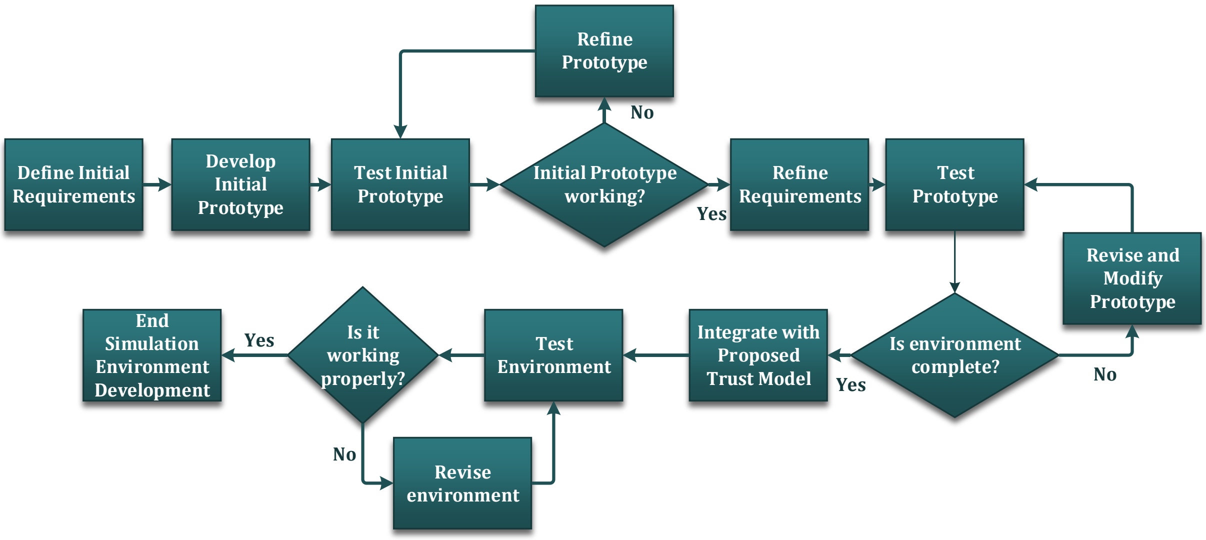

Refine Prototype
No
Define Initial Requirements
Develop Initial Prototype
Test Initial Prototype
Initial Prototype working?
Yes
Refine Requirements
Test Prototype
Revise and Modify Prototype
End Simulation Environment Development
Yes
Is it working properly?
Test Environment
Integrate with Proposed Trust Model
Is environment complete?
Yes
No
No
Revise environment

Figure 2-7: Simulation Environment Development

2.3.2.2.3 *Testing of the Proposed Model.*

This research follows both the experimental and the simulation approach. The trust variables will be manipulated in a simulated environment. Experimenting is difficult in computer science because it is difficult to find benchmarks that can be used in the experiments. The main reason for this is both the artefact and the testing environment affects the output of the experiment. The value of a computer science artefact is measured in many ways that include but not limited to, efficiency, effectiveness, and operationality (Tedre, 2007). The experiment carried out on an artefact should give results on:

- The performance and the efficiency of the system in different testing environments
- The evaluation of all the parameters considered in the experiment
- If possible, the comparison of the artefact with similar artefacts

All these points were taken into consideration during the testing of the model. In computer science, testing is carried out to establish the reliability of a system (Brey & Søraker, 2009). Benchmarks from existing models were used to determine the strength and the weakness of the model. The trust model was tested in a simulated IoT environment. The environment consisted of malicious 'things', selfish 'things' and honest 'things'. Simulation was appropriate because the IoT environment is still in its infancy.

2.3.2.3 Phase 3

During the testing of the model, the output from the trust model was collected as results. The testing of the model was done multiple times in order to validate the consistency of the results. These results were analysed in a reliable, consistent and unbiased way such that they will be reproducible. The details of the results analysis are recorded in chapter 7. This phase also included the conclusion of the research which is chapter 8 of this research.

2.4 Conclusion

This chapter showed that if the focus of the philosophy of computer science is on its aims, methods, and assumptions; it becomes a branch of the philosophy of science (Brey & Søraker, 2009). The chapter also proved that the multi-disciplinary dimension of computer science makes it difficult to carry out its research. In 1976 Wegner stated that "*The computer scientist should be a 'universalist', having the enquiring mind of the empirical scientist, the modeling and abstraction ability of the mathematician, and the tool building and implementation ability of the engineer*" (Wegner, 1976). This statement was confirmed throughout this chapter. The chapter also showed that the value of the artifacts designed in computer science is mainly based on their usefulness and their costs and not necessarily on novelty (Brooks, 1996). The chapter also explained that the usefulness of the artifact is proved through experiments or simulations. This chapter concluded by explaining the research approach and methodology taken to carry out this research. All research is done by reviewing what has been done in order to uncover what hasn't been done, therefore the next two chapter reviews what has been done in regard to trust and trust management.

3

The Concept of Trust

3.1 Introduction

Trust is a social concept that is affected by many factors. The factors that affect trust are determined by the field of the study. A lot of work on trust in '*a human social environment*' has been covered in fields such as economics, philosophy, psychology, and sociology. In computing, the work on trust has been rapidly increasing over the last two decades. However, each of these disciplines limits the aspect of trust to suit their requirements.

Trust is a human element which is complex. It is a social phenomenon because it emanates from the society of human beings. The human mind is capable of addressing uncertain, complex and ambiguous problems using it. Trust enables us to reason based on an approximation of risk before we cooperate with each other. People are cooperative because cooperation may:

- produce profits
- create communal relationships which can lead to collaboration
- It generates coordination is societies

This means trust is important in order to create relationships, produce profits and in carrying out effective collaboration. In trust there is always an element of uncertainty, however with uncertainty also comes optimism. According to (Barber, 1983) we trust because of "*moral social order*" and an expectation of a "*technically competent role performance*". This means trust implies the fulfilment of an expectation of a fiduciary obligation and responsibility (Thomborson, 2010). Trust cannot be generalised. There is always an element of uncertainty and risk in trusting someone. Uncertainty may be caused by ignorance and lack of evidence.

Computational trust seeks to emulate human trust in order to reduce uncertainty and yields profits which include secure collaboration among entities. (S. Marsh, Basu, &

Dwyer, 2012) argues that without the human element, there is a lost link in trust. This chapter reviews the concept of trust and discusses the view that this research will take in order to include the human element of trust in the proposed model.

3.2 Trust

Trust is a human notion which is difficult to estimate in a computing environment. Trust is complex and multi-dimensional (Lewis & Weigert, 1985). Currently, there is no agreed-upon definition of trust, the definition adopted by different authors depends on the discipline. Trust has been defined as follows in different dictionaries:

- Oxford Dictionary: "a *firm belief in the reliability, truth, or ability of someone or something*".
- Merriam Webster: "*assured reliance on the character, ability, strength or truth of someone or something*"
- Cambridge Dictionary: "*to believe that someone is good and honest and will not harm you, or that something is safe and reliable*"

The first definition raises three factors: reliability, truth, and ability. The second definition raise: reliance, ability strength or truth. These factors are estimated as an expectation in the presence of risk and uncertainty. Assured reliance shows the willingness of the truster to be vulnerable by relying on the trustee in the presence of risk and uncertainty. Uncertainty in an IoT environment may be caused by information and opportunism (Gu, Wang, & Sun, 2014). Willingness to be vulnerable enables the building of trust. Trust cannot be built without being vulnerable and taking risks. The ability of the trustee is very important because it adds more assurance to the success of a transaction. Reliability and truth determine the behaviour and the willingness of the trustee. The last definition raises harm and honest. In this sense, trust is having confidence that the trustee in terms of the intentions and behaviour of the trustee. These factors lay the basis of trust between a truster and a trustee which can mature into a trust relationship. Trust estimates the expectation of the truster in the trustee. In addition to these factors, trust also pertains to a particular context and time.

In this research, trust was defined based on the definitions by (Dasgupta, 1988) and (Gambetta, 2000). The first one is by (Dasgupta, 1988) who defined trust as, "*sense of correct expectations about the actions of other people that have a bearing on one's choice* *of action when that action must be chosen before one can monitor the actions of those others*" (Dasgupta, 1988). The actions that the device in the IoT environment will take should depend on the presence or absence of trust. The second one is by (Gambetta, 2000): "t*rust (or symmetrically, distrust) is a particular level of the subjective probability with which an agent will perform a particular action, both before [we] can monitor such action (or independently of his capacity of ever to be able to monitor it) and in a context in which it affects [our] own action*". Trust is affected by those actions that cannot be monitored (Abdul-Rahman & Hailes, 1997). These definitions concentrate on the behaviour and actions of the trustee without including time and context. Time and context are very important factors when evaluating trust which needs to be taken into consideration. The evaluation should focus on both the intentions and behaviour of the trust. The accumulation of the behaviour of the trustee of time may be used to determine the intentions of the trustee.

A discussion on trust is not complete without discussing reputation because direct experiences may not always be available. When direct experiences are not available the trustworthiness of the trustee may be evaluated using reputation. Yao Wang & Vassileva, (2003b) defined reputation as *"a peer's belief in another peer's capabilities, honesty and reliability based on recommendations received from other peers".* This definition highlight that reputation is obtained from recommendations based on past behaviour. However, the truster's experience with the trustee may also be included in computing reputation. Azzedin & Maheswaran (2002) defined reputation as an expectation of an entity's *"behaviour based on other entities' observations or information about the entity's past behaviour within a specific context at a given time".* In this research, the term 'thing' and entity are used interchangeably. Similar to trust, context and time are also important when computing reputation. Although trust and reputation are closely related, they are not synonymous. The main difference between trust and reputation is how they are created. Trust and reputation share the same properties and characteristics. The following section takes a closer look at the properties of trust.

3.2.1 Properties of Trust

Trust is a dynamic construct which is affected by many factors. Trust is obtained from prior knowledge and experience (Firdhous, Ghazali, & Hassan, 2011). Experience determines the reputation of an entity. Reputation is the main factor that has been used to evaluate earlier behaviour and performance of a trustee (Z. Yan, Zhang, & Vasilakos, 2014). The factors that affect trust are difficult to both measure and monitor (Zheng Yan & Holtmanns, 2007). Trust is affected by factors which include the information known, the ability, the available options and their consequences and so on. Table 3-1 shows the factors that were summarised from the definitions of trust by (Zheng Yan & Holtmanns, 2007).

Table 3-1 - Factors Influencing Trust (Zheng Yan & Holtmanns, 2007)

	Property	**Factors**
Trustee's	Subjective	competence; ability; security; dependability; integrity; predictability; reliability; timeliness; (observed) behaviour; strength
	Objective	Honesty; benevolence; goodness
Truster	Subjective	Assessment; a given set of standards; truster's standards
	Objective	confidence; (subjective) expectations or expectancy; subjective probability; willingness; belief; disposition; attitude; feeling; intention; faith; hope; truster's dependence and reliance
Context		Situations entailing risk; structural; risk; the domain of actions

From the digital system point of view; trust focuses on the objective properties of both the truster and trustee (Zheng Yan & Holtmanns, 2007). A research done by (Mcknight & Chervany, 2000) identified benevolence, integrity, competence and predictability as characteristics of trust. The researcher argues that these are factors because they influence trust rather than describe it. The following characteristics of trust were also identified by (Sabater & Sierra, 2005): reliability, honesty, sincerity, Quality of Service, pre-visibility (Sabater & Sierra, 2005) and (Z. Yan et al., 2014) identified cooperative and community of interest. One of the objectives of this research is to come up with properties that are relevant for trust evaluation in the IoT environment. From all the above factors, the following were identified as important for the IoT environment:

- Integrity
- Competence
- Predictability

- Quality of Service (QoS)
- Community of Interest
- Reliability
- Risk
- Consistency
- Sincerity
- Commitment

The behaviour of the IoT device will be used to determine its predictability. The risk involved determines the minimum trust value that is required for any collaboration to be carried out. It should be noted that the properties of trust are infinitely many and it is impossible to take them all into consideration when building a trust model.

3.2.2 Characteristics of Trust

Researchers have suggested different characteristics for trust over the past few decades. The following are the ones that were found to be relative for the IoT environment:

- Context-specific (De Meo et al., 2009; Ries, Kangasharju, & Mühlhäuser, 2006; Y Wang, Cahill, Gray, Harris, & Liao, 2006; Yao Wang & Vassileva, 2003a)
- Based on prior experiences (Abdul-Rahman & Hailes, 2000)
- Dynamic – it changes over time depending on the behaviour of the trustee (Bao, 2013; Firdhous et al., 2011; Ries et al., 2006; Yao Wang & Vassileva, 2003a)
- Non-monotonic (Abdul-Rahman & Hailes, 2000)
- Multi-dimensional (De Meo et al., 2009; Y Wang et al., 2006)
- Trust is not transitive (Abdul-Rahman & Hailes, 2000)
- Subjective (Abdul-Rahman & Hailes, 1997; Firdhous et al., 2011; Ries et al., 2006; Zhang, Yu, & Irwin, 2004) (personal opinion)
- Objective (Zhang et al., 2004) (universal standard)
- Recommendation is obtained through the exchange of reputation information (Abdul-Rahman & Hailes, 2000)
- Scope of relevance (De Meo et al., 2009)
- Trust supports both positive and negative degrees of belief (Abdul-Rahman & Hailes, 2000)

The context of trust is also very important because it determines how and when to trust. Trust can be classified as subjective because its level differs depending on the entity. Since trust is not symmetric, it should follow that if the trustee does not trust the truster, the trustee might refuse any association with trustee.

In the IoT environment, there should be mechanisms that define trust in a dynamic and collaborative way (Roman et al., 2011). Chang, Dillon, & Hussain (2005) elaborated that trust is dynamic because:

- as the number of interactions increases, the truster will get a better understanding of the trustee
- the willingness of the trustee "*may vary over time*"
- recommendations about the trustee from other entities may affect the trust value that the truster has in the trustee

3.2.3 Factors to consider in trust computation

Trust can be classified in different ways such as either behavioural trust (trust among organizations and people) or computational trust (trust among networks, computers and devices). This research focuses on computational trust in IoT. Computational trust can also be classified based on:

- Observation type
- Source
- Mode of trust
- Property

Figure 3-1 shows the classification of computational trust. Observation type describes the method that is used to obtain the information that is used in computing trust values. Direct trust occurs when the trust is computed for the truster's direct experiences while indirect trust is computed from the recommendations. Data trust is based on the evaluation of the data that is exchanged between the truster and the trustee. Communications trust is based on the evaluation of the mode of communication between the trustee and the truster.

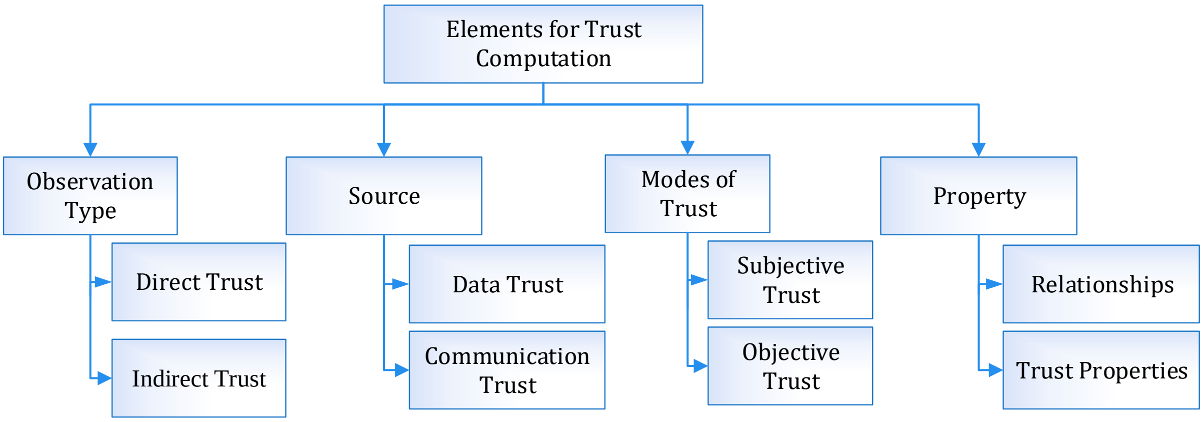


Figure 3-1: Trust Classification

According to (Tong, Zhang, Long, & Huang, 2013) trust has three modes: objective trust, subjective trust and transitive trust. Objective trust is based on experience from personal interaction, subjective trust is based on interaction experience with personal preference and transitive trust is based on recommendations (Tong et al., 2013). (Nitti, Girau, & Atzori, 2014) defined objective and subjective trust in IoT based on social networking. (Nitti et al., 2014) subjective trust as trust that is based on direct experience and recommendations from friends, and objective trust as the reputation of the trustee in a peer-to-peer scenario. (Tong et al., 2013) suggested that the trusters which are agents, in this case, will first obtain objective trust from personal interactions. According to (Tong et al., 2013) the trustees will then add personal preferences to the objective trust in order to get subjective trust and at the end, they will determine if transitive trust exists. In this research The researcher agrees with (Nitti et al., 2014) that trust can either be subjective or objective.

According to (Umarani & Sundaram, 2013), social trust determines intimacy, honesty, privacy, centrality, connectivity and QoS trust considers energy, unselfishness, competence, cooperativeness, reliability, task completion capability. (Gutscher, Heesen, & Siemoneit, 2008) categorised trust as functional trust and recommendation trust. They defined functional trust as the intentions and the competence of the trustee. In this research trust is considered as follows:

- Direct trust
- Indirect Trust
    - Recommendation-based trust – trust computed from recommendations obtained from the truster's trusted recommenders.

- Reputation-based trust – trust obtained from the reputation system.

Recommendations are obtained from reliable recommenders only. Chapter 6 will discuss these elements in detail.

3.3 Trust Considerations in transactions

(Chang et al., 2005) identified some of the building blocks of trust as follows: truster, trustee, willingness, capacity, delivery, mutually agreed service, context, timeslot. Willingness pertains to the interacting intentions of the trustee with the truster in a given context and at a specified time. The researcher acknowledges that there are instances where trust has to be bi-directional. For example, when sensitive data has to be exchanged. In such cases, trust occurs when the truster has confidence in the reliability and integrity of the trustee. However, for the transaction to occur, the trustee also needs to be willing to trust the truster. In the event that that the trustee does not trust the truster, the trustee will reject the transaction. It should be noted that the trust of the trustee pertains to the transaction that the truster is requesting and the information that will be exchanged.

3.4 The Relationship Between Trust and Distrust

(S. Marsh & Dibben, 2005) proposed that apart from trust there exists distrust, mistrust and untrust. Currently, there is no clear agreed-upon definition of distrust. (S. Marsh & Dibben, 2005) defined it as a belief that a trustee will work against the interests of the truster in a specific context. In other words, distrust is being totally convinced that the trustee is intentionally untrustworthy. According to (S. Marsh & Dibben, 2005) distrust is not the negation of trust but rather a form of negative trust. According to (Gutscher et al., 2008), a trustee is distrusted if the truster believes that the trustee is not competent or has malicious intention. This research in partial agreement with (S. Marsh & Dibben, 2005) and disagree with (Gutscher et al., 2008) that incompetence causes distrust. This is because competence is not the only factor that is taken into consideration when making the decision to distrust. Since a distrusted trustee will rarely be trusted again, distrusting an entity should not be based on a single factor. Distrust can be perceived as a partner of trust which is more complex than trust (Cofta, 2006). This makes it important in a trust management model.

(Tang, Hu, & Liu, 2014) in their conclusion raised the point that "*if distrust is the negation of trust, lacking distrust study matters little; while if distrust is a new dimension of trust, ignoring distrust in trust study may yield an incomplete and biased estimate of the effects of trust*". Using social network data, they showed that distrust is not the negation of trust and suggested that there was a need for further research in order to understand distrust. Distrust and trust exist together and they are both based on evidence. A research done by (Lewicki, Mcallister, & Bies, 1998) concluded that trust and distrust are separate because:

- They coexist
- Empirically they are disjointed.

"*Trust and distrust come from different sides of the personality and each finds its basis in a different concept of human nature*" (Mcknight & Chervany, 2000). Considering distrust as a form of negative trust implies that distrust inherits the properties and characteristics of trust such as distrust is context specific and decay over time. This raises the following question: "Can a trustee which is distrusted in a certain context be trusted in a different context?". This research takes the view that a trustee with malicious intentions in one context is malicious in all the other contexts. Therefore, distrust is not context specific. The researcher is also of the view that distrust does not decay over time.

Mistrust is "*misplaced trust*" (S. Marsh & Dibben, 2005). Simply put, mistrust is when a truster trusts an untrustworthy trustee. Even though some researchers on trust sometimes use distrust and mistrust as synonyms for example (Gutscher et al., 2008; Jøsang, 1996; Mcknight & Chervany, 2000), distrust and mistrust are different (Cofta, 2006; S. Marsh & Dibben, 2005). It should also be noted that mistrust is not the lack of trust. In this research, trust will also be taken to mean that a truster has undistrusted a trustworthy trustee since Undistrust is a form of negative trust. Therefore, based on the reviewed literature in this research, it is concluded that mistrust is misplaced trust or misplaced distrust.

Untrust is trust that is not enough for cooperation. It shows that there is a chance that the trustee is trustworthy but there is a need for verification. This kind of trust requires more verification before a decision to trust can be taken. Apart from trust, distrust, mistrust and untrust (Griffiths, Chao, & Younas, 2006) introduced undistrust. They defined it as negative trust that occurs when there is no sufficient information to make a conclusion to distrust. If there is no sufficient evidence to distrust a trustee, the trustee can be undistrusted while more evidence is being gathered. This minimizes the chances of a trustworthy trustee from being distrusted by a truster. Figure 3-2 shows how Trust (T), Untrust (UT), Undistrust (UD) and Distrust (D) are related.

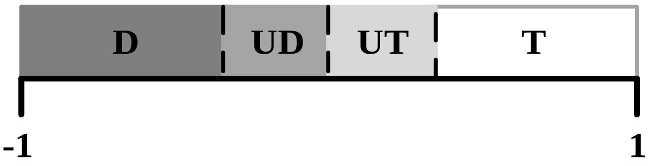


Figure 3-2: Notion of trust (Griffiths et al., 2006)

Trust and untrust is a positive measure while distrust and undistrust are both negative measures. There may be insistences where the truster will be ignorant of the trustee. This ignorance may lead to mistrusting the trustee. It should be noted that ignorance of the trustee is not the same as distrust or undistrust. Distrust should not be the result of lack of information. Modelling ignorance as distrust will disqualify trustworthy trustees. With this in mind, how will the ignorance of the truster about the trustee be labelled in a trust model? Does providing a value for ignorance improve the trust value? These questions will be addressed in Chapter 6 and Chapter 7.

Trust values in the range of [0,1] where 0 represents lack of trust and 1 represent full trust have been widely used. The range of [-1, 1] has also been used. In this range, -1 represents either lack of trust or distrust and 1 represents full trust. (Gutscher et al., 2008) suggested that negative trust values might be useful for environments in which "*the possible harm of unsuccessful interactions is high*". Both of these methods of trust representations treat distrust as a direct lack of trust. However, full trust and full distrust do not exist as there is always an element of uncertainty in the future. Both trust and distrust aim to increase certainty and reduce uncertainty.

3.4.1 Relations of Trust and Distrust

This section reviews different relations for trust and distrust.

- **Reflexive** – reflexive exists if entities in a trust domain trust themselves. In general, trust is not reflexive (Grandison & Sloman, 2003; Gutscher, 2007). Considering the contexts of trust, an entity might trust itself in some context and not trust itself in other contexts because it may be lacking the competence to complete the task.

- **Irreflexive** – irreflexive when all entities do not trust themselves. Trust is not irreflexive (Grandison & Sloman, 2003) because there are instances where entities trust themselves to carry out tasks.
- **Transitive** – transitive is when there are entities ***a***, ***b*** and ***c***; in which it follows that if ***a*** trusts ***b,*** and ***b*** trust ***c*** then ***a*** trusts ***c***. In general functional trust is not transitive in the mathematical sense (Golbeck & Hendler, 2006). However, dividing it into functional trust and recommendation trust (Gutscher et al., 2008) enables it to become partially transitive. Recommendation trust is partially transitive and functional trust is not transitive. Recommendation trust is partially transitive (*conditionally transitive*) because a truster still has the option of ignoring the recommendation. (Abdul-Rahman & Hailes, 1997) termed it as conditional transitivity.
- **Intransitive** – intransitive is when for entities ***a***, ***b*** and ***c***; if ***a*** trusts ***b,*** and ***b*** trust ***c*** then ***a*** does not trust ***c***. With the same argument as that of transitivity, trust is not intransitivity.
- **Symmetric** – Symmetric exists if the truster trusts the trustee and the trustee trusts the truster. "*Trust relations are in general not symmetric*" (Cahill et al., 2003; Grandison & Sloman, 2003; Gutscher, 2007). This is because there are some instances in which this exist but its existence cannot be guaranteed in all relations. Also, the truster and the trustee might not trust each other with the same trust value, this means that in such cases symmetry does not exist in the mathematical sense of symmetry. Most trust models that have been proposed neglect the trust value of the trustee to the truster. This might cause a trustee to be rated as selfish while the trustee might have rejected a task from the truster due to the trustee's lack of trust in the truster. This is going to be taken into consideration in our proposed model.
- **Asymmetry** – asymmetry is when the truster trusts the trustee and the trustee does not trust the truster. (Golbeck & Hendler, 2006) argues that trust is asymmetry but their definition of asymmetry considers trust values only without considering trust relation. Also (Ries et al., 2006) states that trust is asymmetry because it is subjective. This research takes the same argument as that of symmetry above, trust is not totally asymmetric (Grandison & Sloman, 2003).

Even though trust does not support any of these relations fully, there are instances where they are supported. Therefore, the trust model should be flexible enough to allow them whenever the need arises without enforcing them for all trust relations.

3.4.2 Constructs of Trust and Distrust

(Mcknight & Chervany, 2000) argues that trust is an interdisciplinary concept which needs to be treated as such. The Figure 3-3 shows the interdisciplinary trust constructs as proposed by (Mcknight & Chervany, 2000).

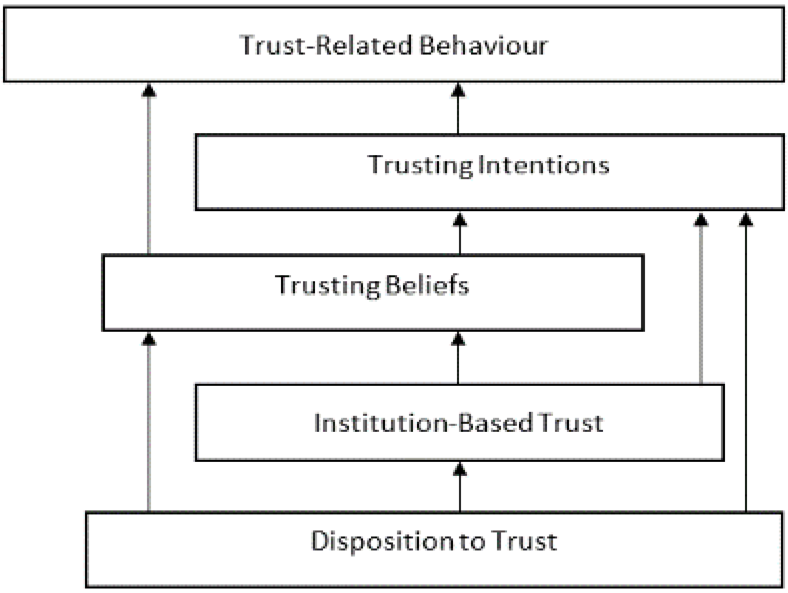


Figure 3-3: Interdisciplinary Trust Constructs (Mcknight & Chervany, 2000)

The links in the diagram are intuitive (Mcknight & Chervany, 2000). (Mcknight & Chervany, 2000) described each of the levels as follows:

- Trusting Intentions – this is the willingness of the truster to depend on the trustee even though there is a possibility of negative consequences.
- Trust-related Behaviour – this is when the truster voluntarily depends on the trustee with a sense of security. The truster has accepted the risk and has given the trustee power.
- Trusting Beliefs – this specifies the extent to which the truster is convinced that the trustee has characteristics that are beneficiary to him/her. Trusting belief includes trusting the competence, benevolence, integrity, and predictability of the trustee.

- Institutional-based Trust – "*Institution-based Trust means one believes, with feelings of relative security, that favourable conditions are in place that is conducive to situational success in a risky endeavour or aspect of one's life*".in this case structures, situations or roles act as an assurance to the truster.
- Disposition to Trust – this is when a trustee is willing to trust others generally. "*Disposition to Trust means the extent to which one displays a consistent tendency to be willing to depend on general others across a broad spectrum of situations and persons. Disposition to trust differs from trusting intentions in that it refers to general other people rather than to specific other people*".

(Mcknight & Chervany, 2000) viewed distrust as negative trust and applied the same constructs to it as shown in Table 3-2. However, even though the researcher agrees with the above levels of trust, the researcher also agree with the findings of (Tang et al., 2014) that distrust is not the opposite of trust. Table 3-2 shows the trust and distrust constructs according to (Mcknight & Chervany, 2000). This research takes these constructs into consideration and investigates their effects in the trust model for both trust and distrust.

Table 3-2 – Levels of Trust and Distrust (Mcknight & Chervany, 2000)

	Interpersonal				
	Dispositional	**Structural**	**Perceptual**	**Intentional**	**Behavioural**
Trust					
Conceptual Level	Disposition to Trust	Institution-Based Trust	Trusting Beliefs	Trusting Intentions	Trust-Related Behaviour
Operational Level	-Faith in Humanity -Trusting Stance	-Structural Assurance -Situational Normality	Trusting Beliefs -Competence -Benevolence -Integrity -Predictability	- Willingness to Depend -Subjective Probability of Depending	-Cooperation -Information Sharing -Informal Agreements -Decreasing Controls -Accepting Influence -Granting Autonomy -Transacting Business
Distrust					
Conceptual Level	Disposition to Distrust	Institution-Based Distrust	Distrusting Beliefs	Distrusting Intentions	Distrust -Related Behaviour
Operational Level	-Suspicion of Humanity -Distrusting Stance	-No Structural Assurance - No Situational Normality	Distrusting Beliefs -Competence -Benevolence -Integrity -Predictability	- No Willingness to Depend -Subjective Probability of Not Depending	-Lack of Cooperation -Information Distortion -Formal Agreements -Increasing Controls -Not Accepting Influence

					-Not Granting Autonomy -No Transacting Business

3.4.3 Discussion

Trust and distrust coexist in the same environment however not necessarily on the same entity by the same truster. To clarify this, take into consideration an example: a patient can trust a doctor with their health issues but untrust them to service their car. We cannot say the patient distrusts the doctor because the doctor's lack of car mechanics knowledge does not mean he has malicious intentions. This shows that distrust is not the negation of trust but rather a form of negative trust. However, such an argument does not hold for distrust. As mentioned earlier, distrust is not context specific. Once an entity is distrusted in one context, it is distrusted in all the available contexts. Therefore, it is imperative to prove the evidence beyond any reasonable doubt before the decision to distrust is taken.

Ignorance of the trustee should not be modelled as either trust or distrust. Ignorance occurs when there is no evidence at all to either trust, untrust distrust or undistrust the trustee. Distrust is justified because it is backed up by evidence. It does not depend on trust or lack of knowledge about the trustee. Distrust should not be seen as an undesired but as a complementing partner of trust which will prevent malicious entities from taking advantage of either the network or the other entries in the network. Once an entity is distrusted, it should be eventually forced out of the network. When distrust is included in a trust management model, it can prevent malicious trustees from being trusted. Undistrust and untrust can prevent an entity which is selfish but not malicious from being distrusted.

3.5 Conclusion

Trust as a human notion, is difficult to formulate in order to enable to allow computers to trust or distrust each other. Trusts depend on the competence and intentions of the trustee (Gutscher et al., 2008). Trust is self-reinforcing. Trust is dynamic and time-dependent.

According to most research that has been done, trust is based on the reputation that is acquired through behaviour over time. This basis has a limitation in the IoT environment because for most devices, won't have time to build a reputation before they can start communication and collaboration with other devices. This means there is a need for other avenues for building trust. This research will also take into consideration ignorance, untrust, distrust and undistrust. Both trust and distrust are based on the intentions and capabilities of the trustee. However, since distrust has not been thoroughly studied for trust management in the IoT environment, the research also investigated its effects by designing two models. One which includes distrust together with undistrust and the other one does not include distrust. Both trust and distrust can be used to avoid malicious 'things' in the IoT environment. Most research on trust and distrust called for further research (Lewicki et al., 1998; S. Marsh & Dibben, 2005; Mcknight & Chervany, 2000).

4

Current State of IoT Trust Management

4.1 Introduction

The IoT has the potential of creating a universal and ubiquitous internet of devices which requires minimal human supervision. However, most of the IoT devices have limited resources such as computation ability, memory and power, and hence are not capable of providing security on their own. The use of trust management models can enhance security on these devices. The past few years have seen more researchers proposing trust management for IoT as well. However, most of the proposed models are for specific IoT environments such as sensor networks.

The main aim of a trust model is to reduce risk by accurately estimating the trust values of the trustee based on the information that is about the trustee. This information may include previous trust information, relationship with truster and reputation. This will enable the truster to avoid malicious entities and malicious transactions. The modelling of trust needs to be done in a cost-effective manner in order to keep the overall cost of the system minimum. This chapter reviews the current state of the IoT trust management. Part of this chapter is an extract of a conference paper in which the researcher was the main author published in IST–Africa 2017 International Conference. The contribution of the author in that paper is 100% of the content.

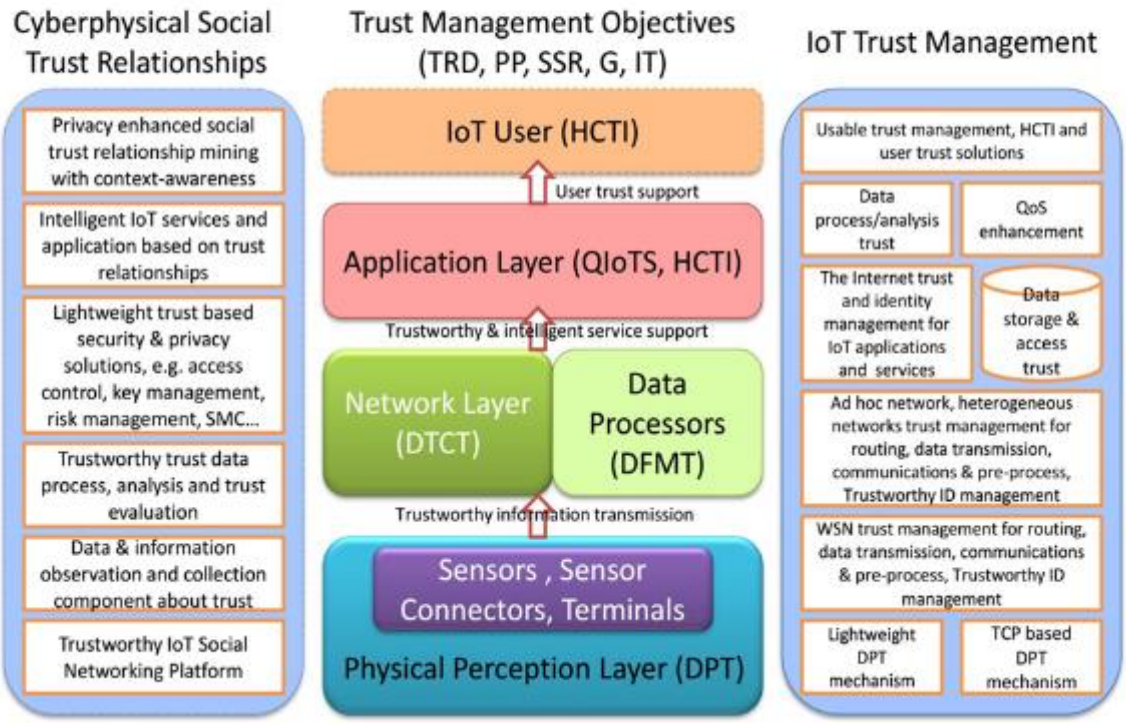
4.2 Objectives of Trust Management in the IoT environment

It is important for a trust management model to exhibit social trust characteristics so as to achieve trust management objectives. In IoT, the main objective of trust management is to identify malicious and selfish entities in an environment. Apart from malicious entities, trust management can also be used to identify malicious activities such as malicious recommendations. This provides the IoT environment with an avenue for "*secure, reliable, seamless communications and services*" (Truong, Lee, Askwith, & Lee, 2017). Trust management can defend the IoT network from malicious internal attacks. Z. Yan et al., (2014) identified the following objectives for trust management:

- **Trust Relationship and decision:** the trust management system has to provide a way to evaluate trust relationships among the IoT entities. Trust relationships can be used as the basis of trust computation among entities.
- **Data perception trust:** the data that is collected should be reliable (sensor sensibility, preciseness, secure, reliable, persistence and it's should be collection efficiency). It is important that entities trust that their data is safe and secure in all collaborations. Also, entities need to be assured that they are obtaining accurate data from other entities.
- **Privacy preservation:** user data should be preserved according to both the user expectations and the policy being implemented.
- **Data fusion and mining trust:** data processing and analysis should be trustworthy (reliability, holographic data process, privacy preservation and accuracy).
- **Data transmission and communication trust:** the transmission of data should be secure in the IoT environment. Only authorized parties should have access to IoT data during transmission. This can be achieved through trusted routing and key management.
- **Quality of IoT services:** "*only here, only me and only now services are expected in IoT*". This requires the personalization of IoT services. This objective requires trust management model to include Quality of Service (QoS) as one of the properties that should be included in trust evaluation.
- **System security and robustness:** "*trust management in IoT should effectively counter system attacks to gain sufficient confidence of IoT system users. This objective concerns all system layers, focusing on system security and dependability, which are about the trustee's objective properties*".
- **Generality:** a trust management system should be able to be used in different contexts.

- **Human-computer trust interaction (HCTI):** this objective will enable the IoT to be adopted by users. It focusses on the subjective properties of the users (trusters).
- **Identity trust:** each 'thing' in the IoT environment should be uniquely identified and the trust management system should be able to manage all these identities. This objective should be supported by all layer.

According to Z. Yan et al. (2014), trust relationship and decision, privacy preservation, system security and robustness, generality and identity trust are very important for achieving trustworthiness in the IoT environment. Trust enables entities to establish trustworthy collaborations and improve the performance of the network. *Figure 4-1* shows a holistic trust management framework proposed by (Z. Yan et al., 2014).



Cyberphysical Social Trust Relationships
Trust Management Objectives (TRD, PP, SSR, G, IT)
IoT Trust Management
Privacy enhanced social trust relationship mining with context-awareness
Intelligent IoT services and application based on trust relationships
Lightweight trust based security & privacy solutions, e.g. access control, key management, risk management, SMC...
Trustworthy trust data process, analysis and trust evaluation
Data & information observation and collection component about trust
Trustworthy IoT Social Networking Platform
IoT User (HCTI)
User trust support
Application Layer (QIoTS, HCTI)
Trustworthy & intelligent service support
Network Layer (DTCT)
Data Processors (DFMT)
Trustworthy information transmission
Sensors , Sensor Connectors, Terminals
Physical Perception Layer (DPT)
Usable trust management, HCTI and user trust solutions
Data process/analysis trust
QoS enhancement
The Internet trust and identity management for IoT applications and services
Data storage & access trust
Ad hoc network, heterogeneous networks trust management for routing, data transmission, communications & pre-process, Trustworthy ID management
WSN trust management for routing, data transmission, communications & pre-process, Trustworthy ID management
Lightweight DPT mechanism
TCP based DPT mechanism

Figure 4-1: A holistic trust management framework proposed by Yan et al. (2014)

The framework shows that a trust model is one of its components. Figure 4-2 shows an SOA-based architecture that was proposed by (Atzori et al., 2010). Both the SOA-based architecture and the framework proposed by (Z. Yan et al., 2014) proposes a trust management framework that covers all the protocols. Such a trust model will use trust properties for different protocols in computing trust and the model will be able to provide both data trust and communication trust.

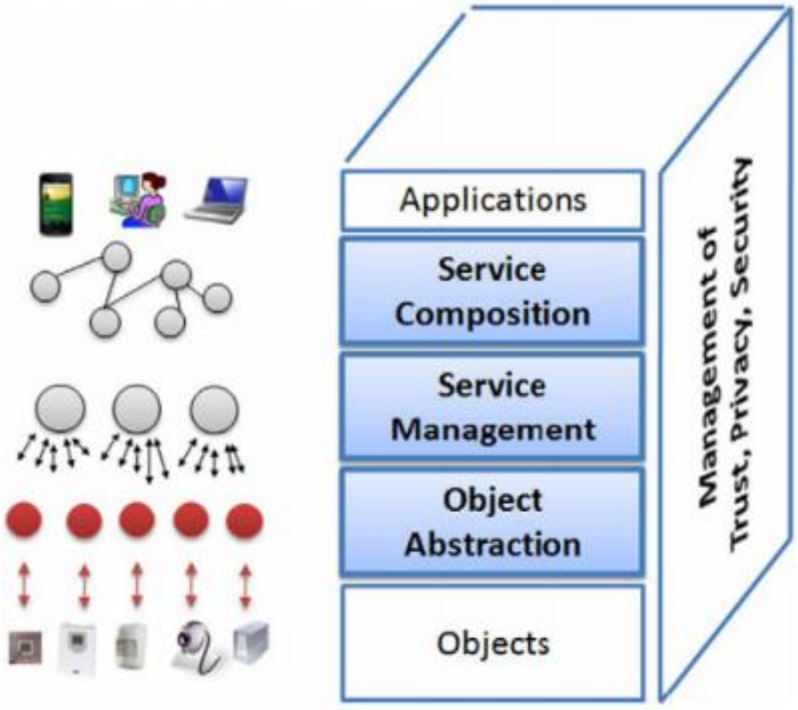


Figure 4-2: SOA-based architecture for the IoT middleware (Atzori et al., 2010)

This research aims to develop a trust model that will take into consideration most of the objectives mentioned above. It is important that the trust framework covers all or multiple layers of the stack. This enables the 'things', communication medium and data to be protected from malicious 'things'. Both frameworks highlight the importance of generic trust models in the IoT. According (Fullam et al., 2005), the objectives when building a trust model should include accuracy, adaptive, quick convergence, multi-dimensional and efficiency. These objectives should be taken into consideration when testing a trust model.

4.3 Composition of a Trust Model

A generic trust model enables different IoT to be connected in the same trust network and to securely cooperate. All this was taken into consideration during the design and development of our proposed trust model. The trust properties mentioned in Chapter 3 are important in evaluating trust values. These properties are the main components of an IoT model which enable the model to achieve the objectives of trust. Different researchers have applied these properties in trust models in different ways. The most outstanding way was done by (Jayasinghe, Otebolaku, Um, & Lee, 2017). (Jayasinghe, Lee, & Lee, 2017) proposed that the properties be modelled as shown in Figure 4-3. This method has the limitation that properties identified under social attributes are only functional if there is a social relationship among the truster and the trustee. Even though social relationships are important in IoT, they are not the only way to define relationships in IoT. As mentioned in Chapter 3, the context of trust is very important. Therefore, is important to identify properties that are relevant in each context and prioritize them.

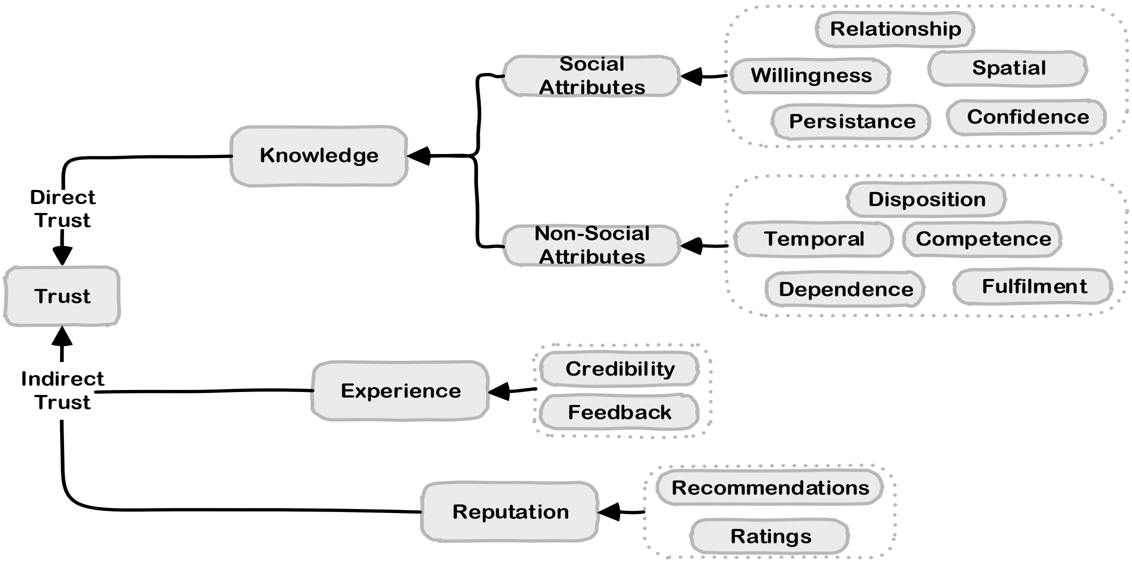


Figure 4-3: A Generic Trust Model (Jayasinghe, Otebolaku, et al., 2017)

Relationships among IoT entities were taken into consideration in this research. Trust relationships improve the accuracy of trust values. Proposed trust relationships in IoT include Social Internet of Things (SIoT) and Community of Interest (CoI). SIoT is based on the social relationships of the owners of the entities. CoI occurs when entities have common interests. Trust relationships enable entities in a network to create trust networks. Knowledge and the ways of discovering knowledge subjective. (Hoffman, Lawson-Jenkins, & Blum, 2006) proposed a generic trust model shown in Figure 4-4.

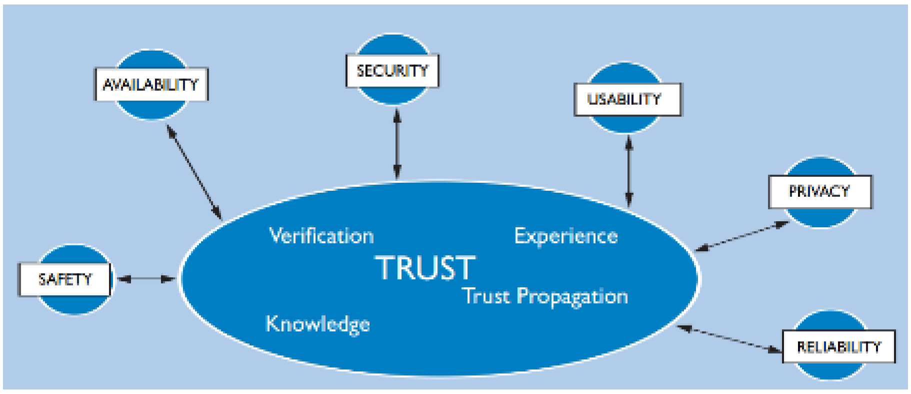


Figure 4-4: Generic Trust Model (Hoffman et al., 2006)

The model has the limitations that it includes hard security measures which are not suitable for the IoT environment and it does not address internal attacks. However, the model highlights the importance of security, reliability and availability as well as trust propagation in a trust model.

4.4 Advances towards Trust Management in IoT

The security and privacy issues in the IoT can be addressed by a trust management solution. The term Trust Management was coined by Blaze, Feigenbaum, & Lacy (1996) and they produced the first trust management framework. The framework focused on security policies, credentials and trust relationships (Blaze et al., 1996). Ever since then, there have been many trust management models that have been proposed. A Trust management model is a means that can be used to differentiate trustworthy and untrustworthy nodes in a network. This section focuses on different types of models that have been proposed for the IoT environment.

Some of the researches which have been done have proposed the use of Social IoT to provide a trustworthy IoT environment. SIoT is "*a social network where every node is an object capable of establishing social relationships with other things in an autonomous way according to rules set by the owner*" (Nitti et al., 2014). The relationships among devices in SIoT are based on the relationships of the owners of the devices. In 2012 Bao and Chen proposed two trust management for IoT (Bao & Chen, 2012b, 2012a) which are based on SIoT and Quality of Service (QoS). (Bao & Chen, 2012a) is an extension of (Bao & Chen, 2012b).They identified that all human-related devices are prone to malicious attacks because they communicate through wireless technologies. The trust management models they proposed are based on social relationships which include friendship, ownership and community. The social relationships of the devices are related to the relationships of their owners. They also identified that in IoT, malicious nodes perform the following trust related attacks: bad-mouthing, good mouthing and self-promoting (Bao & Chen, 2012a). The models are distributed, encounter-based and activity based. The following parameters were used for trust evaluation: honesty, cooperativeness and community interest. These models are able to adapt to the changing environment but their trust evaluation method is not adaptive so as to cope with the IoT dynamic environments. These models can only be applied in an environment where social relationships exist and to devices that have the computational ability.

Another trust model that is based on social trust and QoS was proposed by (I.-R. Chen, Bao, & Guo, 2015). This model is distributed and adaptive to the IoT environment as well. The model also “analyse the trade-off between trust convergence speed and trust fluctuation to identify the best protocol parameter settings for trust propagation and aggregation to best exploit this trade-off for minimizing trust bias”. It also addresses some of the issues of trust formation. However, the model did not address how trust will be evaluated if a node in the network has no social relationships with all the other nodes in the network.

Netti *el al* also proposed a social based trust model (Nitti, Girau, Atzori, Iera, & Morabito, 2012). The social relationship dimensions of this model are also based on the social relationships based on device ownership. The model is a subjective trust management model based on solutions that have been proposed for P2P networks (Kamvar, Schlosser, & Garcia-Molina, 2003; Liang, Shi, & Group, 2004; Selcuk, Uzun, & Pariente, 2004; Xiong & Liu, 2003; B. Y. Bin Yu, Singh, & Sycara, 2004). The model addresses certain types of malicious behaviours by using reputation-based trust mechanism. The model seeks to promote communication between trusted nodes only. Since in this model, trust is calculated based on experience and opinions of common friends. A problem also arises when there are no social relationships.

Two other models, Subjective Trustworthiness and Objective Trustworthiness based on social trust were proposed by (Nitti et al., 2014). The subjective trustworthiness, trust is based on direct experience of each node and the direct experience of the node's friends. While in the objective model, trust is based on the experiences of all the nodes that are in the network and is stored centrally. The belief in objective trustworthiness is that trust is only composable. The objective trustworthiness model is not suitable for the IoT environment as it requires trusted nodes that maintain the trust tables in advance which might not be possible in an IoT environment due to resource constraints. Their subjective model is prone to bias as it relies on the experiences of the friends which can be biased or become compromised.

Mahalle, Thakre, Prasad, & Prasad (2013) proposed a Trust-Based Access Control (FTBAC) model which uses the fuzzy approach (Mahalle et al., 2013). The model uses the FTBAC framework to calculate trust values. The trust values are calculated using factors such as recommendation, knowledge and experience. The permissions and access that a device will be granted are mapped from the trust values. The FTBAC framework has the following layers:

- Device layer: includes all the IoT devices
- Request Layer: is responsible for collecting information about the factors that are used to calculate the trust values.
- Access control Layer: is responsible for decision making. It maps all the trust values that have been calculated to access permissions.

The FTBAC framework is flexible and scalable; this makes the trust-based access control based on this framework more desirable. According to these authors, the fuzzy approach is easier to integrate into utility-based decision making and it is energy efficient. This is one of the few studies that has proposed a trust model that also deals with access control issues. However, the research did not deal with issues pertaining to trust formation. The issue of trust update was not addressed properly. Reputation was also not included in trust evaluation in this research.

Wang, Bin, Yu, & Nui (2013) also proposed a fuzzy approach to trust evaluation for the IoT environment (J. Wang et al., 2013). The research identified service, decision-making and self-organizing as attributes of trust management. The mechanism has 3 layers: sensor layer (devices e.g. RFID, WSN and base stations), the core layer (access network and internet) and the application layer (distributed networks e.g. P2P, Grid, Cloud computing, application system and interfaces). In this research, the IoT network is viewed as a service provider and the user as service requesters. This creates a bidirectional relationship which is highly affected by trust between the users and the IoT environment. The trust management model proposed in this research is composed of three steps: trust extraction, trust transmission, and trust decision-making. The model proposed by (J. Wang et al., 2013) applies layered trust based on formal semantics.

Liu, Chen, Xia, Lv, & Bu (2010) proposed a model that uses location-aware, identity-aware information and authentication history to evaluate the trustworthiness of the requested service (Y. Liu et al., 2010). The trust value ranks can either be low, high or medium. The model uses different authentication for each rank. Biometric information is required for low-rank values. The model uses a fuzzy approach to classify all the services that are provided in order to evaluate the sensitivity of the information to be transmitted. The model is centralized which might not be suitable for some IoT scenarios. The model is specifically for protecting IoT users' security. Another model that is based on the fuzzy technique was proposed by Chen et al (D. Chen et al., 2011). The trust metrics of the model are based on QoS and reputation. This model was designed for a Wireless Sensor Network (WSN) context.

Wen-Mao, Li-Hua, Bin-Xing, & Hong-Li (2012) proposed a hierarchical trust model for the IoT (Wen-Mao et al., 2012). The model can detect malicious organizations using the behaviour of their neighbouring nodes. Trust of the organizations is managed by long-term reputation. A Verifiable Caching Interaction Digest (VCID) scheme is introduced for the purposes of monitoring object-reader interaction in this research (Wen-Mao et al., 2012).

Saied, Olivereau, Zeghlache, & Laurent (2013) proposed a trust management model using a decentralized approach for the IoT for managing cooperation in the heterogeneous architecture (Saied et al., 2013). The model takes into consideration the capacities of the nodes. Trust in this model is defined as evaluated past behaviour and distinct cooperativeness using observations, experience, and second-hand information. The model has five phases:

- Gather information about the nodes
- Sets up a collaborative service with the requesting nodes
- Learns from its past operation by performing self-updates aimed at improving its future operations
- Assigns a quality recommendation score to each node after each interaction during the learning phase

Dong, Guan, Xue, & Wang (2012) designed an attack-resistant model for the distributed routing strategy in IoT. "Such a model can evaluate and propagate reputation in distributed routing systems and it is then proposed to establish reliable trust relations between self-organized nodes and defeat possible attacks in distributed routing systems" (Sicari et al., 2015).

Y. B. Liu, Gong, & Feng (2014) proposed a behaviour detection based trust system. The trust metrics for this model are calculated recommended trust and history statistical trust. The model uses the Bayes algorithm for the calculation (Y. B. Liu et al., 2014).

This section reviewed some of the contributions that have been made to trust management of IoT. There are still some issues that still need to be tackled. The following section highlights the outstanding issues in IoT Trust Management.

4.5 Outstanding issues in IoT Trust Management

The devices in the IoT environment will be generating large amounts of data; therefore, it is important to ensure that the data that is being provided is trustworthy. All the IoT 'things' rely on data which makes data collection and processing very important in IoT. It is important to achieve trust properties in IoT data process. In addition to this, the following issues are still open in IoT trust management:

- Trusted Third Parties to provide identity management – The IoT environment needs trusted third parties who can provide identification and authentication of IoT devices and users. Each user and each device in the IoT environment needs to be uniquely identified and the identity should be verifiable and unchangeable to prevent white-washing.
- Trust formation – Trust formation by a new device in the network has not been considered. The models that have been proposed evaluate trust based on recommendations and/or reputation with the assumption that the new device will have devices it already trusts in the network.
- Trust relationship evaluation, evolution and enhancement (Z. Yan et al., 2014) – there is still need to carry out research on evaluation, evolution, and enhancement of trust relationships in IoT.
- Trust during data collection, process, mining, and usage - The few studies that have considered data fusion and mining trust have not tested it in a practical environment.
- User privacy – some of the data that will be mined and processed will be personal information of users, therefore user privacy is important in the model.
- User-device interaction trust – research on user-device interaction trust still need to be considered since different devices and different users will collaborate in the IoT environment.
- Formal and suitable algorithms for trust computation.
- Trust initialization methods that suit different contexts in the IoT environments.

- Mistrust and Distrust – the proposed methods have not taken into consideration mistrust and distrust. Distrust is an important element of a trust management model which can be perceived as a partner of trust which is more complex than trust (Cofta, 2006).

Literature shows that little has been done in terms of privacy preservation in IoT (Suo, Wan, Zou, & Liu, 2012; Z. Yan et al., 2014). Dynamic reconfiguration of trust management models also needs to be addressed. In order to accomplish security goals in the IoT environment, there is a need to design and create new trust models specifically for the IoT environment which takes into consideration the above-highlighted issues. The following section lists the components that need to be considered in the trust models.

4.6 Components of an IoT Trust Management Model

The following components need to be considered in an IoT Trust Management Model:

- Trust Specification Language – for specifying and managing all the other components. This will allow the framework to be easily adapted to suit any context and environment.
- Trust Relationship – for adding, removing and analysing relationships.
- Trust Requirement Specification – for specifying the requirements based on the environment and context.
- Decision Making – makes communication decision based on trust value and other factors such as risk.
- Authentication – for authenticating devices before trust evaluation and decision making
- Authority Delegation – for scenarios where an IoT 'thing' need to delegate some of its tasks to other 'things' in the network.
- Evidence Collection and Analysis – this component will be responsible for collecting and analysing information on security, available resources, and competence which will be used in calculating the trust value.
- The trust value calculation should also take into consideration other factors such as risk, recommendation, and reputation.

4.7 Secure Multi-Party Computation

"*Secure multi-party computation (SMC) deals with the problem of secure computation among participants who are not trusted with each other, particularly with the preference of privacy-preserving computational geometry*" (Z. Yan et al., 2014). SMC will allow different parties to provide their own secret input during computations and receive their own results computed using their secret input. The parties must have access to their own results only. SMC can be categorized as (Du & Atallah, 2001):

- Privacy-Preserving Database Query
- Privacy-Preserving Scientific Computations
- Privacy-Preserving Intrusion Detection
- Privacy-Preserving Data Mining – this is very important for IoT stakeholders.

IoT trust management also needs to take SMC into consideration.

4.8 Conclusion

Traditional methods of managing trust such as lightweight cryptography, secure protocols and privacy assurance are not sufficient for the IoT environment. Also, current security solutions cannot fully provide a secure IoT environment. That can only be done through IoT trust management models. In this chapter, the progress of research on IoT trust management was reviewed and some of the outstanding issues that still need to be addressed were identified. Trust models can be designed to also cover security, privacy, identity management and access control. Despite the rapid grow of IoT research, there is limited research on IoT trust management. In social IoT, the assumption is that there will be social relationships among the devices based on the social relationships of their owners. This limits the trust models based on social IoT. For privacy preservation to be achieved in IoT, there is a need for the concerned stakeholders to come up with a standard policy which is centred on protecting the users. There is still a need for further research on IoT security problems until the IoT environment has developed into a mature stage. The research will take into consideration mistrust, untrust, distrust and undistrust. However, since distrust has not been thoroughly studied for trust management in the IoT environment, its effect was also investigated in this research. The trustee has the option to defect or betray the truster. A trust model will enable the trustee to be accountable for the action they take.

5

Computational Methods Review

5.1 Introduction

Trust brings in the element of uncertainty in any system that requires the collaboration of multiple entities. In order to compute trust values, a computational method that deals with uncertainty is required. There is still little work done on IoT trust computation (Guo, Chen, & Tsai, 2017). The issue of the effects of the computational method chosen in computing trust has not been tackled. This section discusses the most relevant computation methods that were considered for the proposed trust model.

5.2 Weighted Summation

Weighted summation is a type of Multi-Attribute Value Theory. According to (Kim & de Weck, 2005), weighted summation "*transforms multiple objectives into an aggregated scalar objective function by multiplying each objective function by a weighting factor and summing up all contributors*". Research which employed weighted summation for computing trust values for IoT trust models include (Bao, 2013; I. Chen, Guo, & Bao, 2014; Nitti et al., 2014; J. Wang et al., 2013)

Weighted summation is an easy computation method and it has been used in different fields which include engineering, environmental impact assessments, and risk assessment to measure the performance of different policies. It enables the performance of different objects to be evaluated based on multiple attributes. The attributes used in weighted summation are usually incomparable because they use different scale measurements. However, the use of weighted summation makes them comparable. A weighting factor is assigned to each attribute in order to prioritize them. This causes the weighted summation method to compensate bad criterion scores with good criterion scores.

Weighted summation reduces the amount of information by combining different information into a single metric. Such metrics can be used to compare different policies. The process of combining multiple metrics into a single metric is known as normalization and it can cause valuable information to be lost. The weighted summation equation has the following format:

$$J_{weighted\ Sum} = w_1 J_1 + w_2 J_2 + \cdots + w_m J_m$$

where w_i is the weighting factor (Kim & de Weck, 2005). Weighted summation evaluation follows the following steps:

- Alternative policy definition
- Criteria definition and selection
- Alternative score assessment
- Assigning of weighting factors to the criteria
- Evaluation of the alternative

The main challenge with weighed summation is selecting adequate weight factors. In weighted summation, weight factors are arbitrary because they are selected in an ad hoc manner. In trust management basing the weighting factor for recommendations on its trustworthiness score can cause inaccuracies because its score might have been affected by low resource rather than it being malicious (Saied et al 2013).

5.3 Bayesian Network

A Bayesian network can be defined as a network of relationships that uses statistical methods to represent probability relationships among a set of variables (Heckerman, 1995). Bayesian networks include two components: probability theory and graph theory. The graph theory enables the Bayesian network to exhibit conditional independence among variables. The probability theory gives numerical values (probabilities) to the variables based on the parents of the variables. The probabilities are given to the nodes on the degree of belief of the person who assigns them. Bayesian probability is the degree of belief in a specific event. The probabilities are related to statistical experiments. They give a numerical value of the frequent occurrence of a sample data in a population. The Bayesian probability is based on the Bayes rule:

$$p(h \backslash e) = \frac{p(e \backslash h) p(h)}{p(e)}$$

where p(h) is the prior probability of hypothesis h; p(e) is the prior probability of evidence h; p(h\e) is the probability of h given e and p(e\h) is the probability of e given h.

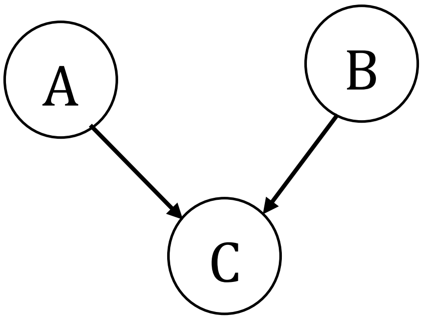


Figure 5-1: Simple Bayesian Network

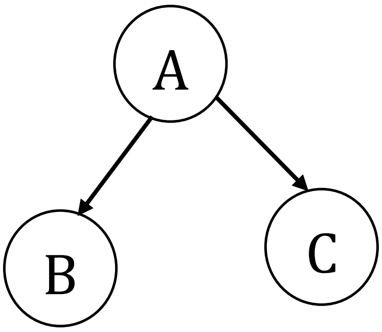


Figure 5-2: Simple Bayesian Network

Figure 5-1 shows a simple Bayesian network in which C is conditionally independent given both A and B. From Figure 5-1 the following equation can be derived:

$$P(A, B, C) = P(C\backslash A, B)P(A)P(B)$$

Figure 5-2 shows another form of a simple Bayesian Network. In Figure 5-2, both B and C are conditionally independent given A. This conditional independent gives the following equation:

$$P(A, B, C) = P(B\backslash A)P(C\backslash A)P(A)$$

In Figure 5-1, A and B are the root nodes and C is the leaf node. In a trust model, A and B can be the different aspects of trust towards a “thing” and C can be the trust value of a thing. A Bayesian network similar to Figure 5-2 can be used to estimate the reputation of a “thing” where A can be the weight of the reputation that a “thing” is providing and B and C can be the reputation values of different aspects that a thing is providing.

Bayesian networks are widely researched. They have been used for risk assessment in different fields including trust management. (Y Wang et al., 2006) used the Bayesian network to calculate reputation values and filter unfair ratings. Each dimension is calculated from a single Bayesian network (Firdhous et al., 2011). The prior probabilities used in the Bayesian network are obtained from expert knowledge and estimates from available data. In a trust model, the initial trust value is arbitrary and updated as reputation is built.

Most trust models based on the Bayesian network uses Probability Density Functions (PDFs) such as the Beta distribution to calculate trust values which include (Bao, 2013; Y Wang et al., 2006). PDFs show the real distribution of the aggregated trust values. (Y Wang et al., 2006) . The equation for the Beta PDF is as follows (Johnson, Kemp, & Kotz, 1999):

$$\mu = \frac{\alpha}{\alpha + \beta}$$

The advantages that Bayesian networks have over other computation methods include:

- enables the evaluation of trust using more than one dimension
- shows the real distribution of aggregated trust values (Y Wang et al., 2006).
- can be used to predict the outcome of actions taken
- enable prior knowledge and data to be combined (Heckerman, 1995)

Bayesian methods show a mathematical way of dealing with uncertainty through the use of priori probabilities. However, sometimes the priori probabilities cannot be found and they may also be difficult to define. Also, probabilities have the limitation that an object has to belong to one particular set. "*Probability is a measure of frequency of occurrence of an event, which has a physical event basis*" (Ponce-Cruz & Ramirez-Figueroa, 2010). Probability values are based on repeatable experiments and this is impossible to carry out when dealing with trust (Abdul-Rahman & Hailes, 2000). The properties of probability include transitive which is partial in trust. (Heckerman, 1995) raised the following questions concerning Bayesian definition of probability:

- why should the degree of trust satisfy rules of probability?
- On what scale should probabilities be measured?

In regard to trust, a follow up to these questions would be: What probabilities should be assigned to the beliefs that not at the extremes? (Alnasser & Sun, 2017) argued "*cannot be strictly treated with the likelihood of probability because the probability model contains an evaluation of uncertainty*" because it is vague and fuzzy.

5.4 Game Theory Based

Game theory is a branch of social science which uses mathematical and logical reasoning to determine steps which should be taken in order to get the best outcome. It is used in making a decision under competition based on at least two decision makers. Any situation that requires decision making in which the decisions are interdependent can be solved using game theory. Those situations are referred to as games and the decision makers are players. In the sense of game theory, a game is a situation that requires strategic decision making. Each player has two or more strategies. A strategy is a way of acting for a player. The strategies have defined possible outcomes called payoffs. A player is any decision-making entity such as a person or a committee. Game theory enables the players to take into consideration the purposefulness and intelligence of other players. In each game, there is a chance of cooperation and conflicting. Players can mix theirs moves so that they become unpredictable to other players. Strategic moves can also be used in game theory. In this case, a player can make threats and promises and he/she has the option of fulfilling those threats or promises. A player can also bargain by holding back their decision or move. Concealing and revealing of information can also be used to a player's advantage. Types of games in game theory include zero-sum games (only one player wins), positive sum (mutual gain) and negative sum (mutual harm) (Dixit & Nalebuff, 1993).

"*A (pure) strategy is a complete plan of action, specifying in advance what moves a player would make – what actions the player would perform – in every contingency that might arise*" (Colman, 2005). Each strategy in game theory takes into consideration all the other players' moves and strategies. Each player in game theory assumes that the other players are as clever as he/she is. Game theory takes into consideration the expected utility and common knowledge assumption (Colman, 2005). Expected utility is the expected payoff. This enables each player to anticipate the other players' strategies to a certain extent. In Game theory the standard common knowledge and rationality assumptions are (Colman, 2005):

- Common knowledge – this includes game specifications, rationality assumptions, strategies, and payoffs.
- Rationality – this is the expected utility theory. Players choose strategies that maximize their payoffs based on their beliefs and knowledge.

Game Theory is commonly explained using the prisoner's dilemma. The Prisoner's Dilemma consists of two prisoners, the prisoners are faced with the following strategies and with their corresponding payoffs:

- If both prisoners do not confess, they both get three years each

- In the case where both confess, then both receive a one-year sentence each
- In the case where only one confesses, the one who confessed will walk free with a reward and the one who didn't confess will receive a five-year sentence

In this situation, the prisoners have the option to either cooperate with each other (not confess) or defect (confess). If they cooperate, they both receive a good payoff, if they both defect, they will receive a moderate payoff and if one defects, the one who defected will get the best payoff while the one who didn't will receive the worst payoff. The payoffs of the prisoners are represented in Table 5-1, C represents Cooperate while D represents Defect. The payoffs are paired starting with Prisoner1's payoff followed by Prisoner 2's payoff respectfully.

Table 5-1 - Prisoner's Dilemma Payoffs

		Prisoner 2	
		C	*D*
Prisoner 1	*C*	3,3	5,0
	D	0,5	1,1

In the prisoner's dilemma, D is the best reply to all strategies which is known as the *dominant strategy* (Colman, 2005). A relational player is most likely to choose a dominant strategy over other choices that might be available.

Interdependent decisions are analysed with the goal of obtaining the optimal strategy that will result in the best payoff. Game theory considers two fundamental games: Sequential and Simultaneous (Dixit & Nalebuff, 1993). In sequential games, players' moves are made in sequence while players act at the same time in simultaneous games. There is a possibility of alteration of the players' move in sequential games and each player knows about all the previous moves of other players. The sequential game follows the "look ahead and reason back" rule (Dixit & Nalebuff, 1993). The rule follows three steps (Barron, 2013):

- Look ahead involves considering all decisions up to the last decision and assumes that all the other players will make decisions the maximizes their payoff.
- Reason Back involves backing up to the second-to-last decision and still assume that the other players will make decisions that maximize their payoffs.

- Repeat all these steps until all decisions are fixed.

The Prisoner's Dilemma is an example of a simultaneous game. A simultaneous game follows the following strategy:

- Use dominant strategy if it can be identified
- Else identify and eliminate any dominated strategy. A dominated strategy is a strategy that results in worst payoffs. Repeat this until a dominant strategy can be identified or until the game cannot be reduced.

The above steps might result in Nash Equilibrium when followed by all the players. In game theory, Nash Equilibrium (Equilibrium) "*is a set of outcomes such that no players have any incentive to change strategy*" (Barron, 2013). The process of determining equilibrium outcome is dynamic and is not clearly defined. An outcome is an equilibrium if "*there is no presumption that each person's privately best choice will lead to a collectively optimal result*" (Dixit & Nalebuff, 1993). Therefore, the Nash equilibrium is a list of strategies that result in the best payoff for each player. Some games have multiple equilibria and this makes it difficult for the players to make their decisions and others do not have the state of equilibrium (Dixit & Nalebuff, 1993). Game theory is also commonly used to model trust using different strategies. Game theory lacks considerations of societal aspects that affect trust and its approach is "*inherently confrontational*" (S. P. Marsh, 1994). Trust in most cases is used to prevent confrontations (S. P. Marsh, 1994). In his thesis, (S. P. Marsh, 1994) argue that game theory does not provide enough to handle trust.

5.5 Fuzzy Logic

Fuzzy logic imitates the human mind in reasoning by approximating values. Fuzzy set theory was designed to mathematically represent uncertainty and vagueness (Smith, 1994). It provides tools for imprecise problems. The concept of fuzzy logic includes fuzzy sets. Fuzzy is the degree to which an event occurs. According to (Zadeh, 1965), a fuzzy set is totally non-statistical. It uses linguistic variables and it can be used in vague systems. Fuzzy logic can be used in decision systems. In such a system, it provides a faster and simpler way for program development. Fuzzy system uses both fuzzy sets and fuzzy logic. Fuzzy logic formulates calculations on fuzzy sets.

Fuzzy logic is composed of three steps:

- Fuzzification: determines the membership value in overlapping sets. It is used to either increase the fuzziness of the set or turn a crisp set into a fuzzy set (Ponce-Cruz & Ramirez-Figueroa, 2010).
- Rules: these are input rules that will determine the output of a system. The rules are formulated from the knowledge of the system. The complication of the system determines the number of iterations required to find the required set of rules in order to produce a stable system. Fuzzy logic can be combined with Neural Networks. This may reduce the number of iterations needed to formulate rules. Combining fuzzy logic with Neural networks enables analysing of clusters of data.
- Combination/Defuzzification: this stage combines all the action and gives a single fuzzy action. This single action is an executable output in the system. Weighted sets may be used in the stage.

Probability and fuzziness exist together. Unlike probability, in fuzzy logic an object doesn't have to belong in a particular set, actually, an object may belong to multiple classes. Fuzzy logic focuses on the relevant properties of an object without restricting the object to a particular set. It assigns any real number in the interval [0,1] to each object to indicated the degree to which the object belongs to a particular class. In this interval 1 indicated that the object belongs to the set and 0 indicates that the object does not belong to the set. According to (Zadeh, 1965), a fuzzy set A in a universal set X is given by:

$$f_A(x) \ \in [0,1]$$

where x is a generic element of X. The degree of fuzziness of each fuzzy set varies. Fuzzy sets can be represented by membership functions which can either be summations or integrals depending on whether the universe of is discreet or continuous. Membership functions defines flexible membership of sets. For example, a fuzzy set from a discreet universe of discourse can be represented as follows:

$$A = \sum\nolimits_{x_i \in X} \mu_A(x_i)/x_i$$

And the continuous universe can be represented as:

$$\int_X \mu_A(x)/x$$

The measure of fuzziness E for a finite set A has properties:

- $E(A) = 0$: if A is a crisp set. In this kind of set, all the elements of A are known with certainty and A is not a fuzzy set.
- $E(A) = max$: if E has a maximum value, $f_A\ (x) = \frac{1}{2}\ \forall\ x\ \in X$.
- $E(A) \geq E(B)$: if B is more crisp than A, that is $f_B(x) \leq\ f_A\ (x)$ if $f_A\ (x) \leq \frac{1}{2}$ and $f_B(x) \geq\ f_A\ (x)$ if $f_A\ (x) > \frac{1}{2}$. This shows that fuzzy sets are relative.
- $E(A) = E(A')$: if A' is a complement of set A.

E can be defined by using classical information theory and entropy as follows:

$$E(A) = H(A) + H(A'), x\ \in X$$

$$H(A) = \ -k \sum_{i=1}^{n} f_A(x_i)\ In(f_A(x_i))$$

H is the entropy definition. Entropy is the degree of uncertainty in a system. A set is called properly fuzzy if $A \cap A' \neq \emptyset$ and $A \cup A' \neq X$.

Fuzziness and probability are two unique phenomena that that complement each other. The main difference between fuzzy logic and probability is that probability theory is additive. That is the probabilities must add up to one.

According to (Zadeh, 2003), probability theory and fuzzy logic are complementary. (Zadeh, 2003) argues that probability theory is based on fuzzy logic. In probability, all events are assumed to be binary in nature. this means an event can either occur or not occur. Probability and fuzzy logic both lie in the interval [0, 1]. Fuzzy logic estimates the *degree of truth* while probability estimates the *probability of truth*.

Fuzzy logic deals with partial degrees of truth and probability theory with crisp notions and propositions. The propositions in probability are estimated degrees of belief. The propositions are either true or false. Therefore, "*the probability of a proposition is the degree of belief on the truth of that proposition*" (Hajek, Godo, & Esteva, 1995). Fuzzy logic determines the degree of occurrence in fuzzy events while probability determines the frequency of occurrence in crisp events.

5.5.1 Fuzzy Inference System

Before a Fuzzy Inference System is developed, the following need to be specified (Abraham, 2001):

- Fuzzy Sets
- Fuzzy Operators
- Knowledge Base – the knowledge base can be expressed as fuzzy if-then rules.

FIS emulates the fuzzy reasoning of the human brain. The main characteristics of FIS are:

- Appropriate for uncertain and approximate reasoning
- Enable decision making in environments which have incomplete information

FIS lacks the ability to learn.

5.6 Neuro-Fuzzy

This section reviews the Neuro-Fuzzy method. Neuro-Fuzzy combines Fuzzy Logic and Neural Networks to produce a system which better results compared to using Fuzzy Logic.

5.6.1 Artificial Neural Networks

An Artificial Neural Network (ANN) is a system based on the biological neural system developed for data processing (Suparta & Alhasa, 2016). ANN emulates the functioning of the biological neural system. The processing is done by elements that make up the network. The function and operation of a Neural Network is determined by a neuron. The use of ANN enables (Sushmita & Sankar, 1996):

- Incorporating parallelism
- Handling optimization problems

An ANN system is constructed by defining its architecture and learning algorithm (Abraham, 2001). ANN focuses on the structure of the human brain. ANN are trained to perform specific tasks and this enables them to adapt. Adapting enables ANN to adjust their weights thereby optimizing their behaviour. The characteristics of a neuron include (Fullér, 1995):

- Output value
- Input connections
- Bias value
- Output connections

Each neuron has a weight which determines its effect on the network. Figure 5-3 shows an example of a neuron with n inputs where (X1, X2, X3,…Xn) ∈ X ⊂ $\mathbb{R}^n$.

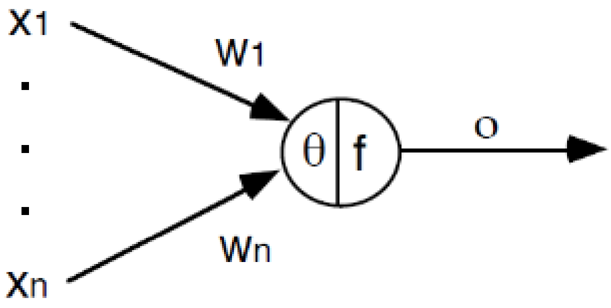


Figure 5-3: An example of a neuron

In the figure above the output is given the equation (Fullér, 1995):

$$o = f(< w, x >) = f(w^T x) = f\left(\sum_{j=1}^{n} w_j x_j)\right)$$

Where $w = (w_1, \ldots, w_n)^T \in \mathbb{R}^n$ is the weight and $f(w^T x)$ is the activation function. ANNs carries out parallel data processing. The ability of ANN to learn is based on modelling of the human brain. The knowledge in ANNs is stored as synaptic weights. The knowledge of an ANN is embedded in the whole network. This causes minor changes in the weight to produce unpredictable results.

5.6.2 Neuro-Fuzzy Systems

A neuro-fuzzy system is composed of fuzzy logic and a multi-layer neural network. Fuzzy Logic (FL) and Neural Networks (NN) complement each other. FL carries out high level reasoning and NN are a computational structure which enables the system to apply standard learning algorithms. the back-propagation algorithm can be used to enable a neuro-fuzzy system to learn. Neuro-Fuzzy system is used for the analysis of uncertain and imprecise information (Abraham, 2001). In a Neuro-Fuzzy system, NN provides learning ability to train the fuzzy inference system. Combining FL and NN has the potential to provide an intelligent system which is fault tolerant and adaptive with the ability to handle uncertainty in decision making problems (Sushmita & Sankar, 1996). FL systems emulate human reasoning when handling uncertainty and NN models emulate the human brain in processing information. Learning in NF systems fine-tunes the fuzzy inference system. According to (Nauck, 1997) neuro-fuzzy is the development of a fuzzy system that uses heuristic learning strategies from neural networks.

the types of Neuro-Fuzzy include:

- “A fuzzy rule-based model constructed using a supervised NN learning technique”
- “A fuzzy rule-based model constructed using reinforcement-based learning”
- “A fuzzy rule-based model constructed using NN to construct its fuzzy partition of the input space”

An NFS has the following advantages:

- Parallel computation
- Learning and generalization ability (Viharos & Kis, 2014)
- Contains a linguistic rule base
- Human-like knowledge representation
- Explanation abilities.

Learning can be in cooperated into a FS using a special ANN architecture (Abraham, 2001). In an NFS, the NN carries out the processing of the information while FL handles the reasoning. Training of NF enables it to develop rules and determine membership functions. An NF system has an input layer, hidden layers and an output layer. The hidden layer includes the membership functions and the fuzzy rules. Figure 5-4 shows an NF system with three hidden layers.

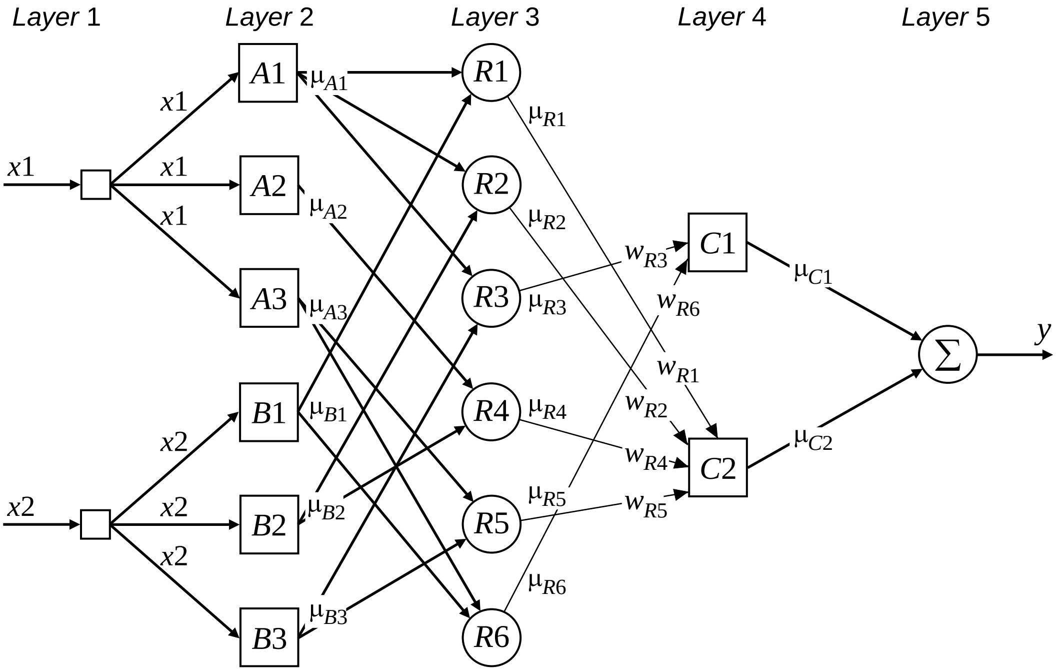


Figure 5-4: A Neuro-Fuzzy System with 5 layers

Developing neuro-fuzzy systems requires prior knowledge of fuzzy rules and fuzzy sets. Neuro-fuzzy systems are difficult to develop because the membership functions are found through trial and error (Nauck, 1997). The manual tuning of the system is time-consuming and prone to errors. Neuro-fuzzy can be categorized into the following categories:

- Cooperative – in this model ANs and fuzzy systems work independently of each other. The AN learns for the FS.
- Hybrid – the system is homogeneous. Fuzzy system is implemented as a special neural network. In this model, the rules, inputs, and outputs are neurons while the fuzzy sets are the weights. This eliminates the drawbacks of both fuzzy systems and neural networks.

5.7 Discussion

Trust is a human element that is determined by the human mind. Human decision processes do not follow the axioms of probability. Probability can either be frequency based, subjective or axiomatic. The human mind possesses a great ability to store and process imprecise and uncertain information. Uncertainty is caused by complexity, ignorance, and imprecision. Other factors that affect the degree of trust includes context, time and alternative options. Both fuzzy logic and probability address uncertainty. Both probability and fuzzy logic uses the same interval (0, 1). These methods are complementary and not competitive. Fuzzy logic studies what is different from probability theory.

Trust is complex and difficult to quantify. It consists of both subjective and objective knowledge. This and other highlighted problems of other computation methods led us to consider fuzzy logic for computing trust. A system that uses fuzzy logic is known as a fuzzy system. Fuzzy systems can handle both linguistic and numerical data. A fuzzy system utilizes fuzzy mathematics. Fuzzy logic systems can model and represent non-statistical uncertainty. “Non-statistical uncertainty is best represented with the concept of fuzziness where fuzzy logic is used to describe partial truth and approximate reasoning” (Meghdadi & Akbarzadeh-T, 2001). As humans, we quantify our feeling using linguistic variables such as “very angry or he can be trusted to a certain extend”. Fuzzy logic provides mathematical concepts that are suitable for translating linguistic variables to computational logic.

In fuzzy sets, the transition of membership is gradual rather than abrupt. Fuzzy logic's ability to represent linguistic data enables human expert knowledge to be incorporated through the use of if-then statements and membership functions.

5.8 Conclusion

Trust Models can enable IoT ‘things’ to collaborate in an effective way. In order to select a suitable computation for the trust model, a review of the most relevant ones was carried out in this chapter. The chapter reviews the most commonly used computational methods in trust management. The purpose of the review was to identify the best-suited method for the trust management model being proposed. Trust information is usually fuzzy and incomplete. The researcher is of the opinion that such information is best handled using fuzzy logic. Trust can be simplified greatly by fuzzy set logic because fuzzy logic is able to handle incomplete and uncertain information. Fuzzy Inference Systems are also computationally efficient which makes them more desirable for IoT.

6

Proposed Trust Model Design

6.1 Introduction

Trust is belief that the trustee will not work against the truster's goals where uncertainty and vulnerability are present. In an IoT environment, trust can be evaluated through a trust management system (TMS). It is essential for a TMS to fulfil the requirements of the network security as well as the collaboration objectives of all the entities involved. A TMS suitable for the IoT should support interoperability since the success of the IoT predominantly lies in the interoperability of the environment. The TMS also needs to be adaptive since the environment is dynamic with 'things' constantly leaving and joining the network. As such, reliable cooperation among trust management techniques in all the layers is a requirement in IoT trust management (Z. Yan et al., 2014).

Since trust is a social phenomenon, trust modelling in a virtual world should be based on how it functions in a social environment. Socially, trust develops gradually over time. However, in an IoT environment, it might not possible to have sufficient time to develop trust this way. This situation is also experienced in the social environment; therefore, the proposed model seeks to extend the same strategies used in the social environment in the digital environment.

Due to limited resources and limited computational capability, some of the IoT 'things' are not capable of computing and storing their own trust values. Such devices benefit from an Agent-Based Trust Management System (ATMS). An A TMS will immensely benefit the IoT environment. However, the system comes with its own disadvantages which include communication overheads and malicious agents.

An IoT trust model should be customizable in order to meet the requirements of the heterogeneous IoT environment. The IoT environment requires a model that can be integrated into different entities. Both of these requirements for the IoT are vital and were considered in the design of the model. The proposed trust model uses the fuzzy model to calculate trust from the gathered information. This chapter details the design of the proposed model.

6.2 Network Description

Trust evaluation is difficult in the IoT because of the heterogeneity of the environment and the context of the collaboration among the 'things'. The IoT 'things' includes but is not limited to home automation devices, medical devices, phones, computers and servers. Figure 6-1 shows some of the devices that can be found in the IoT network.

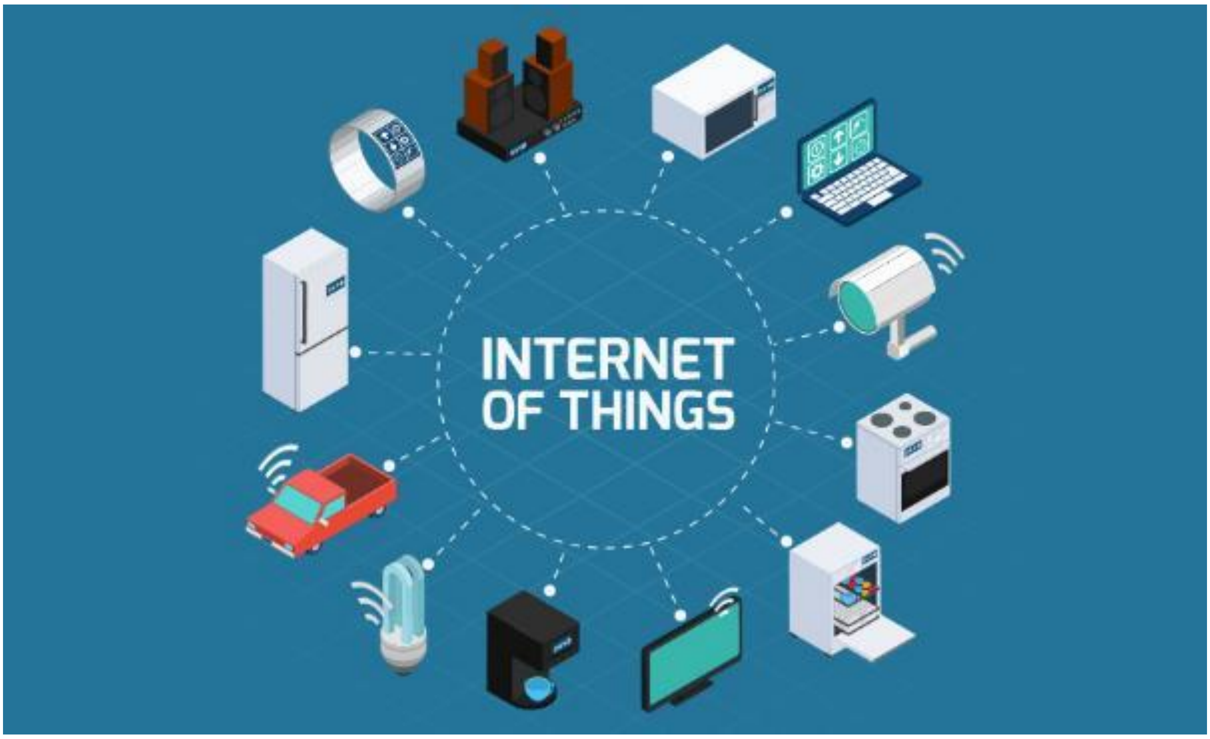


Figure 6-1: Examples of some of IoT Devices

The IoT network includes Wireless Sensor Networks (WSN) as well. WSN includes base nodes which have more computational capability, energy and communication resources compared to the regular nodes. The base node can connect to the internet using either the basic mode or the advanced mode. The basic mode is for basic gateway service while advanced mode has additional functionalities which include data analysis (Elkhodr, Shahrestani, & Cheung, 2016). The base node can be responsible for handling security issues in the sensor network. This frees up the regular nodes to focus on their tasks in the network. This kind of network was also considered in the development of the trust model. Types of attacks that needs to be combated by a TMS include but are not limited to:

- Bad mouthing
- Selective behaviour attack
- Ballot stuffing attack

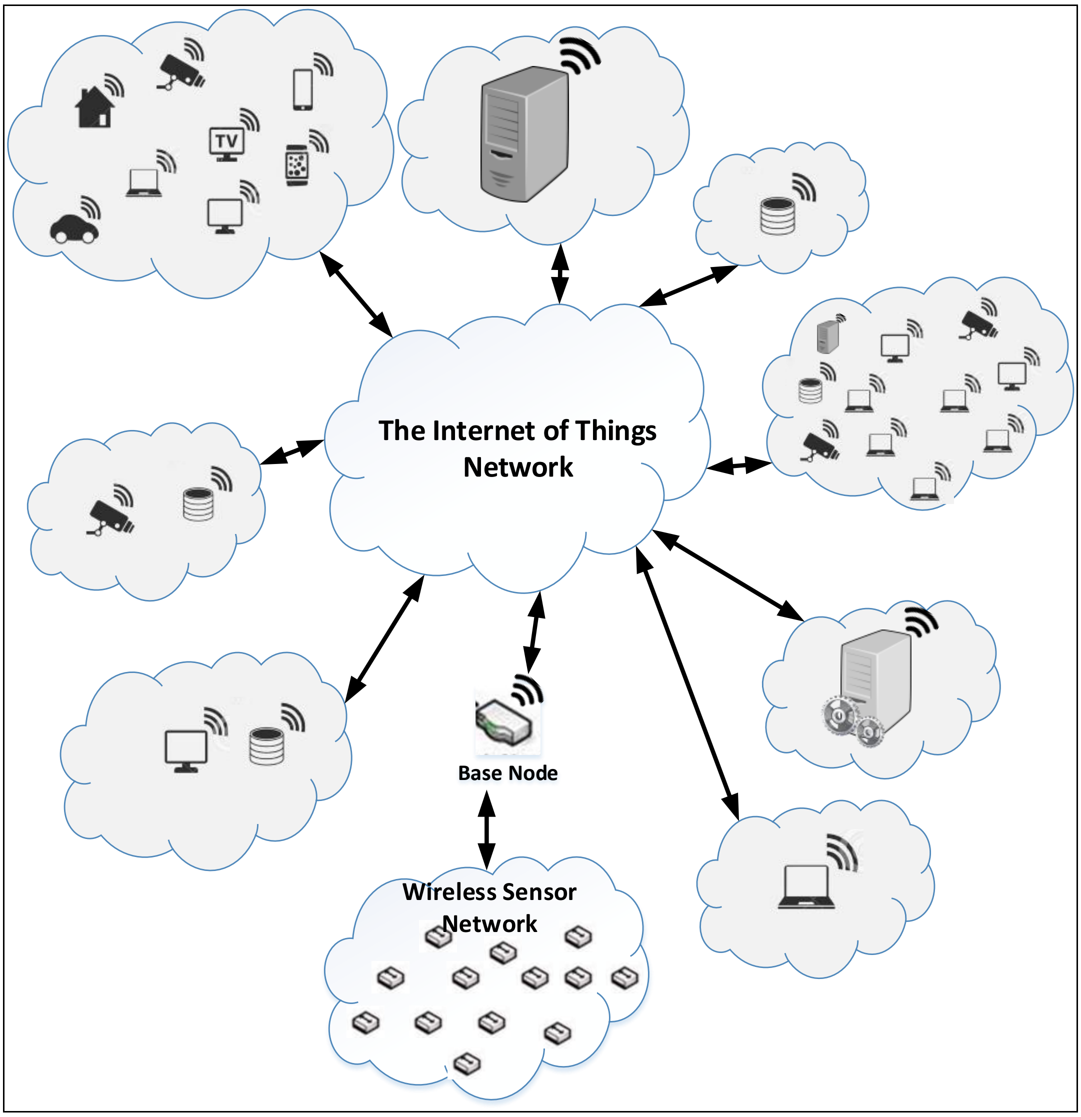


Figure 6-2: IoT Environment Example

Figure 6-2 shows an example of an IoT environment. The IoT environment consists of different devices with varying resource capabilities. In the IoT environment, devices may be associated based on associations such as:

- Ownership
- Network Service provider
- Location

- Common interest
- Service provider
- Manufacturer

Such associations may be automatic as the entity joins the network. Some of these associations have been used to create a trust relationship such as Social IoT (SIoT), ownership and community of interest. As highlighted in Chapter 3, the research is of the view that the IoT relationship is not limited to these relationships. In Section 6.6.3, new ways of defining relationships are explained.

6.3 Definitions of Key Terms

In the proposed model, the following terms will be defined as follows:

- **Trust** is "the firm belief in the competence of an entity to act dependably, securely, and reliably within a specified context" (Grandison & Sloman, 2003).
- **Distrust** is a belief that a trustee will work against the interests of the truster in a specific context (S. Marsh & Dibben, 2005).
- **Mistrust** is when a truster trusts an untrustworthy trustee or when a truster does not trust a trustworthy trustee.
- **Untrust** is trust that is not enough for cooperation.
- **Undistrust** is negative trust when there is no sufficient information to make a conclusion to distrust (Griffiths et al., 2006).
- **Ignorance** is when there is a lack of evidence to either trust or distrust.
- **Reputation value** is a value computed from experiences of multiple "things" in the environment. Reputation value may be used to evaluate trust in an attempt to predict its behaviour based on past experiences.
- **Trust value** is the opinion of the truster based on their own knowledge and experiences together with the reputation value

6.4 Assumptions

The proposed model is a theoretical model which will be tested in a simulated environment. In designing the model, the following assumptions were made:

- Initial authentication of a new 'thing' is handled by a Trusted Third Party (TTP)
- Each IoT 'thing' is uniquely identified.
- 'Things' cannot change their identity.

- ‘Things’ with computational capability will have their own opinion about the trustworthiness of some of the ‘things’ they are interested in.
- Some of the ‘things’ in the IoT network have limited memory to store trust information.
- The trustee has a choice of refusing to cooperate with the truster if the trustee deems the truster untrustworthy.

6.5 Questions to Consider in Decision Making

1. What are the trust requirements for an IoT environment?
2. Does the trustee have enough resources?
3. Is the trustee willing to collaborate with the truster?
4. Does the trustee trust the truster?
    a. If the trustee does not trust the truster, how does that affect the truster’s trust value for the trustee?
5. Is there a need to factor in the time element in computing trust in an IoT environment?
6. How can contradicting recommendations be handled?
7. Is it possible for a single IoT model to meet the trust requirements of an IoT environment?

6.6 Proposed Trust Model

Trust models attempt to restrict or encourage interaction among entities, depending on the behaviour of the entity. Trust management improves security in an environment by collecting and combining the necessary information and compute the trustworthiness value of each entity. Although it is not always possible to predict the future accurately, the past behaviour may be used to estimate the future actions of the entity. This section describes the trust model that was proposed for the IoT environment in this research. The proposed model is a multi-dimensional hierarchical trust model which takes into consideration different properties of trust based on human trust. Humans take into consideration the following factors when making trust decisions:

- Information known
- Ability
- Available options and consequences

These factors affect how trust is modelled among humans and they also need to be included when computing trust in a digital environment. Therefore, trust computation in the model will include knowledge, competence and risk as some of the trust properties. The decision to trust will be based on evidence, confidence, or good intentions of the trustee. Betrayal gradually reduces trust. As mentioned in Section 6.4, for this research, it was assumed that authentication is handled by a TTP. However, the proposed trust model will validate the authentication before allowing the new 'thing' to join it trusted network.

6.6.1 Trust Model Architecture

After taking into consideration the scalability of the trust model and resource constrained 'things' in the IoT environment, the proposed model is a distributed hierarchical trust model. This will enable the model to remain scalable while providing trust related service to all the 'things' in the IoT environment. Taking into consideration all 'things' in the IoT, it surfaces that some of these entities are not capable of both computing and storing trust information. In order to provide a trust model that caters for all the 'things' in the network, it is recommended that trust agents be used. The research proposes that the trust model is made up of distributed specialised trust agents whose only purpose in the network is to manage trust. The specialised trust agent will create a trust network amongst themselves. Since the model is distributed, it follows that the model uses a hierarchical decentralized model.

Trust agents are ideal for the IoT environment because agents can be created to be rational and intelligent. According to (S. P. Marsh, 1994), the use of a rational agent in trust management is advantageous because an agent is:

- Intelligent
- Rational
- Cooperative
- Geographically distributed
- Independent

Apart from the advantages mentioned above, agents also seem to be the best option for modelling trust because they are designed to act on behalf of humans. Agents possess the ability to reason. Figure 6-3 shows the architecture of the proposed model. The proposed model consists of multiple root agents. Each of the root agents can be accessed by any trust agent in the network. This prevents failure of the trust network when one of the root agents fails. The trust value of each root agent and trust agent is maintained by other root agents and trust agents. This enables the model to identify malicious and selfish trust agents. Once a trust agent or a root agent is identified as malicious, it is immediately excluded from the trust network.

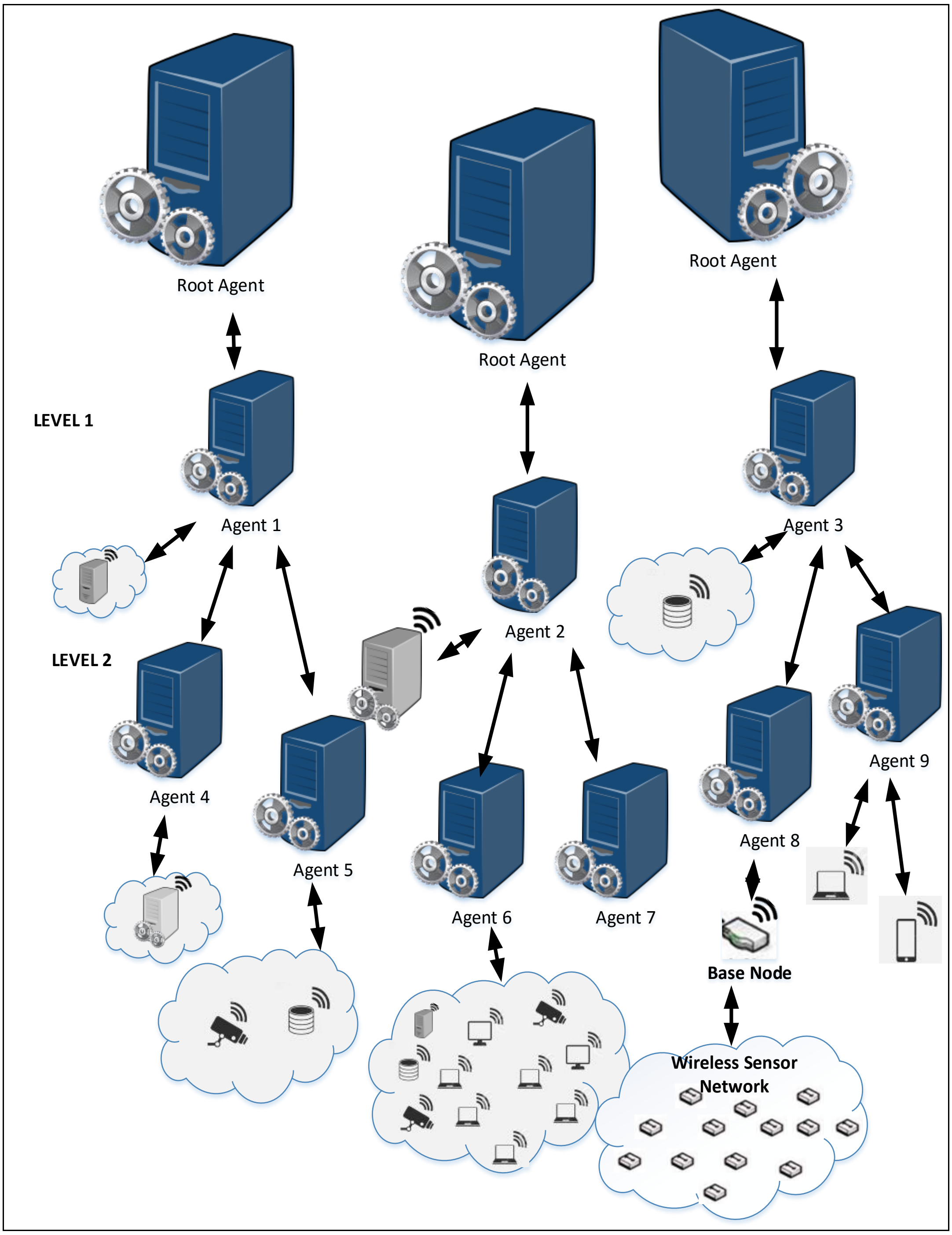


Figure 6-3: Proposed Trust Model Architecture

The root agents have the following responsibilities:

- Keep record of all trust agents and other root agents in the network.
- Evaluation of trust values of both trust agents and root agents.
- Keep records of all the identities and resources of the entities associated with each trust agent.

The root agents are not involved in any trust computation for entities. They are responsible for calculating trust values for trust agents as well as for each other. Trust agents have the following roles:

- Keep records of all entities in its domain
- Keep trust values of all entities in its domain as well as entities in other domains that have cooperated with any of the entities in their domain
- Evaluates trust value upon request from an entity
- Keep records and trust values of all trust agents and root agents it's interested in
- Collect and analyse trust data in its domain
- Validation of trust on entities using the root agent
- Keep track of the resources that each of the entities in its domain has and provide ratings for them as per request of the truster.
- Evaluate network activities to identify malicious entities and malicious trust agents

The trust agents will also contain information about other trust agents in the network they have interacted with and trust. This means ultimately trust agents will end up creating trust networks amongst themselves. Whenever a trust agent becomes untrustworthy, the information will be propagated by the root agent to all the other root agents as well as the trust agents. In the trust model, historical evidence is only kept by the trust agent and root agents. The watchdog scheme can be used but trust agents to observe the behaviour of the entities in their domain. This is done at a local level depending on the traffic of the network.

Instead of having multiple specific trust management models for the different IoT environment, trust agent can be tailored to suit the environment they are managing.

6.6.2 Components of the Proposed Framework

The trust model will consist mainly of five components: evidence collection and analysis module, trust requirement specification module, trust relationship manager, trust computation module and decision making and authority delegation module. Apart from the root agents and the trust agents, the other roles in the trust model include:

- Trustee
- Truster
- Recommender

Figure 6-4 shows the proposed trust model for the root agents and trust agents. A trust specification language allows different entities in the network to communicate and exchange trust information and trust data. Trust specification language and authentication were not included in the implementation of the model.

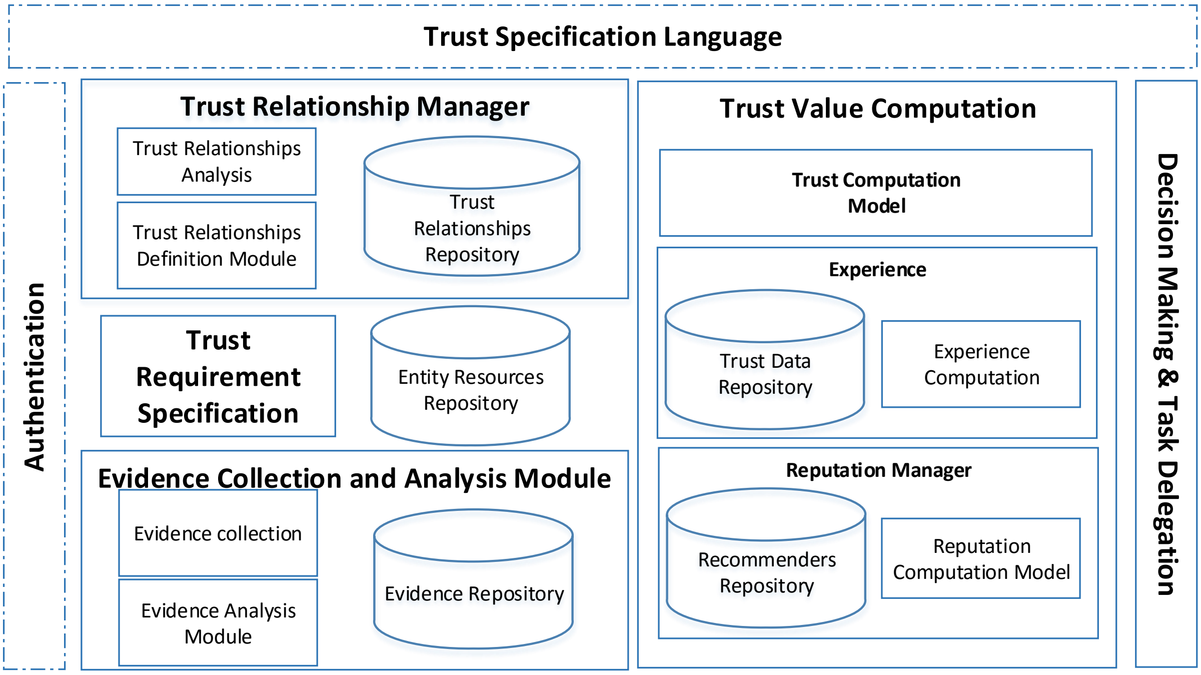

Trust Specification Language
Authentication
Trust Relationship Manager
Trust Relationships Analysis
Trust Relationships Definition Module
Trust Relationships Repository
Trust Requirement Specification
Entity Resources Repository
Evidence Collection and Analysis Module
Evidence collection
Evidence Analysis Module
Evidence Repository
Trust Value Computation
Trust Computation Model
Experience
Trust Data Repository
Experience Computation
Reputation Manager
Recommenders Repository
Reputation Computation Model
Decision Making & Task Delegation

Figure 6-4: Proposed trust model for trust agents

In order to cater for entities with limited resources, a different model was proposed for entities in the network as shown in Figure 6-5.

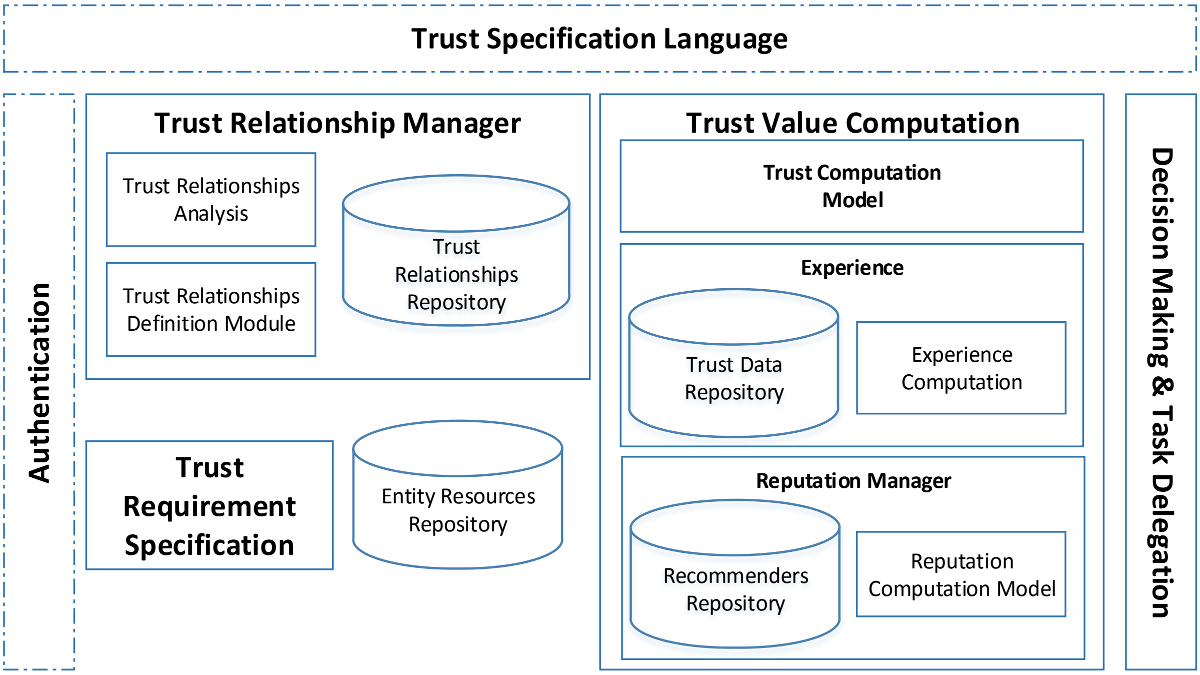

Trust Specification Language
Authentication
Trust Relationship Manager
Trust Relationships Analysis
Trust Relationships Definition Module
Trust Relationships Repository
Trust Requirement Specification
Entity Resources Repository
Trust Value Computation
Trust Computation Model
Experience
Trust Data Repository
Experience Computation
Reputation Manager
Recommenders Repository
Reputation Computation Model
Decision Making & Task Delegation

Figure 6-5: Proposed model for entities

Before any entity can join any domain network, its authentication is validated by one of the root agents. The trust model for the entities includes the following repositories:

- Trust data
- Recommenders' information
- Trust relationships
- Trusted trust agent(s) and root agent(s)

The components of the trust model have the following responsibilities:

- **Evidence Collection and Analysis Module:** The component periodically collects and analyses trust data. In root agents, the component is also responsible for validating and authenticating all entities before they join the trust network. This component is also responsible for initializing the trust of all 'things' as they join the environment. It is responsible for optimizing trust values in the network. The evidence collected includes information about the data, services and resources that the entities provide.
- **Trust Requirement Specification Module:** The module enables the entities to specify their trust requirement. The component consists of a repository that consists of all the resources and requirements available from the available entities.

- **Trust Relationship Manager:** The module contains defined relationships. The module enables the addition of new relationships definition. This will enable the environment to adapt to the ever-changing IoT environment.
- **Trust Computation Module:** this component will be responsible for evaluating the trust value based on direct and indirect observation.
  - **Reputation Evaluation Component**
    - Direct observation: Direct observation is the first-hand experience that the truster obtained while interacting with the trustee.
    - Indirect observation: Indirect observation is the recommendation obtained from either other entities or from other trust agents.
- **Decision Making and Authority Delegation Module:** This component is responsible for making the decision to either communicate or collaborate with an entity based on the trust value and the risk involved in the transaction.
- **Resource Availability Register:** The resource availability register keeps records of the resources that each entity has. This component is also responsible for validating the available resources for each entity in the trust agent's domain.
- **Trust Update Entity –** This component is responsible for updating trust values after each interaction. It also updates trust based on trust decay over time.

The deployment of the trust agent depends on the resources available on the entity. Constrained devices will depend on the actual trust agents for trust information. Figure 6-6 shows the interaction of different modules of the trust models.

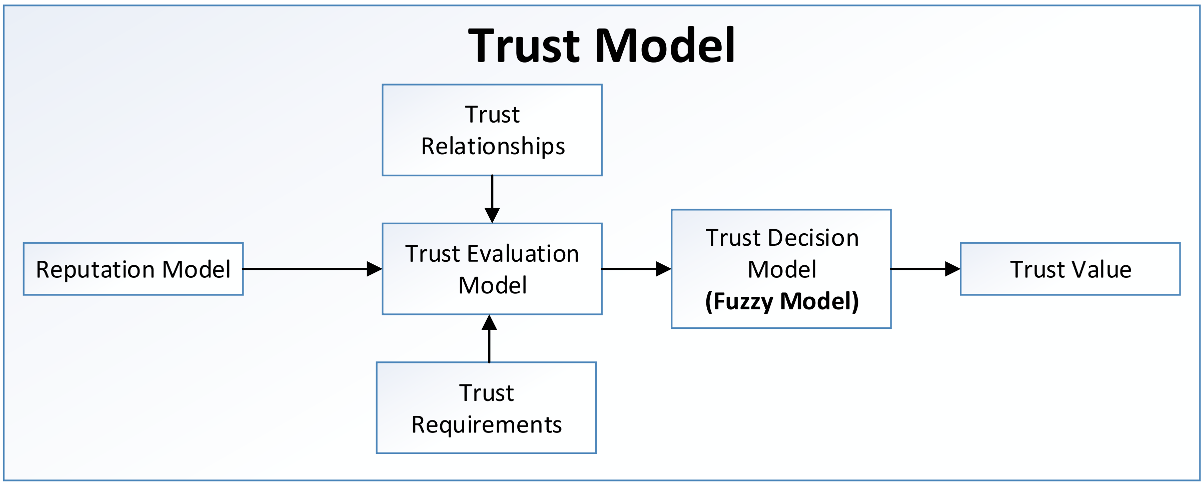


Figure 6-6: Relationships among trust entities

6.6.3 Trust Relationships

Trust relationships give meaning to trust (Chang et al., 2005). As highlighted in Chapter 3, trust direction can either be one-way or two-way. In one-way trust, a trustee trusts the truster but the truster does not trust the trustee. Two-way trust is when the trustee trusts the truster and the truster trusts the trustee. The two-way trust does not imply symmetry because the truster and the trustee might have different trust values for each other. A one-way trust may develop into a two-way trust as trust continues to build up. Two-way trust is the basis of a trust relationship. In a trust relationship, the relationship defines the basis of trust and the trust value define the strength of the relationship. (Chang et al., 2005) stated that trust relationships are determined by context and a timeslot.

Trust can be evaluated based on direct interactions, reputation and recommendations. Prior to having direct interactions, reputation information and recommendations, trust can be evaluated based on trust relationships. Trust relationships in IoT have been established using Social IoT (SIoT). However, the use of SIoT assumes that trust is transitive, which is not always the case. Also in SIoT, the assumption is that there exists a social relationship among the owners of the entities which is not always the case. The main issue in IoT trust relationships is the initiation of the relationship. Trust relationship can be initialized through direct interaction, reputation and history (Chang et al., 2005). Currently, in addition to SIoT, trust relationships can be created based on:

- Location
- Community
- Recommendation
- Identity

In this research, in addition to the above, the researcher also propose that a trust relationship can be created through the use of advertisement. In this case, the researcher proposes that entities can advertise themselves. After an entity has advertised itself, trust agent will evaluate its trustworthiness. If the entity is found to be trustworthy, the trust agent will then recommend the entity to entities in their own domains. In the advert, an entity can also provide references which can be used by the trust agent to obtain reputation information. In this case, the trust agent can penalise any referee for providing false information. Trust relationship is also factored in the computation of trust as a trust property. Trust relationships may also be created based on the service(s) that the entities consume. However, when creating relationships, deception and distrust should be taken into consideration. The researcher also proposes that ratings can be used as a way of creating trust relationships. When an entity joins the network, a rating for it needs to be provided after each communication through the reputation component. The information provided by the reputation component can be used to define the type of relationship the entity can be involved in. (De Oliveira Albuquerque, García-Villalba, & Kim, 2014) proposed a trust model that enables the calculation of trust values for trust groups. They defined a trust group as "*a collection of entities connected together with common goals or even common contexts*". Trust values can be beneficiary in IoT. Community of Interest (CoI) can be used to create a trust group. In this case, a trust group is a collection of entities that have a common interest and capabilities.

6.6.4 Trust Computation

Chapter 4 discussed different computation methods that have been used to compute trust and highlighted that fuzzy will be used in this research. Fuzzy has demonstrated its power in many applications. In 2001 (Meghdadi & Akbarzadeh-T, 2001) proposed the use of probabilistic fuzzy logic for modelling complex non-deterministic systems that deals with both non-statistical and statistical uncertainties. Fuzzy logic determines the degree of occurrence in fuzzy events while probability determines the frequency of occurrence in crisp events. This makes probability unsuitable for trust computation. Trust is the degree of belief; therefore, it makes sense to employ fuzzy logic in the computation method. Probability models uncertainty events while fuzzy logic focuses on the occurrence of an event to a certain degree. Fuzzy logic measures the degree to which an event occurs, not whether it occurs. "Fuzzy logic provides a natural framework to deal with uncertainty and the tolerance of imprecise data inputs to fuzzy-based systems makes fuzzy reasoning especially attractive for the subjective tasks of trust evaluation, business-interaction review and credibility adjustment" (Schmidt, Steele, Dillon, & Chang, 2007).

This research proposes a fuzzy based framework because trust is a fuzzy probability. Also, since trust is a degree of belief, it is fuzzy. Hence it cannot be estimated using probability theory because probability theory lacks methods that estimate fuzzy probabilities. A question like "Can A be trusted?" cannot be quantified using probability theory.

6.6.4.1 Computational Issues to Consider

1. How can IoT things model trust?
2. How is trust evaluated after each interaction?
3. How is trust updated after each interaction and as time elapses?
4. How are recommendations obtained?
5. How are recommendations combined?
6. How can direct experience and recommendations be combined?
7. How can recommendations be used to update trust?

6.6.4.2 Situations Considered

There is a problem in trust where an individual A is faced with the decision of deciding whether an agent is being truthful, based on the fact that if the agent knows something about itself in the IoT environment or something about another agent which agent A does not know, will the agent reveal it? How can this affect the knowing agent's trust value if the deciding agent finally knows the truth?

6.6.4.3 Computation Description

Trust can be divided into two constructs: trusting intention (the willingness of one to trust in another) and trusting beliefs (the belief that someone is trustworthy). Both of these trust constructs can be estimated using computation methods in a digital environment. In developing trust models, it is important to implement algorithms and computation methods that imitate human reasoning. Any form of human computation is based on approximations and uncertainties; this is best modelled using fuzzy logic. Fuzzy logic can bridge the gap between computational logic and human reasoning (Schmidt et al., 2007). These are some of the reason why fuzzy reasoning was chosen for this research. Trust computation is crucial in the IoT environment. Building a trust-based network helps improve security in communication and interaction. The trust computation module is responsible for calculating and estimating trust values. Trust value calculation specifies how entities calculate reputation value of other entities which are participating in the network. Both the truster and the trustee have the potential to be malicious. This section describes the computation of trust values in detail.

6.6.4.3.1 Trust Initialization

At the initial stage, trust is generated through identification and also through the information gathered during communication with the new entity. A new entity's or trust agent's trust value is set to either zero or 0.5 depending on the knowledge that is available about the new entity or agent. As the entity continues to participate in the network, it's trust value will either go up or go down depending on its behaviour. The trust model uses continuous trust values.

6.6.4.3.2 Trust Composition

Emulating patterns of human behaviour enables the trust model to deduce trustworthiness. Patterns of human behaviour include trust properties such as integrity, relationships, and consistency. That makes the translation of trust into the digital format a complex and rigorous task. Trust composition outlines the components considered in trust computation (Guo et al., 2017). The trust model proposed in this research is multi-dimensional which enables entities to select and specify the trust properties that they need. Different dimensions are split into measurable categories and sub-categories. This makes some of the trust properties optional in trust calculation. All the values for the trust properties are in the range $[-1, 1]$. An object can have different values for different contexts for the same entity and same properties. The trust properties considered in trust computation depend on the policies, context, and requirements of the truster. The properties that make other properties can be weighed according to the requirements of the truster.

As mentioned in Chapter 3, the following trust properties were identified for IoT and are defined in the proposed model as follows:

- **Competence** - the ability to carry out a transaction successfully and efficiently. This property takes into consideration the resources available. Competence is a pre-request in all trust calculation. In terms of resource availability, the following should be taken into consideration: bandwidth, availability, and latency. Competence is indicated by $Comp_{ij}^{context}$. Where i is the truster, j is the trustee.
- **Availability** – the ability of the trustee to perform the requested task in the specified time. In the model, availability is denoted by A_{ij}. Where i is the truster, j is the trustee.

- **Quality of Service (QoS)** – the description of the quality of the overall performance of all the transactions carried out by the trustee in a specific context. (Nitti et al., 2014) measured QoS using transaction performance. In addition to transaction performance; competence, cooperativeness, reliability and task competition are some of the properties that can be used to measure QoS of trust (Guo et al., 2017). QoS can be computed as follows:

  $$QoS_{ij}^{Context} = \sum_{k=1}^{n} W_k P_K$$

  Where i is the truster, j is the trustee, W_k is the weight of the property and $P_k$is the trust value of the property.
- **Consistency** - the quality of achieving a level of performance taking into consideration competence, QoS, and reliability. Consistency is denoted by $Con_{ij}^{Context}$ in the model. Where i is the truster, j is the trustee.
- **Predictability** - the ability to be predicted in terms of QoS and consistency. All the activities, functions and services of the trustee must be known. The activities of the entities in the network can be monitored from time to time by a trust agent. Any attempt by an entity to secretly manipulate data or another entity results in the loss of trust. Predictability is denoted by $P_{ij}^{Context}$. Where i is the truster, j is the trustee.
- **Relationship:**
  - **Centrality:** The concept of centrality comes from social networks. It measures that attraction that an entity has based on the number of entities it interacts within the network. An entity many relationships is considered to have a central role. Centrality may be computed as proposed by (Duan, Gao, Foh, & Zhang, 2013; Gangal, Narwekar, Ravindran, & Narayanam, 2016). Centrality is denoted by $Cent_{ij}^{Context}$. Where i is the truster, j is the trustee.
  - **Community of Interest (CoI)** – a community of a shared common interest among entities. In the model $CoI_{ij}^{context}$ denotes CoI. Where i is the truster, j is the trustee.
  - **Reputation-Based** – the reputation value of an entity can be used to create a relationship if it exceeds a certain threshold. In the model $Rep_{ij}^{context}$

denotes a reputation based relationship. Where i is the truster, j is the trustee.

- **Reliability –** the overall trust value of an entity taking into consideration all trust values of all the property that the trustee keeps for a specific purpose. Reliability considers the stability of the connection as well as fault tolerance. (Xia, Jia, Ju, Li, & Zhu, 2011) proposed a method for calculating which can also be used in IoT to evaluate the reliability of the path. $Rel_{ij}^{context}$ denotes reliability in the proposed model. Where i is the truster, j is the trustee.
- **Risk** – the risk involved in the transaction. This takes into consideration the security of both the entity and the network. Security objectives based on the CIA model include confidentiality, integrity and availability. In dealing with sensitive data, privacy also becomes an issue that needs to be addressed. Privacy issues include data confidentiality. Security also includes the physical security of the entity as well as access control. Risk can be evaluated in IoT as proposed by (Duan et al., 2013). Risk is denoted by $Ris_{ij}^{context}$ in the model. Where i is the truster, j is the trustee. Risk can be identified and estimated using policies. Risks have to be taken in order to build trust (Swinth, 1967).
- **Commitment** – the quality of being dedicated. Commitment is denoted by $Comm_{ij}^{context}$ in the model. Commitment can be determined through relationships. Where i is the truster, j is the trustee.
- **Willingness** – the perceived preparedness of the trustee to collaborate or communicate with the truster. A trustee may send a request to check for the willingness of the trustee. Resources of the trustee may be considered in determining the willingness. Willingness is denoted by $Will_{ij}^{context}$ in the model. Where i is the truster, j is the trustee. Cost can be used in requesting for the willingness of entity. A trustee that rejects collaboration of communication for a high cost activity that will strain its resources is justified and should not be rated as malicious. Since trust is not symmetrical, it is important for willingness to take into consideration the trust that the trustee has in the truster. If the trustee does not trust the truster, it might not be willing to communicate or collaborate with the truster. Willingness also depends on the trustee's available resources.

- **Motive**: belief that the trustee is motivated to interact with the truster. An entity that is new in the network might be motivated to interact in order to gain the trust of other entities in the network. A malicious entity is motivated to collaborate with entities that will only benefit its cause.
    - **Reciprocate** – the act of the trustee responding to an action done by the truster in previous interactions. Reciprocate is denoted by $Reci_{ij}^{context}$ in the model. Where i is the truster, j is the trustee. Reciprocity may be a response to favour or revenge.
    - **Incentive** – a motive from the truster that encourages the trustee to collaborate or communicate. This includes the profits that the trustee may gain from the interaction. Incentive is denoted by $Ince_{ij}^{context}$ in the model. Where i is the truster, j is the trustee.
- **Persistence**: belief that the trustee will not change its decision to help the truster after making a commitment. Persistence is denoted by $Pers_{ij}^{context}$ in the model. Persistence maybe be computed as follows:

$$Pers_{ij}^{context} = 1 - \frac{n(c)}{n(tl)}$$

  Where $n(c)$ is the total number of all cancelled tasks and $n(tl)$ is the total number of tasks that the trustee has been requested to perform in the network, i is the truster, j is the trustee.
- **Defect** – the shortcomings of the trustee. These include failing to complete tasks and dishonest experiences that the truster has of the trustee. The trust property value for defect is calculated as follows:

$$\boldsymbol{Def_{ij}^{Context}} = \begin{cases} \displaystyle\sum_{k=1}^{n} \boldsymbol{W_k p_k} \\ \\ \boldsymbol{Def_{ij}^{context} + e^t} \end{cases}$$

  Any discrepancy in recommendations is also taken into consideration. If the recommender provided false information its defect property value will be updated.
- **Transaction** – the total number of transactions between the truster and the trustee. This property may be used to detect suspiciously high interaction. It also

helps in determining the validity of the recommendation. Transaction is denoted by Tra_{ij}.

- **Encounter** – direct interaction between the truster or the trustee or between the recommender and the trustee. This is the total number of times that the trustee and the truster have encountered each other directly. Let n be the number of times that the truster has encountered the trustee and v be the outcome of the encounter. For each encounter, $v = -1$ if the experience from the encounter is negative and $v = 1$ if the experience of the encounter is positive. If the trust value increases or remains constant, the experience is positive. However, if the trust value decreases, the experience is negative. The trust property value for encounter is calculated as follows:

$$Enc_{ij}^{context} = \frac{\sum_{k=1}^{n} v_k}{\sum_{k=1}^{n} |v_k|}$$

- **Belief** – confidence that the truster has in the trustee or the confidence that the recommender has in the trustee. Belief is based on encounter, knowledge, and reputation of the trustee. Reputation includes both direct and indirect observations. Belief is denoted by $Bel_{ij}^{context}$ in the model.
- **Behaviour** – patterns of interactions. Compares the variation of the positive and negative experiences. Behaviour is denoted by $Beh_{ij}^{context}$.
- **Cooperative** - measures the level of cooperation between the truster and the trustee. It can be evaluated based on trust relationship and behaviour. Cooperative can be computed using formulas such as those proposed by (Jayasinghe, Lee, et al., 2017). In the model, $Coo_{ij}^{context}$ denotes cooperative.
- **Cost** – the cost of carrying out the transaction with the trustee. The cost can include energy consumption which can be calculated as proposed by (D. Chen et al., 2011). Entities may be more willing to carry out tasks that use less energy especially if it has limited resources. $Cos_{ij}^{context}$ represents cost in the trust model.
- **Credibility** - can be used to evaluate the services provided by a trustee. Credibility is represented by $Cre_{ij}^{context}$ in the model. It can be calculated as proposed by either of the following authors (Nitti et al., 2012; Yan Wang, 2005).

- **Sincerity** - the absence of deceit. This is important for selecting recommenders. The researcher proposes that when computing sincerity all the following properties are included:
  - **Reciprocate**
  - **Incentive**
  - **Encounter**
  - **Reliability**
  - **Defect**
  - **Belief**

  Sincerity can be computed as follows in the model:

$$Sinc_{ij} = \sum_{k=1}^{n} W_k P_k$$

- **Integrity –** According to Oxford dictionary integrity is the "*quality of being honest and having strong moral principles*". Integrity is a vital property in trust assessment. Integrity measures both the correctness of the functions, services and data provided by the entity. Integrity also takes into consideration unauthorised access and modification of information or data. Some researchers such as (Jayasinghe, Lee, et al., 2017) proposed the computation of honesty which took into consideration properties which include relationships and cooperative which is similar to the one proposed for the integrity property above. However, their model is limited to SIoT. After a thorough review of trust and proposed trust models, the following trust properties were identified to have a direct impact on integrity. Therefore, the following properties should be considered when computing integrity in an IoT environment:
  - **Relationship**
  - **Consistency**
  - **Willingness**
  - **Sincerity**
  - **Security**
  - **Commitment**
  - **Reliability**
  - **QoS**
  - **Defect**

  The proposes computation formula for integrity is as follows:

$$Inte_{ij}^{Context} = \sum_{k=1}^{n} W_k TVP_k$$

  Where W_k is the weight of the property. The weight will depend on the requirement of the truster. TVP is the trust value of the property in the specified contextAnd n is the total number of properties to be considered, in this case our n i,s equal to 10. Policies may also be taken into consideration under integrity.

- **Dependability** – a justifiable belief that the trustee will perform the requested task without failure. It can be computed from security, reliability, availability, QoS, and integrity. Dependability is denoted by $Depe_{ij}^{context}$ in the model.
- **Fulfilment –** achievement of the truster's requirements by the trustee based on the truster's predictions or expectations. It includes efficient and security.

Designing trust computation formulae for IoT is complex. The properties maintained by the IoT entities depends on the trust requirements of the entity. Each property kept by an entity will be associated with a trust value. Figure 6-7 shows some of the properties that can be considered for an IoT environment together with the relationships amongst them.

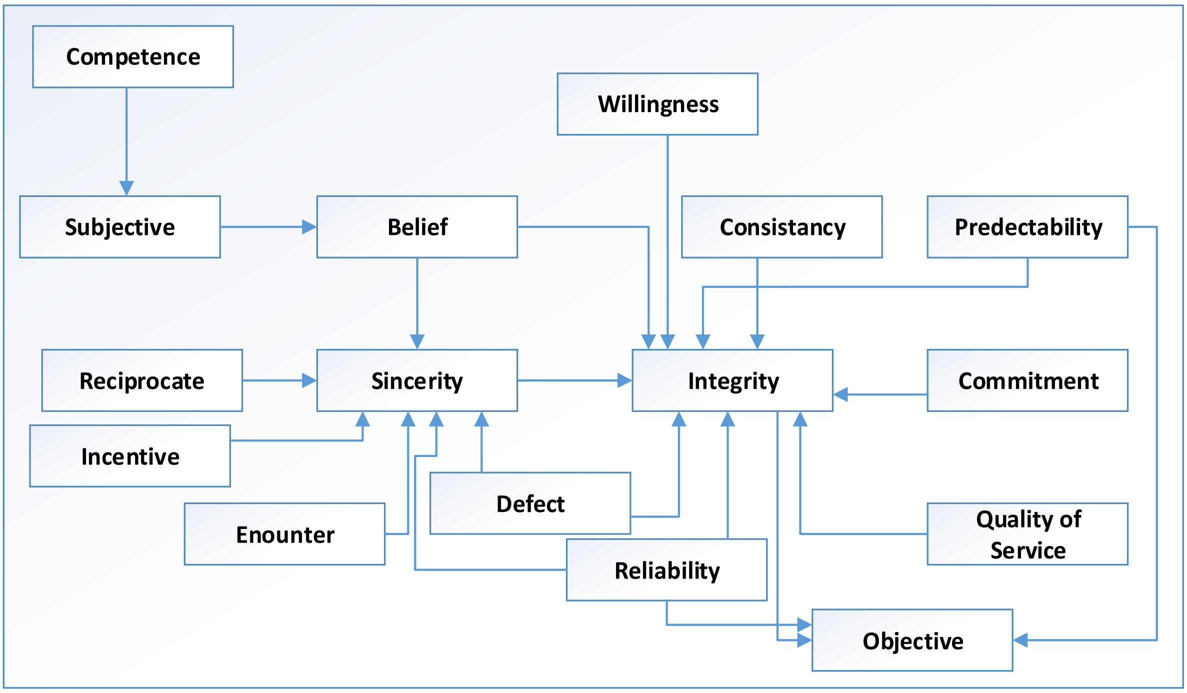


Figure 6-7: Relationships among trust properties

Each trust property x is measured as a real number such that $x \in [-1, 1]$. The trust agent keeps all trust information of all the entities in its domain. The trust values are kept together with their associated trust properties. After the interaction, the trust properties involved in the trust computation are re-evaluated and their trust value is updated. The trustee also evaluates its interaction with the truster and use the information to update the truster's trust information for both itself as well as for the trust agent. The trust agent can decide to update or not to update any trust information that is based on an entity's evaluation depending on its trust for the entity that is providing the information.

Suppose i is the truster and j is the trustee. For any trust property TP, its trust value is denoted by $TP_{ij}^{context}(t)$, where t is time. At a time t, $TP_{ij}^{p}(t)$ is updated as follows:

$$TP_{ij}^{p}(t) = \begin{cases} TPA + \sum_{k=1}^{n} W_k RPV_k \\ TPA(t) + TPA(t - \Delta t) \end{cases}$$

Properties availability, cost, and competence were added to the model in order to properly evaluate the trust values of each entity. If an entity is unavailable it doesn't mean that the entity is malicious or selfish, this also applies to entities that are incompetent. Properties like encounter and behaviour may be estimated by monitoring packet forwarding between the trustee and the truster.

6.6.4.3.3 Trust Propagation

In order to make the model scalable, the model uses a hybrid of both centralised and distributed propagation. Centralised propagation enables entities with limited resources to obtain their trust values from the trust agents. The distributed propagation enables the propagation of trust values among the trust agents and the entities. Trust propagation in the model will only be done by trust agents. All the other entities will only propagate trust information to their trust agent. Trust propagation also deals with how recommendations are handled by the truster.

6.6.4.3.4 Trust Formation

"*Trust formation refers to how to form the overall trust out of multiple trust properties*" (Guo et al., 2017). The multiple properties of trust will be combined using weighted summation. Weight assignment will depend on the requirement of the truster as well as the environmental situation.

- Trust context
- Trust recommendation
- Trust computation

Trust formation includes the defining the relevance of each trust property in calculating the total trust value. The proposed trust model is multi-dimensional. The model keeps both the total trust value and the trust values of all the properties. This enables the model to adapt the weight of each trust property according to the requirement of the truster.

6.6.4.3.5 Trust Aggregation

Trust aggregation deals with the evaluation of trust from the gathered information. This section covers the methods that are used to evaluate and analyse trust data. As highlighted in Chapter 5, the model uses fuzzy logic to aggregate trust evidence. Both trust and reputation are aggregated using a fuzzy inference system. Trust computation depends on the trust requirement of the trustee. Trust can either be subjective or objective. Subjective trust is based on the personal experience of the truster while objective trust is based on reputation and recommendation. The Initial stage of trust computation is information gathering. The following section explains how trusts data is gathered and analysed.

6.6.4.3.5.1 Information Collection and Analysis

As mentioned in Section 6.6.1, trust agents are responsible for information gathering. This will include agents occasionally initiating communication with random entities in their network in order to evaluate their behaviour. The root agents will also analyse the behaviour of trust agent in their networks.

Taking into consideration the fact that some ‘things’ lack computational ability, the trust agent is responsible for gathering information about each ‘thing’ in the network. The information gathered depends on the requirement of the entities in the network and it is determined beforehand. Reputation metrics can be obtained from the information that is gathered. The information can be gathered as first-hand information or indirect observation that is second-hand information (reputation). Second-hand information is obtained through observation of the network activities.

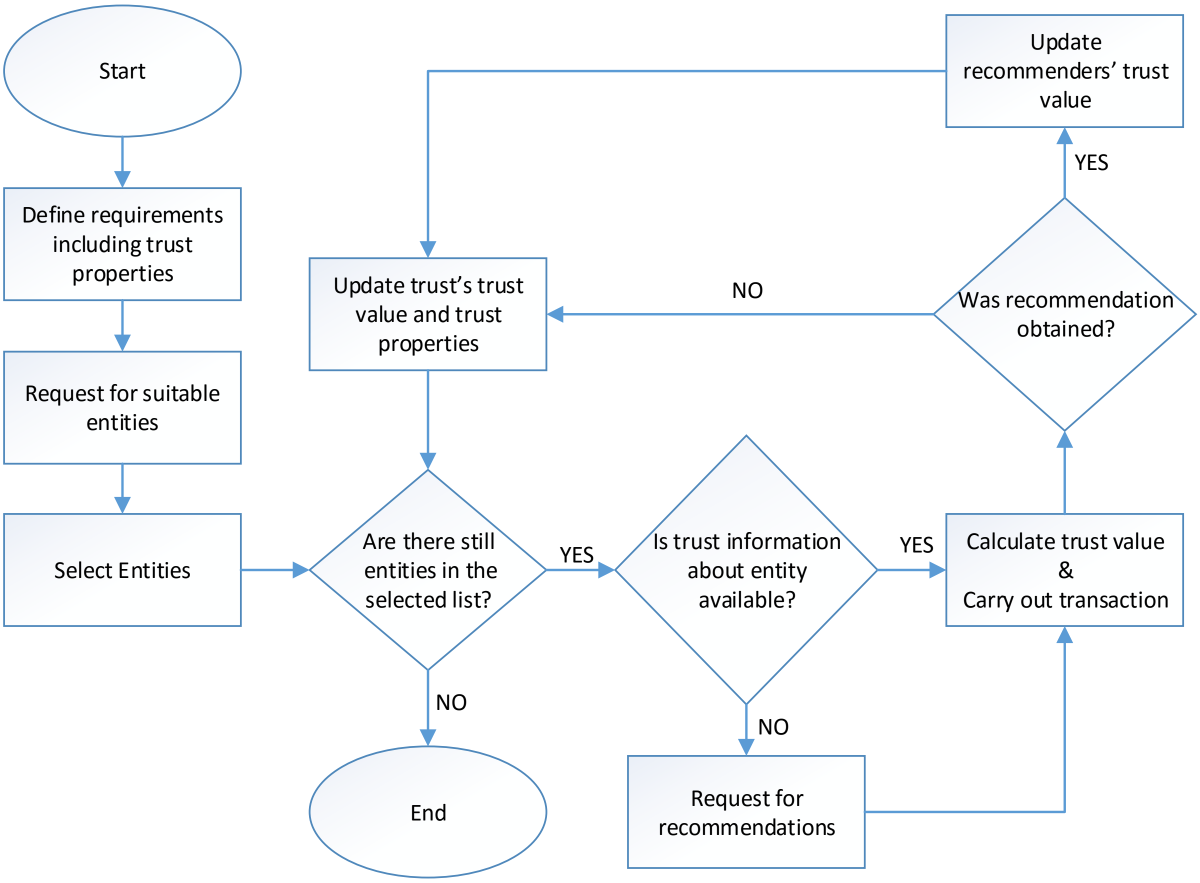

Start
Define requirements including trust properties
Request for suitable entities
Select Entities
Are there still entities in the selected list?
YES
NO
End
Is trust information about entity available?
YES
NO
Request for recommendations
Calculate trust value & Carry out transaction
Was recommendation obtained?
YES
NO
Update recommenders' trust value
Update trust's trust value and trust properties

Figure 6-8: Information gathering process

Figure 6-8 shows how the trust agents gather trust information. Recommenders include trust agents and entities. The trust information that is gathered by the trust agents depends on the trust requirements of the entities in the agent's domain.

6.6.4.3.5.2 Trustworthiness Evaluation

Trust evaluation can either be subjective or objective depending on the requirements of the truster and the information about the truster that the trustee has. Objective and subjective trust can be defined as follows:

- **Subjective** – the personal belief of the recommender or the truster. Subjective trust is denoted by $Sub_{ij}^{context}$. Subjective trust does not include recommendation.
- **Objective** – belief based on facts, knowledge, and experience. This can take into consideration packet delivery which can be computed as proposed by (D. Chen et al., 2011). Objective is based on the reputation of the trustee's capacity attained over time. $Obj_{ij}^{context}$ denotes objective trust in the model.

Both subjective and objective trust describes the type of information that is used when computing a trust value. Truster can compute a subjective trust value based on its own experience and recommendation from its own recommenders. Subjective trust enables entities to compute trust values based on personal experience which might be different from the experiences of other entities. This will help entities to create relationships among themselves. In the proposed model, objective trust is trust that is obtained from trust agents. The following section discusses how trust properties are combined in order to get the trust values of the trustee.

6.6.4.3.5.3 Fuzzy Inference Trust System

The fuzzy inference system is responsible for the evaluation the final trust value. One of the reasons for choosing fuzzy reasoning was because it is more tolerant of imprecise data compared to other methods and it effectively manages uncertainty. Fuzzy logic provides a method for evaluating imprecise and vague information. Rather than attempt to model a system mathematically, fuzzy logic incorporates rules to control the problem(Alnasser & Sun, 2017; Y. Yu, Li, Zhou, & Li, 2012). Also, a fuzzy system requires fewer computation resources compared to other computation methods. Figure 6-9 shows the position of the FIS in the trust model.

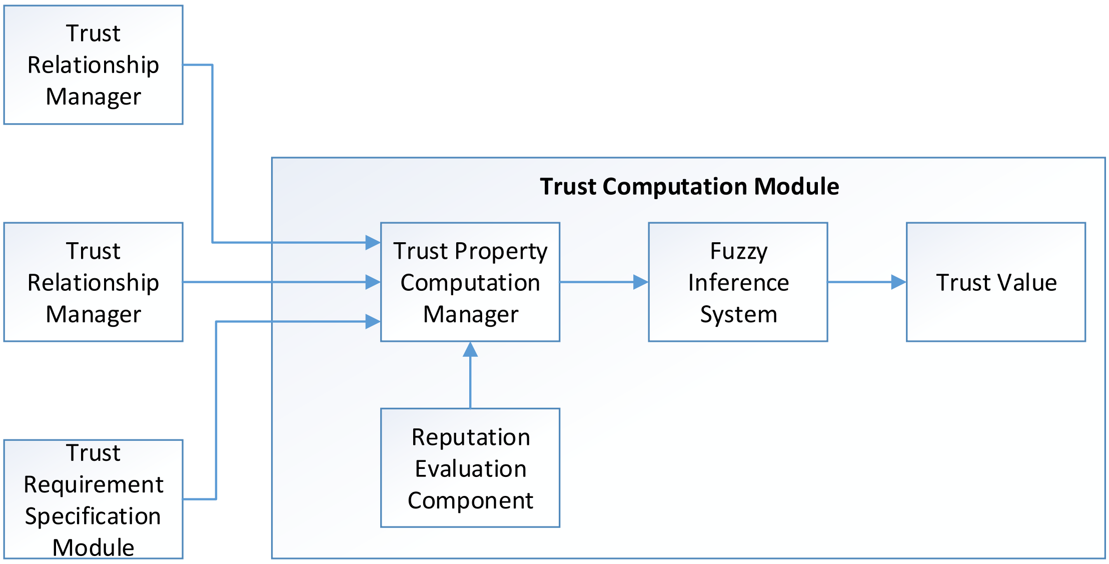

Trust Relationship Manager
Trust Relationship Manager
Trust Requirement Specification Module
Trust Computation Module
Trust Property Computation Manager
Fuzzy Inference System
Trust Value
Reputation Evaluation Component

Figure 6-9: Relationships among trust computation components

Fuzzy reasoning is achieved through the use of a fuzzy inference system. A fuzzy inference system is composed of four components: fuzzification, knowledge base, inference engine and the defuzzifier. The fuzzy inference engine evaluates several linguistic values through the use of IF-Then rules defined in the rule base. The fuzzy rules are in the form:

$$IF\ x_1 is\ A_1\ and\ X_2\ is\ A_2 \ldots and\ X_n\ is\ A_n\ THEN\ Y\ is\ B_n$$

Where A_i is a linguistic variable and B_n is an output which is also a linguistic variable associated with the given input(s). Each of the linguistic variable is associated with a membership function. The membership functions converts the crisp input x_i to the linguistic variable A_i. Figure 6-10 shows the components of a Fuzzy Inference System.

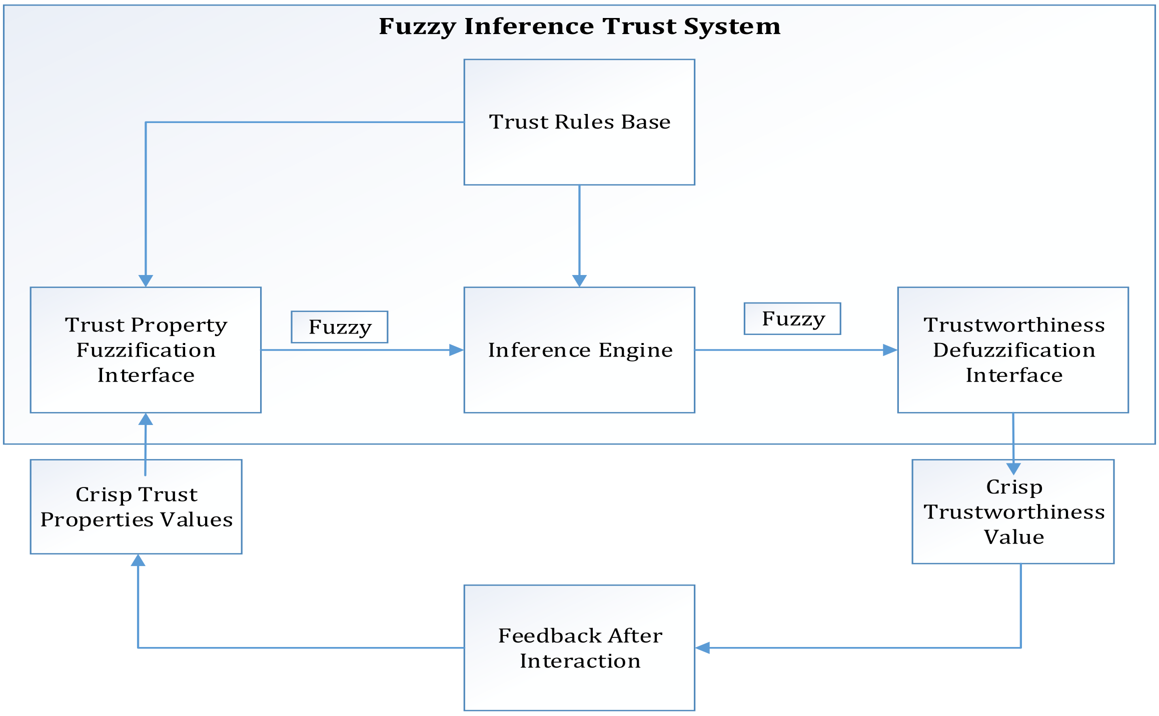

Fuzzy Inference Trust System
Trust Rules Base
Trust Property Fuzzification Interface
Fuzzy
Inference Engine
Fuzzy
Trustworthiness Defuzzification Interface
Crisp Trust Properties Values
Crisp Trustworthiness Value
Feedback After Interaction

Figure 6-10: The fuzzy inference system

The Mamdani and the Sugeno are the two common fuzzy inference systems. The Sugeno systems is well suited for the IoT trust model because it is “more compact and computationally efficient than the Mamdani” (The Mathwork Inc, 2017). The membership function of the Sugeno is a singleton which is an exact value. This shortens the process time of the fuzzy inference (Negnevitsky, 2005). The process for the fuzzy inference system is carried out through four steps:

1. Define the fuzzy sets.
2. Determine the degree to which the input values belong to the fuzzy sets
3. Apply the fuzzy rules to the input data
4. Evaluate the results

Fuzziness defines the degree to which a member belongs to a certain fuzzy group. In the fuzzy inference system, the membership function $M(x)$ define membership degree of a variable x. The membership of variable x is such that $x \in [0, 1]$. Therefore, $M(x)$ will map x to $[0, 1]$.

The universe of discourse for the trust model is $[-1, 1]$. The entities can customize the weights and select the rules that can be used. The weight for each property is determined by the importance of the property to the requirements of the truster.

6.6.4.3.5.3.1 Fuzzification

The input to the fuzzy system are crisp numerical values. These values are obtained from the trust properties. The crisp values are fuzzified against appropriate linguistic fuzzy sets. The range of the universe of discourse is defined by determining the extent that the crisp input belongs to the fuzzy set. The range of the universe of discourse is $[-1, 1]$ as defined by (S. P. Marsh, 1994). Since trust values cannot be measured directly, they are based on expert estimates. The fuzzification step consists of the following steps:

- Gather the crisp inputs
- Convert the input to fuzzy sets using membership functions and linguistic variables

For the trust model, the input variables will depend on the trust requirements of the truster. Each of the trust property will be fed into the FIS as a crisp variable. Each of the input variables (trust property) is mapped to linguistic variables using membership functions.

The following fuzzy values were identified for different trust properties: *VeryHigh*, *High*, *Low*, *VeryLow*, *Unknown* or *Negative*, *Neutral*, *Positive* or *Low*, *Medium*, *High*. All the membership functions for the fuzzy variables are trapezoidal membership functions. Trapezoidal membership functions are used because they are computationally efficient. Figure 6-11 shows the trapezoidal membership functions that defines some of the input membership sets.

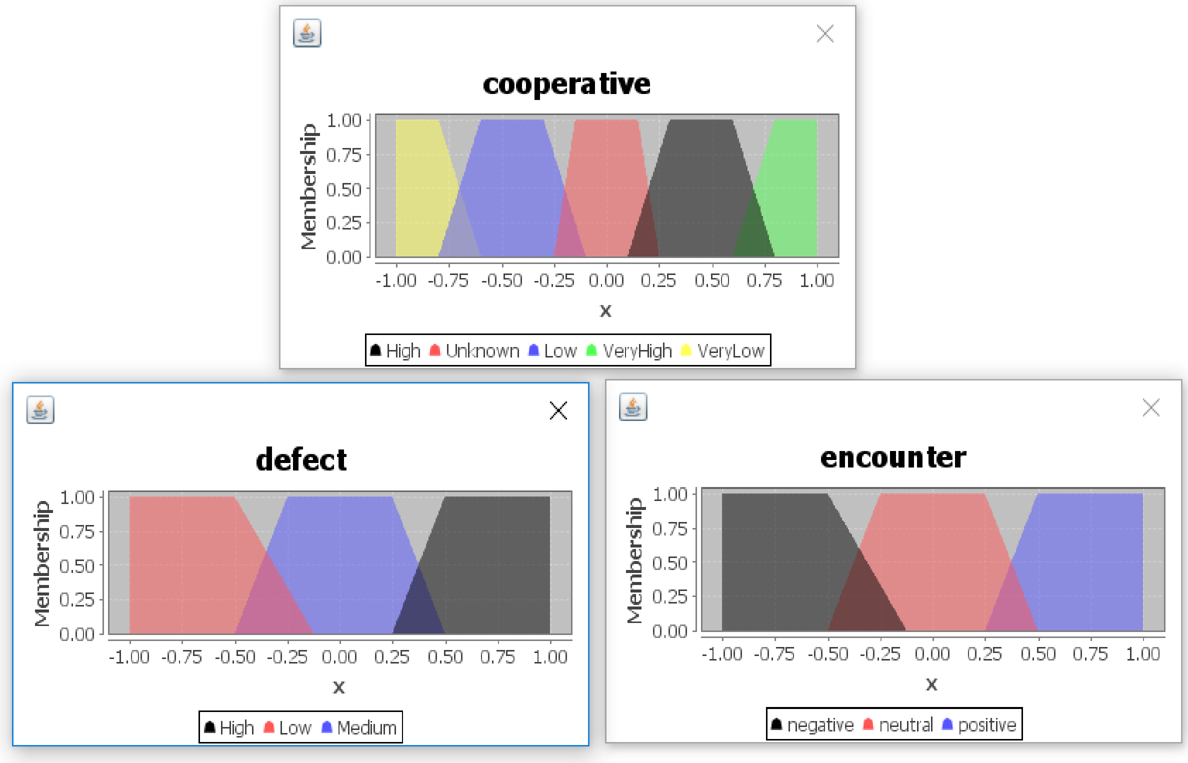


Figure 6-11: Membership functions for input variables

These membership sets define linguistic variables that are used in the inference engine rules. The membership functions are defined as follows:

- $$\mu_{high}(x) = \begin{cases} 0, & x < a_1 \\ \frac{x - a_1}{a_2 - a_1}, & a_1 \leq x \leq a_2 \\ 1, & a_2 \leq x \leq a_3 \\ \frac{a_4 - x}{a_4 - a_3}, & a_3 \leq x \leq a_4 \\ 0, & x > a_4 \end{cases}$$

Where x is a fuzzy value and $a_{1,}\ a_2, a_3, a_4\ \in\ \mathbb{R}$. Table 6-1 shows the proposed ranges of the membership functions.

Table 6-1: Ranges for the Linguistic Variables

Linguistic Variable	***Range***
Very High	(0.6, 1)
High	(0.1, 0.8)
Unknown	(-0.25, 0.25)
Low	(-0.8, -0.1)
Very Low	(-1, -0.6)

The number of inputs for the FIS will depend on the requirements of the truster. Each crisp input is mapped to a fuzzy membership function. As mentioned earlier, the universe of discourse for the linguistic input variable is [-1, 1].

6.6.4.3.5.3.2 Rule Evaluation

The set of rules defines how trust information is gathered and how trust values are computed. In this stage, the fuzzified inputs are applied to the antecedents of the rules. In rules with multiple antecedents, the AND and OR operators are used to obtain a single number that represents the result of the antecedent evaluation (Negnevitsky, 2005). The obtained number is applied to the consequent membership function (Negnevitsky, 2005). The rules are of the form: If input then output. The ma operator is used to aggregate the outputs into a single value. The rules take into consideration the relationships among the trust properties shown in Figure 6-7. **Error! Reference source not found.** shows examples of rules defined in the Trust Rule base.

Rule 1: IF security is Very Low Then Trustworthiness is Undistrust
Rule 2: IF Risk is unknown then Untrust
Rule 3: IF Integrity is Unknown THEN Untrust
Rule 4: IF Sincerity is Low Then Trustworthiness is Low
Rule 5: IF Integrity is High THEN Trustworthiness IS Trust
Rule 7: IF Integrity is Very Low THEN Trustworthiness is Distrust
Rule 8: IF Defect is High and Encounter is High Then Trustworthiness is Undistrust
…
n

Figure 6-12: Examples of rules

6.6.4.3.5.3.3 Defuzzification

The final stage of the FIS is to aggregate the output from the defuzzification process into a single output value. In this model, the final output of the fuzzy model is the trustworthiness of the trustee which is a single linguistic variable. The linguistic variable trustworthiness may be defined by any one of the following fuzzy values:

- Distrust – there is enough evidence to conclude that the trustee is intentionally malicious. One an entity is distrusted; it cannot be trusted again. It becomes black-listed.
- Undistrust – there is little evidence which is not enough to conclude that the trustee is malicious.

- Ignorance – there is no evidence to either trust or distrust an entity.
- Untrust – there is little positive evidence that can enable the truster to trust the trustee or vice versa.
- Trust – there is enough evidence to conclude that the trustee can be trusted.

In defuzzification, the fuzzy output is mapped into crisp variables using membership functions. Since trust is a positive expectation and distrust is a negative expectation, therefore, I am going to use values ranging from -1 to 1. Figure 6-13 shows the membership functions of output fuzzy value.

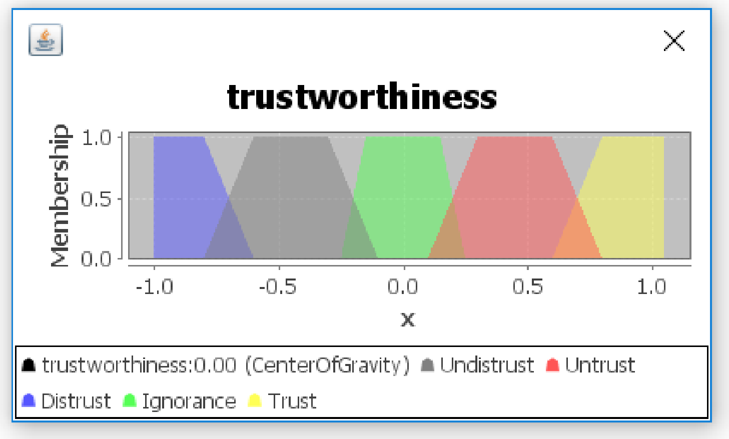


Figure 6-13: Membership function for trustworthiness

Trust smoothly transitions to distrust. As mentioned above, the FIS uses the Sugeno model. The Center of Gravity (CoG) or centroid method is the commonly used method for defuzzification. It is defined as follows:

$$CoG(A) = \frac{\int_x \mu_A(x).x.dx}{\int_X \mu_A(x)dx}$$

$$CoG(A) = \frac{\sum_{q=1}^{N_q} \mu_A(x).x}{\sum_{q=1}^{N_q} \mu_A(x)}$$

6.6.4.3.6 Direct Trust Computation Process

Trust can be based on either direct experience or indirect experience. The process for computing trust value depends on the truster as well as the requirement of the truster. If the truster prefers computing its own trust value, then the entity follows the process in Figure 6-14.

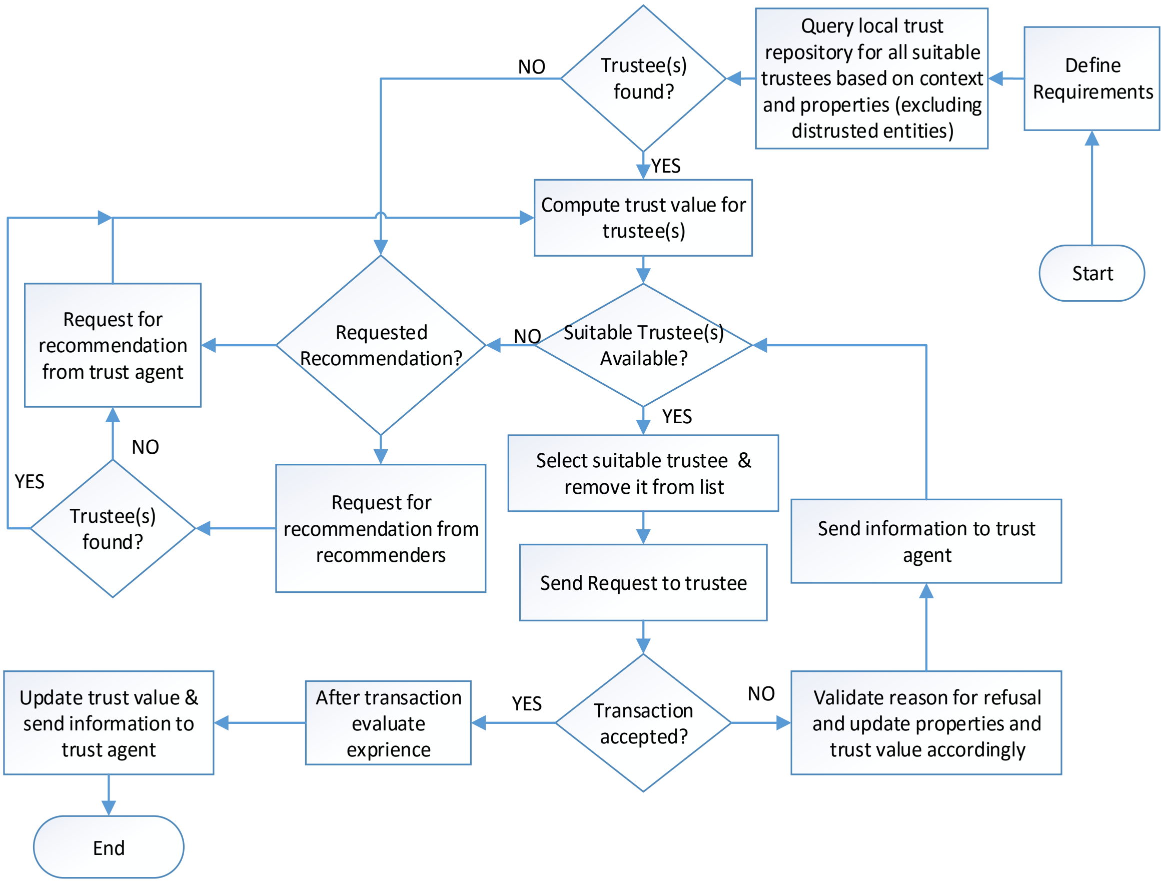

Define Requirements
Start
Query local trust repository for all suitable trustees based on context and properties (excluding distrusted entities)
Trustee(s) found?
NO
YES
Compute trust value for trustee(s)
Suitable Trustee(s) Available?
NO
YES
Requested Recommendation?
Request for recommendation from trust agent
Request for recommendation from recommenders
Trustee(s) found?
NO
YES
Select suitable trustee & remove it from list
Send Request to trustee
Send information to trust agent
Transaction accepted?
YES
NO
Validate reason for refusal and update properties and trust value accordingly
After transaction evaluate exprience
Update trust value & send information to trust agent
End

Figure 6-14: Trust value computation by an entity

The assumption that a trust agent will always provide the truster with trustee(s) was made because of the following reasons:

- If the trust agent doesn't find a suitable trustee it will request suitable trustees from its recommenders (both entities and trust agents).
- If the recommenders cannot provide suitable recommendation the trust agent will request recommendation from the root agent.

Since the root agents keep records of a register of entities that include their trust agents and the services they offer, this guarantees that suitable trustee(s) will be found. However, this does not guarantee the trustworthiness of the trustee(s). It is up to both the trust agent and the truster to validate the trustworthiness of the trustee(s). Trustee has the option of declining the transaction with the truster, but the trustee needs to have a valid reason otherwise it will be given a bad report and this will affect its trust value.

If a trustee rejects a transaction, the truster can evaluate its experience about the trustee and use it to update its trust value and trust properties values for the trustee. The truster also sends its experience to its trust agent. This helps in identifying selfish entities in the network. The trust agent can validate the received report and update the trust information of both the truster and the trustee accordingly. If the truster sends false information its trust information will be impacted negatively. This prevents bad-mouthing among entities in the network. The experience is sent as feedback to the trust agent and is denoted by $ER_{ij}^{t}(c)$. t is the transaction identity, i is the truster who is sending the report, j is the trustee being evaluated and c is the context.

Figure 6-15 shows the process of how a truster obtains trust information about suitable trustees from the trust agent. The trust agent sends both the trust value and trust properties' values to the truster. This gives the truster the option of re-calculating the trust value according to its own requirements. This is important because trust is a personal decision and the proposed model supports this element. If the truster lacks computational capability it can discard the unnecessary information.

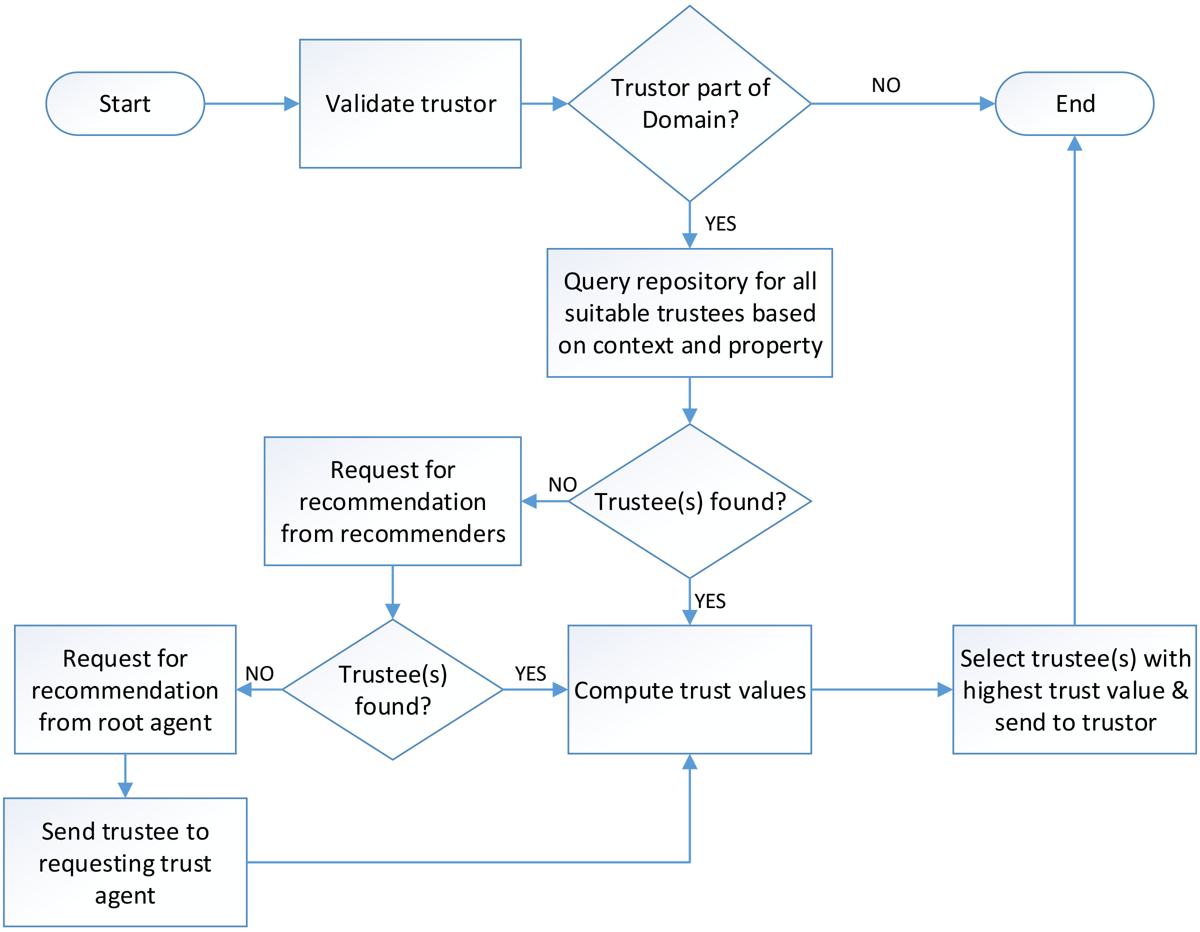

Start
Validate trustor
Trustor part of Domain?
NO
End
YES
Query repository for all suitable trustees based on context and property
Request for recommendation from recommenders
NO
Trustee(s) found?
YES
Request for recommendation from root agent
NO
Trustee(s) found?
YES
Compute trust values
Select trustee(s) with highest trust value & send to trustor
Send trustee to requesting trust agent

Figure 6-15: Trust value computation by a trust agent

The trustee also evaluates its experience with the truster and update it trust value and trust properties accordingly. The trustee can also send its experience to the trust agent. If the trustee sends false information its trust value will be impacted negatively. This prevents bad mouthing attack. Trust values are updated as follows:

$$TV(i,j)_x^{tr} = TV(i,j)_x^{tr-1} + CTV_x^{tr}$$

Where i is the truster, j is the trustee, tr is the transaction number and x is the context.

Since trust is bi-direction, Figure 6-16 shows the process that a trustee may go through when it receives a request for a transaction. Continuous rejection of a transaction without a valid reason will result in the trustee being labelled selfish. A selfish may regain trust from other entities if it starts collaborating with other entities.

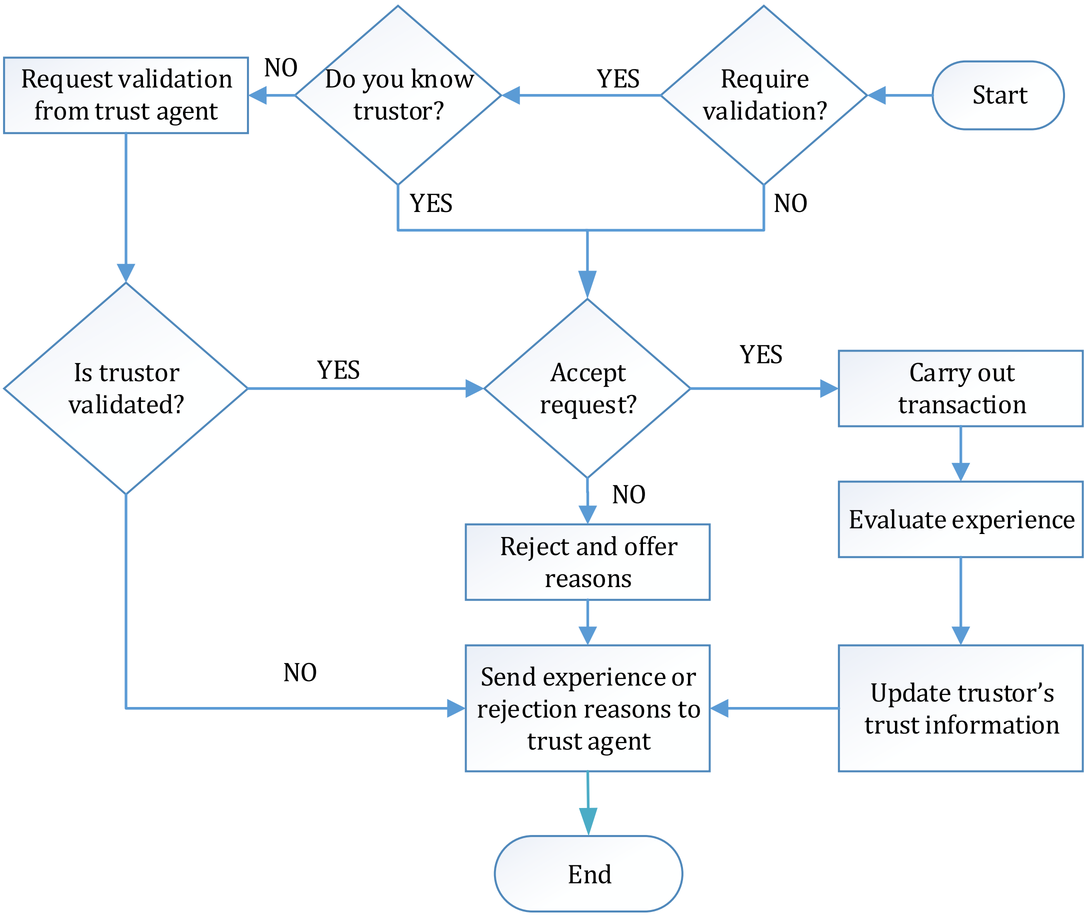


Figure 6-16: Process for trustee

The valid reasons for a trustworthy trustee rejecting a transaction include competence, availability and lack of trust of the truster. The consideration of the trustee's trust for the truster prevents malicious entities bad mouthing trustworthy entities. A malicious or selfish entity have no valid reasons for rejecting a transaction. After computing trust values, the truster has to make a decision based on trust value and other factors when selecting the suitable trustee. The next section briefly explains the process of decision making and authority delegation that may be done by the truster.

6.6.4.3.7 Indirect Trust computation

Objective trust computation is divided into two types: recommendation and reputation. Objective trust takes into consideration the experiences of other entities. The following section discusses how recommendation will be calculated in the proposed trust model.

6.6.4.3.7.1 Recommendation

Recommendations enable a truster to make an informed decision in the absence of direct interactions. One of the properties of trust is composability. This makes it possible to obtain and combine recommendations from other entities. The recommendation contributed by a 'thing' depends on the trust that the truster has on the recommender. In this case, the truster takes into account its trust for the recommender. Weighted summation has been proposed in calculating recommendation. The weight of the recommender depends on the trust that the truster has on the recommender.

An entity chooses recommenders based on the context of its requirements and the set of recommenders it trusts. A recommender's trust value in a particular context is evaluated after each transaction based on accuracy. Each recommendation value provided is weighed with a value based on the recommender's trust value. The recommender can provide recommendation as either a real number or a linguistic variable. Fuzzy logic is an appropriate computation method because it uses qualitative terms and linguistic labels. When entities provide recommendation, instead of using values only, they may use linguistic labels: trust, untrust, ignorance, undistrust, distrust. Values can then be assigned using fuzzy membership functions.

Recommenders are part of the trusted entities but a trusted entity might not be part of the recommenders because the trustee might not have enough knowledge to trust the entity as a recommender. The recommender's trust value is dependent on context to a greater extent. An entity uses only trusted entities as its recommenders. Apart from obtaining information from its recommenders, an entity can obtain information from trusted agents. A trusted agent can recommend another agent to an entity.

Therefore, in cases where recommendation is used the trust that the truster has for the trustee will depend on both the trust that the truster has on the recommenders and the trust the recommenders have on the trustee. In order to prevent false recommendation, only trusted 'things' will be requested for recommendation. The following conditions have to be met for each recommender:

- The truster trusts the recommender
- The recommender explicitly recommends the trustee to the truster

Even after a recommendation has been made, the truster still has to decide how much it is going to trust the trustee.

The recommendation provided is also weighed by a context weight. The context weight depends on the recommender's trust context relevance. The weight of the context has the range (0,1]. The recommendations provided may be computed using a modification of the formula suggested by (Mahalle et al., 2013):

$$RT(j, c, RV) = \frac{\sum_{k=1}^{n} W_k RV_k}{\sum_{k=1}^{n} RV_k}$$

Where j is the recommender's identity, c is the context, W is the weight of the recommender and RV is the recommender's recommendation value. The weight W can be obtained as follows:

$$W_{ij}^{c}\,(t, x) = \frac{RT_j}{\sum_{k=1}^{n} RT_k}$$

Where c is the context, i is the truster, j is the recommender, t is the time, x is the trustee, n is the total number of recommenders that i has requested recommendation from and RT is k's recommendation trust value. The following section discusses how reputation will be calculated.

6.6.4.3.7.2 Reputation Component

Reputation shows the overall past behaviour of the trustee obtained from all the entities that the trustee has transacted with; it doesn't indicate the trust value of a particular truster in the trustee. In the proposed model reputation is computed by the trust agents and dedicated root agents. Reputation can be used in determining the intentions of the trustee whenever the trustee is unknown to the truster. Policies can be used to specify how reputation can be used apart from the trust model (S. Marsh et al., 2012). Security breaches cause entities to lose their reputation and this will affect their interaction with other entities in the network. Reputation is important when determining trust in a social environment. (Gutscher et al., 2008) suggested that there are three basic types of reputation systems which have different ways of calculating reputation values:

- Flat Reputation System – in this case reputation is computed from all available trust opinions and all the opinions weighed.
- Recursively Weighting Reputation System – reputation is computed by ranking opinions. High ranked opinions will be given a higher weight. "*The new reputation values of all entities are computed from the opinions of all other entities weighted by their reputation values of the last iteration*."
- Personalized Reputation System with Trust Anchor – this type of system prevents malicious entities from manipulating computed reputation value by dominating *"public opinion"* by only taking into consideration the opinions of the apriori trusted entities and trustworthy entities based previous experience. This enables the systems to ignore the opinions of malicious entities.

I propose a personalised reputation system with trust anchor for this research. It is important to ensure that the reputation component is able to handle the computation of the same entity in different contexts. The reputation system should be able to identify false or misleading reputations. The reputation value of a trustee is a global value estimated by all the trusters that have interacted with the trustee. It is built from the overall behaviour of the trustee. It is based on the context of the past behaviour. This means an entity can have a different reputation for different contexts. The reputation component supports the following:

- handle the computation of the reputation of the same entity in different contexts. This means an entity can have different reputations for different contexts.
- identify false or misleading feedback. The system should not assume that all feedback is honest.
- support decaying of old transactions.

The reputation is integrated to the trust model. The trust agents and root agents also handle the reputation computation. In addition to the roles highlighted before, the root agents also evaluate the reputation values of both trust agents and root agents. The root agents are not involved in any reputation computation for entities. They are responsible for calculating reputation values for trust agents as well as for each other. This will eliminate malicious agents. In addition to the already highlighted responsibilities the trust agents also:

- Keep reputation values of all entities in their domain as well as entities in other domains that have cooperated with any of the entities in their domain
- Evaluates reputation value upon request from an entity in their domain
- Keep reputation values of all trust agents and root agents they are interested in
- Collect and analyse reputation data in their domain

The root agents will evaluate network activities to identify malicious trust agents while trust agent will be evaluating the network activities to identify malicious entities and trust agents as well. Reputation estimates the trustworthiness of an entity based on the opinions of recommenders. In the proposed model the recommenders could be trust agents or entities in the network who have communicated with the entity. An entity can query the reputation of another entity from its trust agent.

6.1.1.1.1.1.1.1 Reputation Computation

The trust agents store each recommendation values for each context from each entity separately and updates the value as the recommender provides new recommendation. When computing reputation value of an entity, a weight is assigned to each recommender. Reputation depends on the recommendations from other entities or trust agents. Each recommendation is weighed according to the trust that the reputation agent has in the recommender. The sum of all the weights considered for each recommender in a single computation is equal to one. Reputation will be computed as follows:

- reputation values which related to the defined context are collected
- a weight is applied to each reputation value based on its relevance and on the reputation trust value

Reputation is the accumulation of behaviour of the entity based on all the transaction the entity has participated in. The recommendation provided for reputation computation is provided in the form of values of properties that were considered in the collaboration. The trust agents keep these properties values separately and combines them when a reputation request is made. This enables the trusters to obtain personalized reputation values. When computing reputation from multiple entities similar properties are combined into a single value using the following formula:

$$P_i^x = \frac{\sum_{j \in S} w_j t_{ji}}{\sum_{j \in S} w_j}$$

where P_i is the total value for property x of entity i, S is the set of all the entities that have interacted with entity i, t_{ji} is property value of i rated by j, w_j is the weight for entity j. The weight w_j for each recommender can be computed using a Fuzzy Inference System (FIS) using the relevant properties selected by reputation requestor. **Error! Reference source not found.** show how w_j is computed.

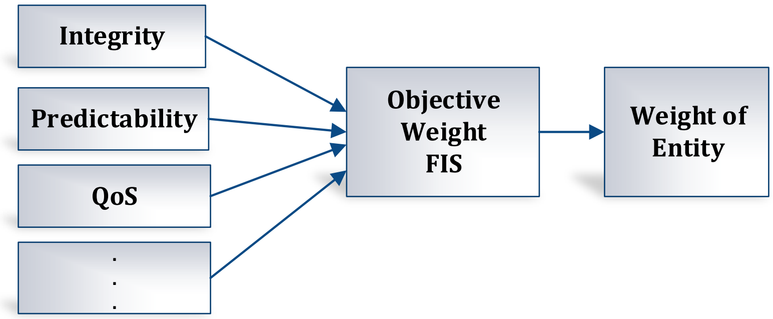


Figure 6-17: w_j Computation

6.1.2 FIS for recommender weight computation

The FIS for computing the weight of entities takes in the values of the properties and outputs the weight of the recommender. The universe of discourse for the properties is $[-1, 1]$. The linguistic variables for the properties are: *very low*, *low*, *medium*, *high* and *very high*. Figure 6-18 shows the membership function for QoS.

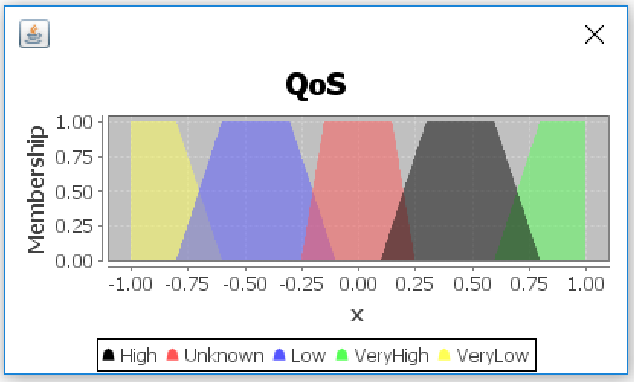
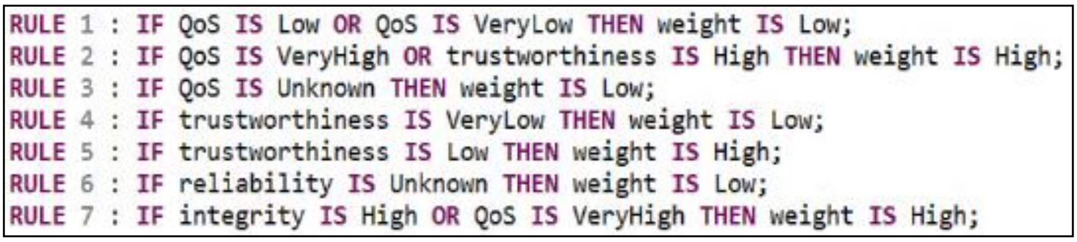


Figure 6-18: Membership functions

The properties used in computing reputation will depend on the requirements of the reputation requestor. After determining the fuzzy sets to which the input sets belong, fuzzy rules are applied to the input data. Figure 6-19 shows some of the rules that were used in computing the weight of the recommender.



```
RULE 1 : IF QoS IS Low OR QoS IS VeryLow THEN weight IS Low;
RULE 2 : IF QoS IS VeryHigh OR trustworthiness IS High THEN weight IS High;
RULE 3 : IF QoS IS Unknown THEN weight IS Low;
RULE 4 : IF trustworthiness IS VeryLow THEN weight IS Low;
RULE 5 : IF trustworthiness IS Low THEN weight IS High;
RULE 6 : IF reliability IS Unknown THEN weight IS Low;
RULE 7 : IF integrity IS High OR QoS IS VeryHigh THEN weight IS High;
```

Figure 6-19: Rules for weight computation

The evaluation of the rules produces the output. In this case the output is the weight. The process of evaluating the results is known as the defuzzification process. The weight of the recommender is out as real number in range (0,1). Figure 6-20 shows the membership functions for the weight.

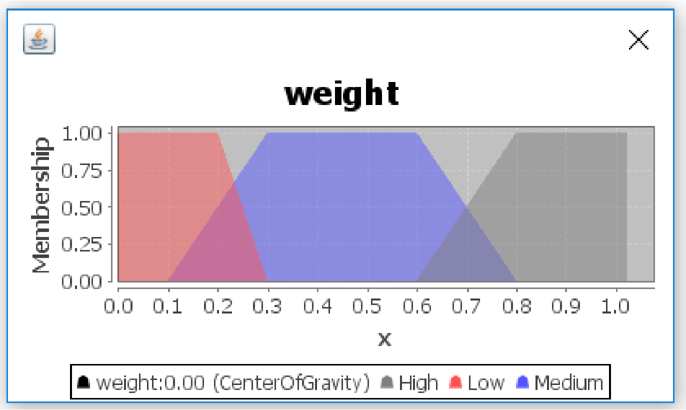


Figure 6-20: Membership function for weight

The linguistic variables for the weight are *low, medium* and *high*. In defuzzification process, the fuzzy output is mapped into crisp variables using membership functions. The CoG method was used in the defuzzification process.

6.1.2.1.1.1.1.1 Fuzzy component for reputation computation

The FIS for reputation takes in properties as input outputs the reputation value. The reputation value obtained from the reputation system may be used together with personal trust values. This will enable entities to make more reliable decisions. Trust relationships can be easily created from reputation. Fuzzy logic enables multiple properties to be combine into a single reputation value. The properties are fed into the FIS as real numbers. Figure 6-21 shows FIS for reputation computation.

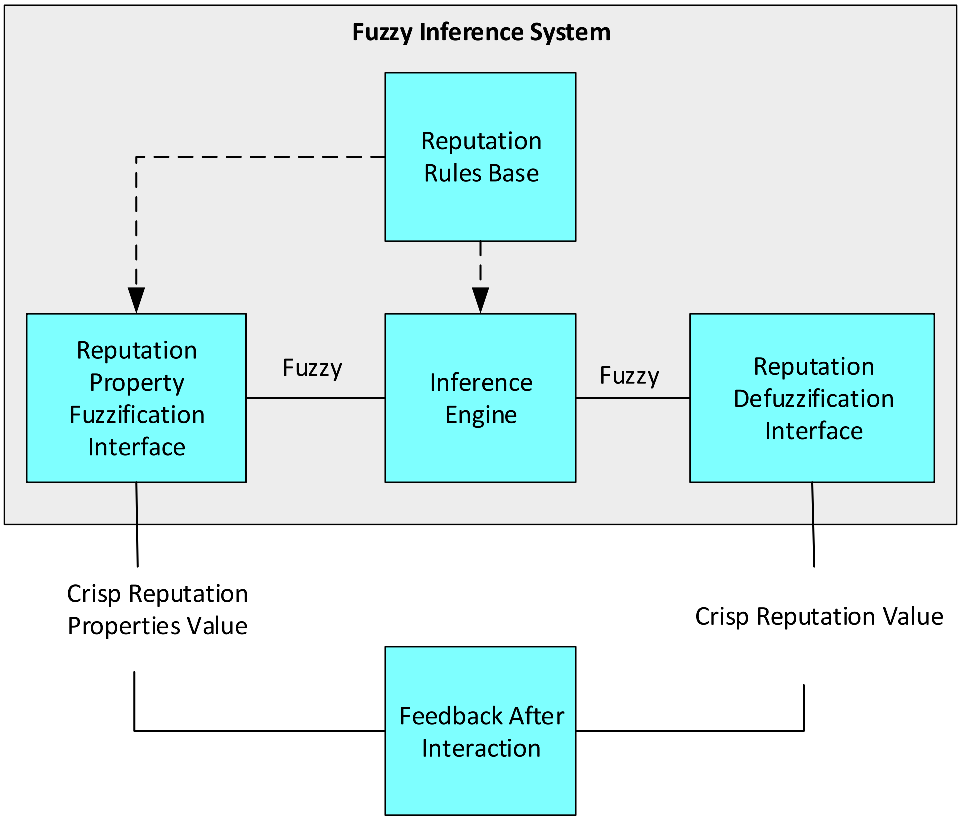
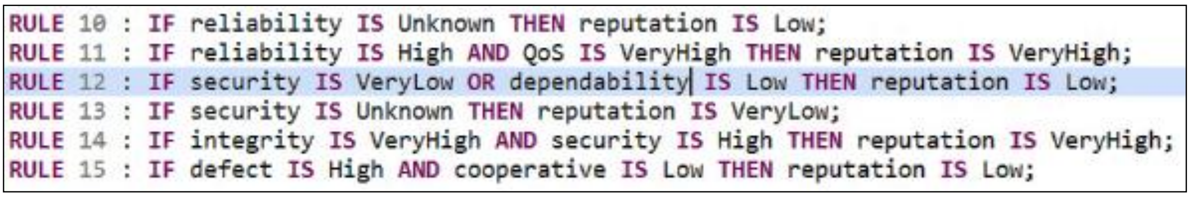

Fuzzy Inference System
Reputation Rules Base
Reputation Property Fuzzification Interface
Fuzzy
Inference Engine
Fuzzy
Reputation Defuzzification Interface
Crisp Reputation Properties Value
Crisp Reputation Value
Feedback After Interaction

Figure 6-21: FIS for reputation computation

The crisp values are fuzzified against the linguistic fuzzy sets. This process is followed by rule evaluation. Figure 6-22 shows a summary of some of the rules used in reputation computation.



```
RULE 10 : IF reliability IS Unknown THEN reputation IS Low;
RULE 11 : IF reliability IS High AND QoS IS VeryHigh THEN reputation IS VeryHigh;
RULE 12 : IF security IS VeryLow OR dependability IS Low THEN reputation IS Low;
RULE 13 : IF security IS Unknown THEN reputation IS VeryLow;
RULE 14 : IF integrity IS VeryHigh AND security IS High THEN reputation IS VeryHigh;
RULE 15 : IF defect IS High AND cooperative IS Low THEN reputation IS Low;
```

Figure 6-22: Rules for reputation computation

As mentioned earlier, defuzzification it the last process. The linguistic variables for reputation are: *very low, low, unknown, high and very high*. The universe of discourse for the variables is $[-1, 1]$. The ranges of the membership functions are the same as the ones shown in.

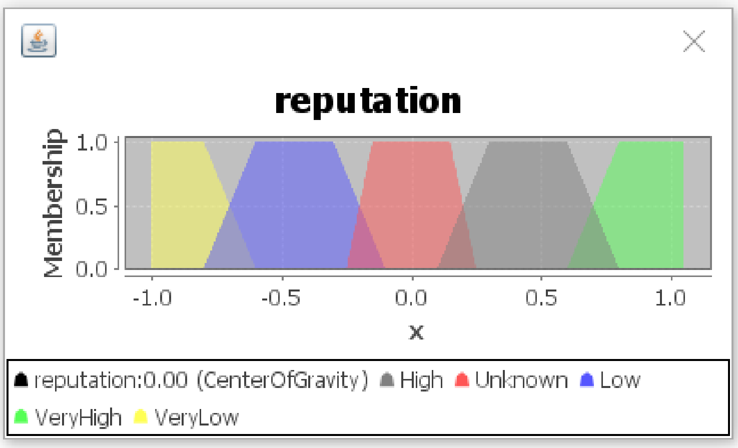


Figure 6-23: Membership functions for Reputation

Reputation is a type of a recommender system where each entity in the system is a recommender after each transaction. Reputation values can be used to create trust relationships. A reputation system allows entities to share trust information with each other. Reciprocity is important in reputation which should be taken into consideration when computing reputation values. However, malicious entities might try to use reciprocity to boost each other's reputation and this needs to be prevented by the system. The proposed reputation system attempts to prevent this attack by taking note of abnormally high number of transactions between entities.

6.6.4.3.8 Trust Property Update

Trust is dynamic and non-monotonic. The proposed model keeps the values of the trust properties instead of the actual trust value because the IoT environment is dynamic and its changes affect the variables used in trust computation. Therefore, the property values of an entity change over time. In the proposed model, the properties are updated in two ways:

- Trust deterioration over time: Trust value can deteriorate with the time with a factor of $T_j = \frac{1}{e^{1-t}}$, where j is the trustee and t is the timestamp.
- After interaction: After each interaction either the trust value or the recommendation trust value updated depending on the type of interaction. The update of the trust value will depend on the trust properties involved as well as the outcome of the interaction.

The trust property values decay over time.

6.6.5 Decision Making and Authority Delegation

It is inevitable for entities in the network to take risks at some point in time. Taking risks is necessary in order to create relationships or strengthen the relationships. However, the risk that is taken needs to be justified. In the model, i Trusts j where i is the truster and j is the trustee means:

- i has a choice to work with j
- is j willing to cooperate with i
- i has knowledge of j
- i does not owe j
- i has knowledge of the context and network

This is based on the computational values and the outcome of the computation module. Each entity in the network should be able to decide:

- whom to cooperate with
- what extent should it trust
- When to cooperate

It is crucial that the trustee selects a trustworthy entity because the IoT consists of numerous entities with varying behaviours. This brings us to the issue of authority delegation in IoT. Authority delegation is the transfer of authority to make decisions and complete specific tasks. In IoT devices can delegate other devices to access remote resources or carry out some tasks on their behalf. Authority delegation includes authorization. Authority delegated to any entity in the IoT network depends to the trust that the truster has in the trustee. Authority delegation is important in an IoT environment as it allows machine to machine communication without human intervention. Authority delegation is based on the policies of the system or the policies governing the device and it can be integrated into a trust management model. Trust management enables the trusters to select trustworthy trustees. This will increase the chances of successful transactions and prevents the loss or unauthorised access of private information.

Delegation can be done with or without trust depending on the task and information being exchanged. Authority delegation includes authorization. The authority delegated to anything in the IoT network depends on the trust that the 'thing' has about the other 'thing'.

6.6.6 Trust Networks

Trust help entities to create a stable environment. Once the trust for entities has been established based on experience and knowledge, the trust agents will begin to create trust networks among themselves. The trust values for the trust agents are affected by the behaviour of the entities in their trust domain. The aim of the trust network is to increase the response time of the trust agents and to eliminate any untrustworthy trust agents and entities. The trust networks may be used to define trust relationships among entities.

6.7 Conclusion

The ability of the FIS to handle both numerical and linguistic data make it appropriate for trust evaluation. The use of linear membership functions reduces computation complexity. The main purpose of the trust model is to prevent the entities in the network against malicious attacks. Understanding the structure of a trust model for IoT is difficult because both trust and the IoT network are complex. Trust is complex because it is a human element. Trust is both subjective and objective. This means it can be computed from experience, knowledge and recommendation. Over time trust can be evaluated using reputation. The IoT network is complex because of the heterogeneous devices found in the network. The proposed model implemented experience, knowledge and recommendation in computing trust. After proposing the model, the next step is to test and validate the model. The next chapter explains in detail how the model tested and validated. Each entity that has computational capability maintains trust data of the entities that it constantly interacts with. A trustee entity is allowed to reject communication with a truster if it does not trust the truster.

7

Proposed Model Evaluation

7.1 Introduction

The purpose of a trust model is to limit interaction of trustworthy entities with untrustworthy entities. This chapter aims to demonstrate the feasibility of the proposed trust model in achieving this purpose. The chapter further details the evaluation of the performance of the proposed trust model in a simulated environment. The simulated environment consisted of root agents, trust agents, and entities. The entities in the environment had different computational capabilities and offered different services. The main aim of the tests done on the simulated environment is to validate the effectiveness of the model.

Taking into consideration the resources available for IoT entities, it is important to ensure that the trust model is light and that the computation method is light as well. This was taken into consideration during the design of the model. The simulation environment that suited the requirements of the experiments that needed to be carried out in order to test for the feasibility of the model was designed by the researcher. Given trust information, the model should estimate the behaviour of the trustee as accurately as possible. The accuracy of the trust model validates the feasibility of the trust model. The following section describes the simulated environment.

7.2 Implementation of the Simulated IoT Environment

The proposed trust model and the simulated environment were implemented in Java. The trust model was implemented in two steps:

- Create the Fuzzy Inference System (FIS):
    - Define linguistic variable for trust properties
    - Define relations among linguistic variable

    - Create the rule base
    - Define the defuzzification method
- Implement the trust model application

7.2.1 Fuzzy Inference System Implementation

As mentioned in Chapter 6, the FIS consists of four main components: fuzzifier, rules, inference and defuzzifier. The FIS is the central component for trust value evaluation. The FIS was created using jFuzzyLogic which is an open source Java library "*which offers a fully functional and complete implementation of a fuzzy inference system*" (Cingolani & Alcalá-Fdez, 2013). The JFuzzyLogic enables the easy development of the FIS in java. The jFuzzyLogic library uses the Fuzzy Control Language (FCL) to create the FCL file. FCL is a Fuzzy Control Programming Language defined in the International Electrotechnical Commission (IEC) 61131 part 7 (IEC 61131, 2000). The jFuzzyLogic library also supports the execution of the FCL file. The input values to the FIS are crisp values. In the trust model, the trust property values are the crisp input values.

Trust property values are fed into the fuzzy system as crisp values and are resolved into linguistic variable by the fuzzification process. The fuzzification process quantify the crisp values into linguistic variable using membership functions. The jFuzzyLogic only supports one type data which is REAL. This data type suits the testing of the trust model because the crisp values for input are in the range of either [0,1] or [-1, 1] depending on the trust property. Figure 7-1 shows the groups of linguistic variables that the trust property may take.

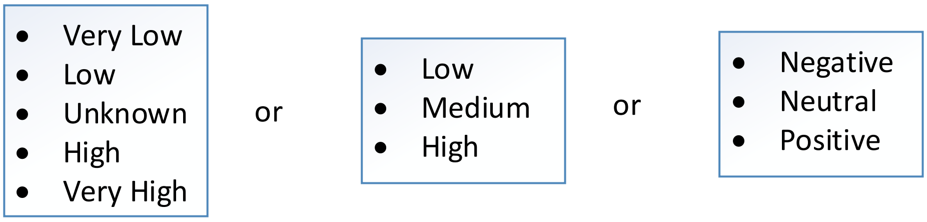


Figure 7-1: Input Linguistic Variables

7.2.2 The FCL file

Each FCL file is composed of a function block which consists of the following sections: definition of the linguistic variables, membership function for both fuzzification and defuzzification respectively and definition of the fuzzy rules which is the rule block. The first part of the FCL file is composed of the declaration of the input and output linguistic variables. This declaration is followed by the definition of the fuzzification and defuzzification membership functions definition. For the proposed trust model, membership functions for both the input variables and output variable are trapezoidal. Trapezoidal functions are suitable for IoT because they are computationally efficient. Figure 7-2 shows membership functions for three trust properties inputs and the output variable which is trustworthiness.

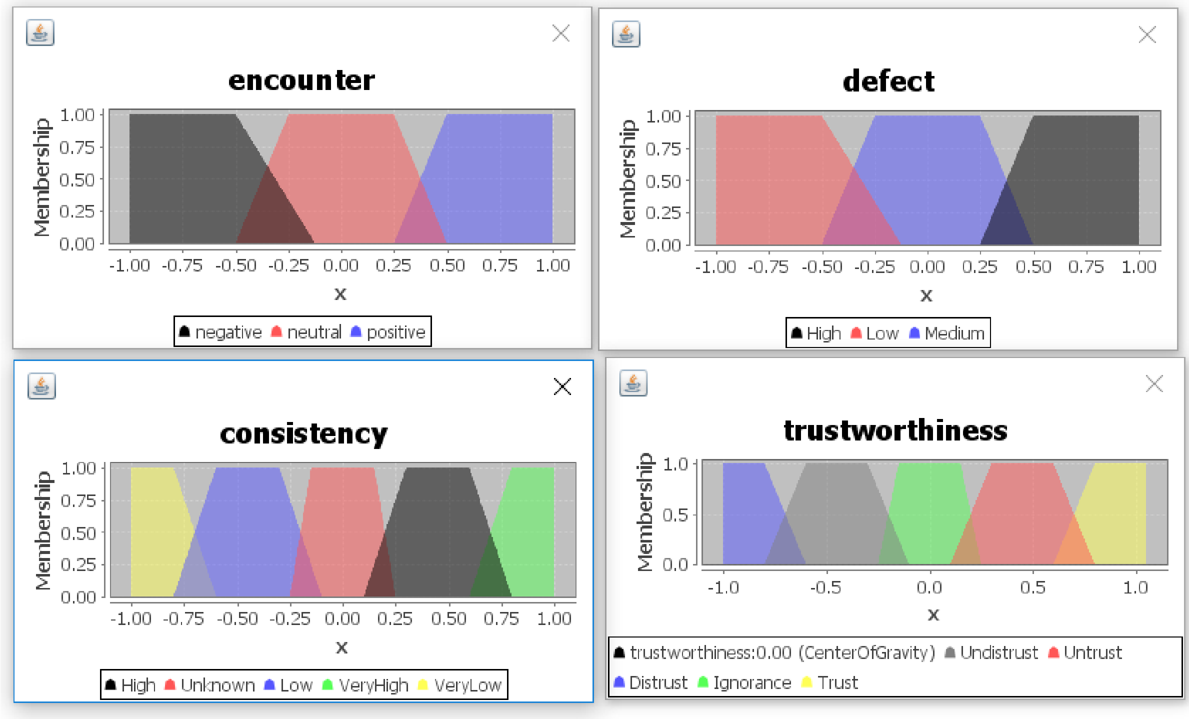
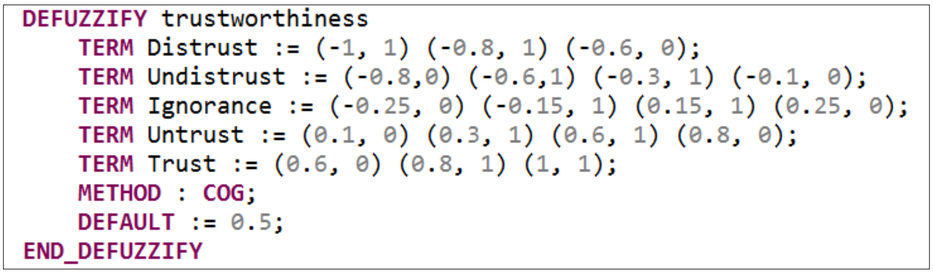


Figure 7-2: Examples of Membership Functions

The inference component is responsible for evaluating the rules. Rules consists of linguistic terms which resulted from fuzzification. The linguistic terms in the rules may be combined using the AND or the OR operator. The trust model uses the accumulation method that applies the maximum algorithm.

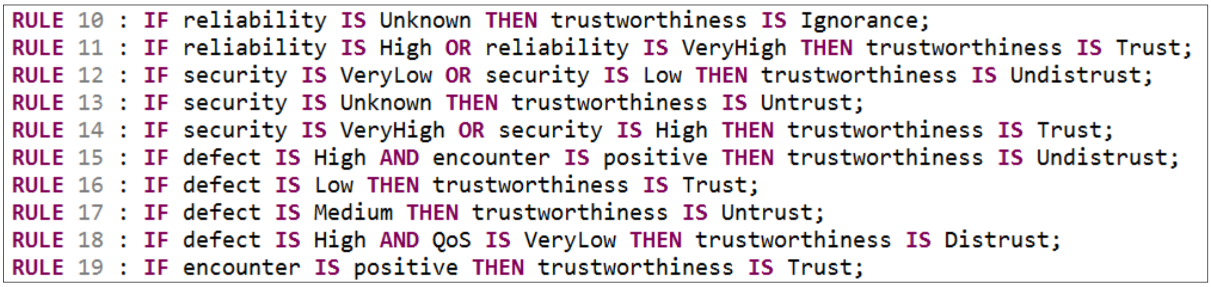
After inference, that last step of the FIS is defuzzification. This step converts the fuzzy value obtained from the inference component into a crisp output value in the range [-1, 1]. The proposed trust model uses the membership functions for trustworthiness shown in Figure 7-2. As mentioned in Chapter 6, the linguistic variable for trustworthiness are: Distrust, Undistrust, Ignorant, Untrust and Trust. Figure 7-3 shows the membership functions definition for trustworthiness.



```
DEFUZZIFY trustworthiness
    TERM Distrust := (-1, 1) (-0.8, 1) (-0.6, 0);
    TERM Undistrust := (-0.8,0) (-0.6,1) (-0.3, 1) (-0.1, 0);
    TERM Ignorance := (-0.25, 0) (-0.15, 1) (0.15, 1) (0.25, 0);
    TERM Untrust := (0.1, 0) (0.3, 1) (0.6, 1) (0.8, 0);
    TERM Trust := (0.6, 0) (0.8, 1) (1, 1);
    METHOD : COG;
    DEFAULT := 0.5;
END_DEFUZZIFY
```

Figure 7-3: Definition of the Defuzzification membership function

The Center of Gravity (CoG) method was used in the defuzzification process. The crisp trustworthiness value output from defuzzification process using the CoG method is based on the CoG of the trustworthiness fuzzy set. The last part of the FCL file consist of rule blocks. Figure 7-4 shows some of the rules that were used in the testing of the trust model.



```
RULE 10 : IF reliability IS Unknown THEN trustworthiness IS Ignorance;
RULE 11 : IF reliability IS High OR reliability IS VeryHigh THEN trustworthiness IS Trust;
RULE 12 : IF security IS VeryLow OR security IS Low THEN trustworthiness IS Undistrust;
RULE 13 : IF security IS Unknown THEN trustworthiness IS Untrust;
RULE 14 : IF security IS VeryHigh OR security IS High THEN trustworthiness IS Trust;
RULE 15 : IF defect IS High AND encounter IS positive THEN trustworthiness IS Undistrust;
RULE 16 : IF defect IS Low THEN trustworthiness IS Trust;
RULE 17 : IF defect IS Medium THEN trustworthiness IS Untrust;
RULE 18 : IF defect IS High AND QoS IS VeryLow THEN trustworthiness IS Distrust;
RULE 19 : IF encounter IS positive THEN trustworthiness IS Trust;
```

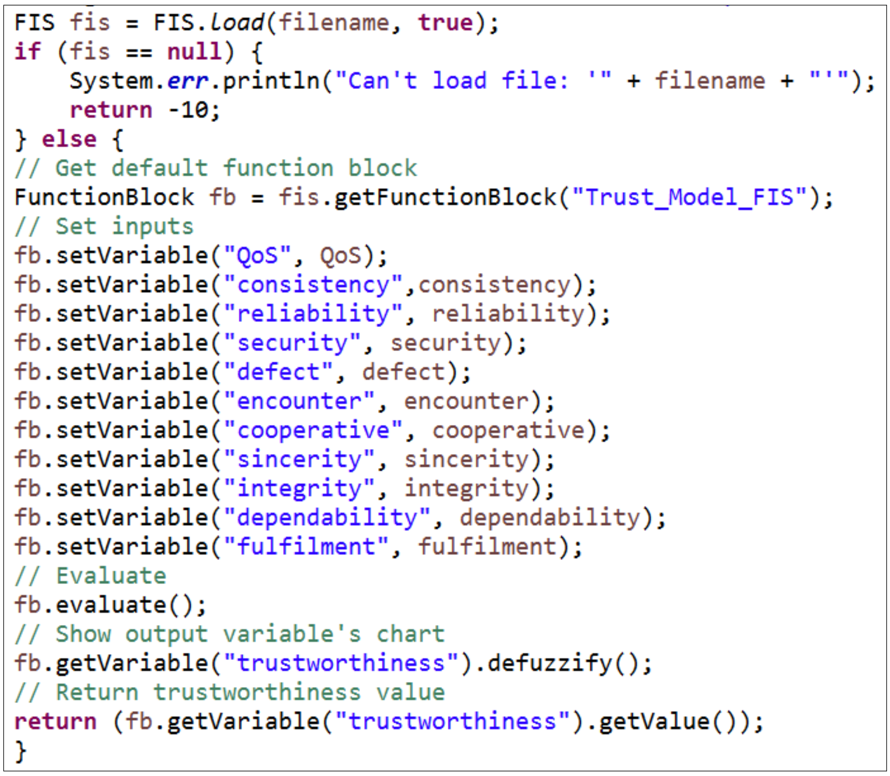
Figure 7-4: Sample Rules

The CoG methods outputs a crisp value that is based on the fuzzy set's centre of gravity. This is done by dividing the total area of membership functions. The defuzzified value is obtained by finding the summation of the area and the CoG of each sub-area. This is handled by the jFuzzyLogic. The rules are added to the FCL file. This completed the creation of an FCL file. The FIS takes in trust property values as input and outputs trust values. The following section describes the trust model application.

7.2.3 Trust Model Application

The agents were implemented using Java Agent Development Framework (JADE) which is an open source middleware that supports the efficient implementation of multi-agent systems (Bellifemine, Poggi, & Rimassa, 2001). The MySQL database was used as the repository for the trust model. Each trust agent had its own repository for trust information. Each trust agent had access to at least one root agent.

Each trust agent had access to the FCL files depending on the trust requirements. Figure 7-5 shows the code for accessing the FCL file from a trust agent. Apart from trust agents, entities that had computational capabilities also had access to the FCL files and they also had their own repositories for trust information.



```
FIS fis = FIS.load(filename, true);
if (fis == null) {
    System.err.println("Can't load file: '" + filename + "'");
    return -10;
} else {
// Get default function block
FunctionBlock fb = fis.getFunctionBlock("Trust_Model_FIS");
// Set inputs
fb.setVariable("QoS", QoS);
fb.setVariable("consistency",consistency);
fb.setVariable("reliability", reliability);
fb.setVariable("security", security);
fb.setVariable("defect", defect);
fb.setVariable("encounter", encounter);
fb.setVariable("cooperative", cooperative);
fb.setVariable("sincerity", sincerity);
fb.setVariable("integrity", integrity);
fb.setVariable("dependability", dependability);
fb.setVariable("fulfilment", fulfilment);
// Evaluate
fb.evaluate();
// Show output variable's chart
fb.getVariable("trustworthiness").defuzzify();
// Return trustworthiness value
return (fb.getVariable("trustworthiness").getValue());
}
```

Figure 7-5: Sample of Java code for evaluation of the trustworthiness of an entity

If an entity had computational capability, it had the option of carrying out its own trust computation and making its own decisions. However, trusting decisions for entities without computation capabilities were made by the trust agents.

7.2.4 Simulation Application

A Java application was implemented which was responsible for managing the simulated environment. The simulated environment consisted of 200 entities and a maximum of ten services. Each entity had predefined resources, trust requirements, and services it provides. Each entity provided a minimum of two services. This enabled the testing of trust values of the same entity in different contexts. The application was responsible for selecting the truster for each transaction. The application also kept track of the truster, trustee and the outcome of each transaction. The following section describes the simulated environment.

7.2.5 Simulated Environment Description

The simulation was carried out on a PC machine with 8GB of RAM and an i7 processor with 2.60 GHz. The environment consisted of honest entities, malicious entities, and selfish entities. Table 7-1 shows the parameters of the simulation of the IoT network. An average of 1200 transactions was carried out among the simulated devices.

Table 7-1: Simulation Parameters

Entities	
Number of Entities	200
Entities with computational ability	110
Malicious Entities	25
Selfish Entities	25
Agents	
Root Agents	4
Trust Agents	15
Malicious trust agents	3
Simulation Details	
Total Simulation Runs	9
Average Total Transactions Per Run	25 000

Selfish entities are entities that are not willing to cooperate with other entities in the network. In the simulated environment, malicious entities carried out denial of service attacks and provided false recommendations. The false recommendations include bad mouthing and good mouthing. Malicious entities also try to promote themselves by sometimes behaving like good entities. As mentioned in chapter 6, authentication for the proposed model is assumed to be handled by a trusted third party. Therefore, it was not considered in the simulated environment. To evaluate the performance of the proposed model, the simulation was run 9 times while taking note of its performance each time. Figure 7-6 shows an example of the simulated IoT environment.

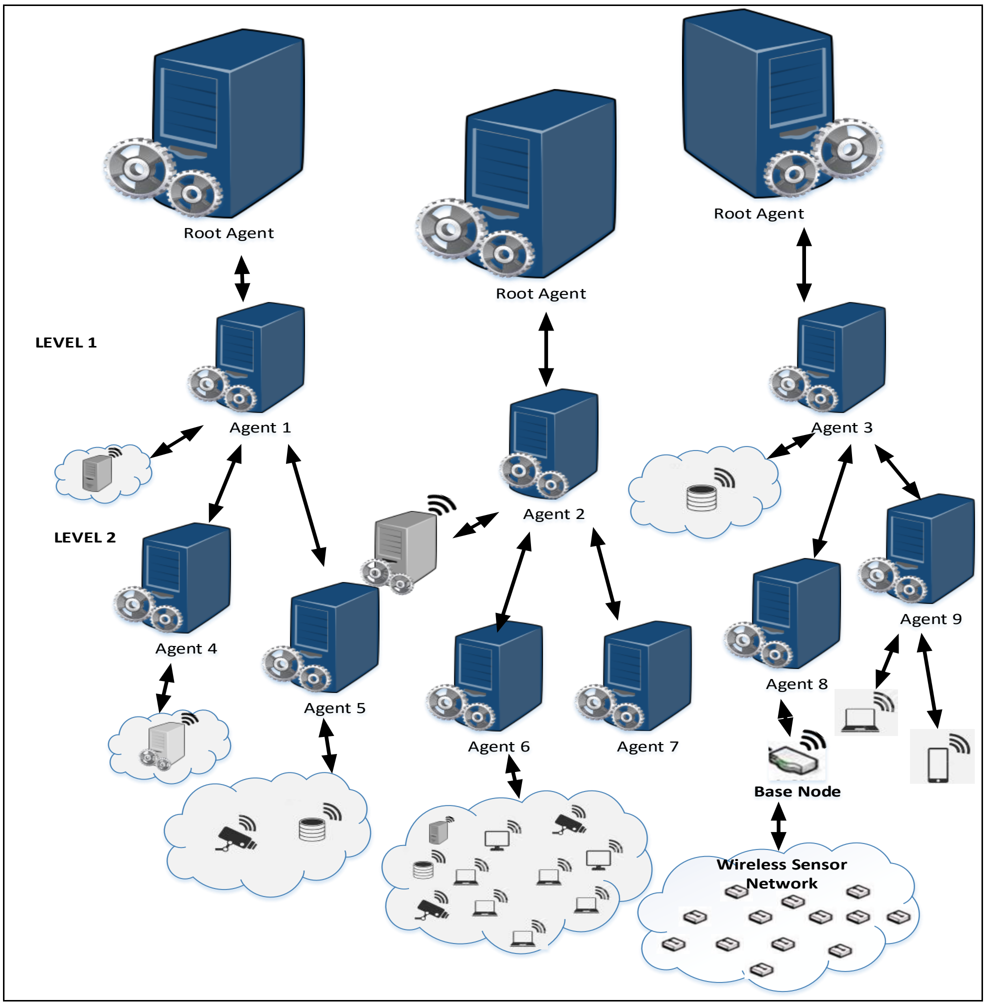


Figure 7-6: Simulated IoT Environment

7.3 Proposed Trust Model Testing

At the beginning, all the entities were assigned a trust value of zero. An entity that keeps its own trust records updates its trust value and properties values every time it interacts with the trustee and forwards its experience to the trust agent as well. It is up to the trust agent to either update the trustee trust value or carry out an investigation on both the truster and the trustee. The testing of the model began by testing the effects of competence and availability on trust value computation.

7.3.1 Effects of Competence and Availability on Trust Value Computation

The trust properties competence and availability were identified as major properties that affect the performance of an entity in a network. The first evaluation tests the accuracy of the trust model based on excluding either competence or availability. Figure 7-7 shows the effects that competence and availability have on the accuracy of the trust value. A trustee's trust value is based on accumulated observation of the trustee.

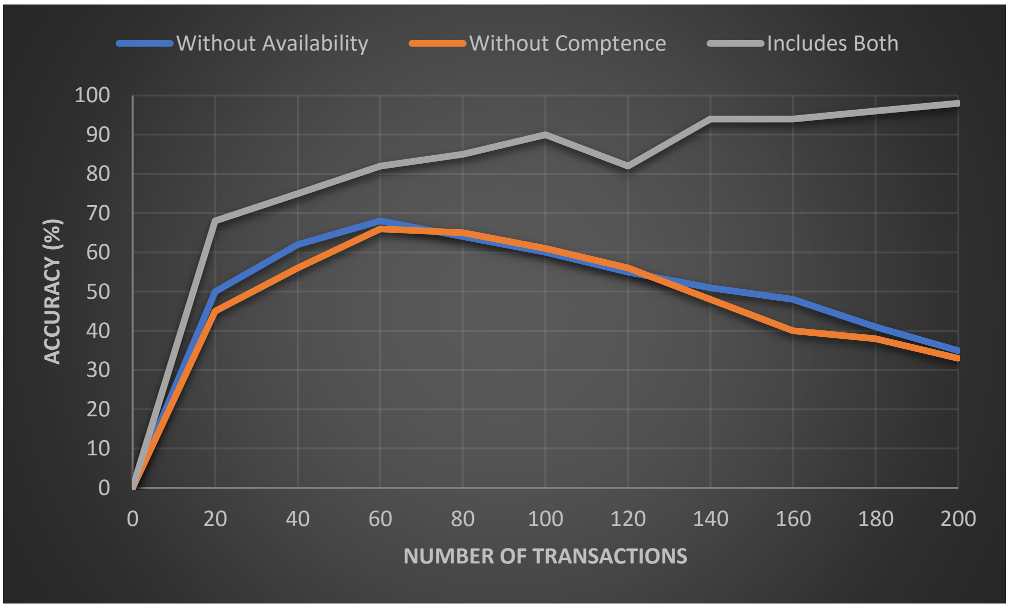


Figure 7-7: Effects of Competence and Availability

As the number of transactions increased, the accuracy when either competence or availability was not considered continued to decrease. Neglecting either competence or availability may cause trustworthy entities to be rated as untrustworthy. Based on the results obtained during this evaluation, I propose that before any request for transaction is made, the availability and competence of the trustee should be validated. Competence and availability were taken into consideration for all the other testing. The following section discusses the trust value fluctuation of all types of entities in the simulated environment.

7.3.2 Fluctuation of Trust Values

Of all the entities in the simulation environment, 75% were trustworthy. In the simulated environment, a malicious entity would give false recommendation, bad experiences and false feedback. Also, malicious entities worked together by giving good services and good recommendation to each other. Sometimes the malicious entities would disguise themselves as trustworthy entities by offering good services to other entities in the network. This attack is called contradictory behaviour attack. This kind of attack is sometimes difficult to detect. The model proposed in this research uses recommendation trust value and trust properties such as the consistency, predictability, persistence, defect and dependability to detect these kind of attacks. Figure 7-8 shows the fluctuation of the trust value of a trustworthy entity, a malicious entity and a selfish entity over 200 transactions.

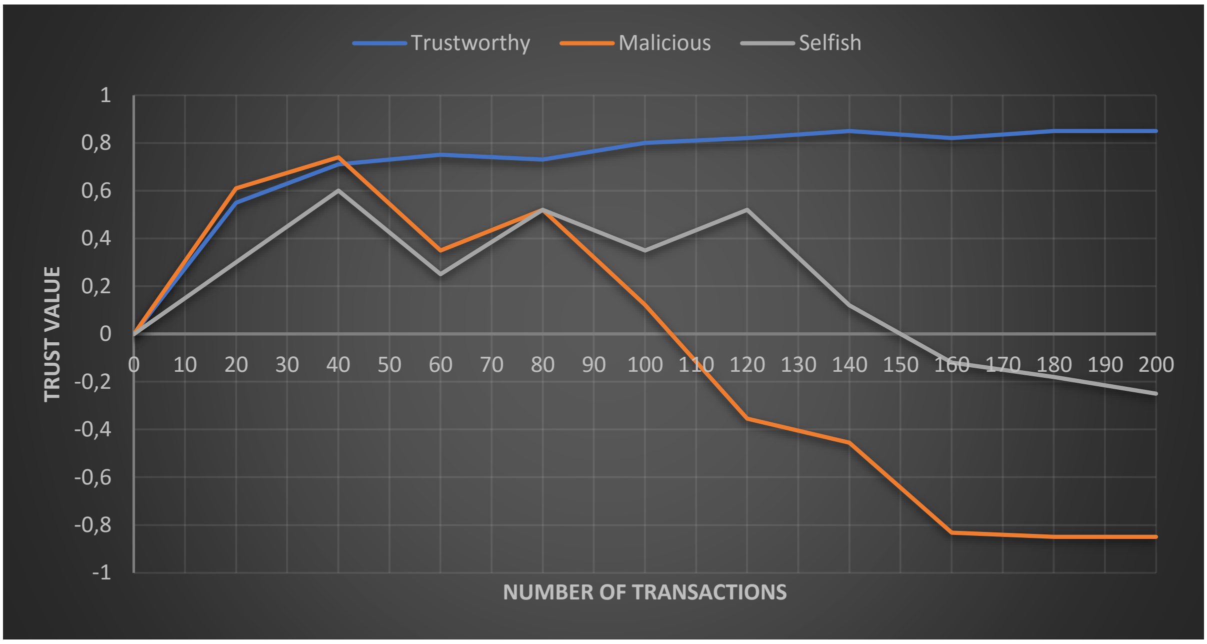


Figure 7-8: Trust value fluctuations

Malicious entities tried to masquerade as trustworthy entities by participating honestly at regular intervals in order to maintain a certain trust value. Results from testing the proposed model shows that even when a malicious entity tries to masquerade as a trustworthy entity it will eventually be identified. The trust value for a malicious entity will continue to drop until it falls into the distrust range. It took an average of 160 transactions for a malicious entity to be identified.

An entity is selfish if it does not want to cooperate with others even though it is competent and available. A selfish entity tries to conserve its own resources while benefiting from the other entities. It usually participates in communications or collaborations that will benefit it. In the proposed model, selfishness can be determined using trust properties that include predictability, willingness, incentive, and persistence. Figure 7-8 shows that the trust value for a selfish entity falls to the undistrust range but not total distrust. The trust values for the selfish entity and malicious entity fluctuate due to the entities masquerading as honest entities in some of the transactions. Over time the trust values become more stable.

In general, if the trust value of entities quickly reaches honest trustworthy values or the distrust values it means that the trust model may be prone to newcomer attack. On average our proposed model takes about 130 transactions to identify an entity as malicious or honest. The fluctuations of the trust values may be credited to the inconsistent behaviour of the entity. The fluctuations also show that it takes more transactions for the trust value to increase than for it to decrease. Even after an entity has shown good behaviour in the past, if it starts misbehaving its trust value will start to deteriorate. As a result, for a node to maintain a high trust value, it has to maintain a good behaviour. However, due to the uncertainty of the environment and the network, the trust value of an entity does not abruptly deteriorate to untrustworthy values.

7.3.3 Trust Evaluation Error

As mentioned earlier, at the beginning of the simulation all entities were given the same trust value. As transactions increase, malicious and selfish entities are identified as shown in Figure 7-8. In order to ensure the accuracy of the trust model, I also evaluated the error in the evaluation of the trust values among multiple entities regardless of context. The results obtained from the evaluation are illustrated in Figure 7-9. The evaluation was carried out over 200 transactions. Figure 7-9 shows that as the number of transactions increases, the error continued to decrease. Trust evaluation error is expected to drop as the number of transactions among the entities increases.

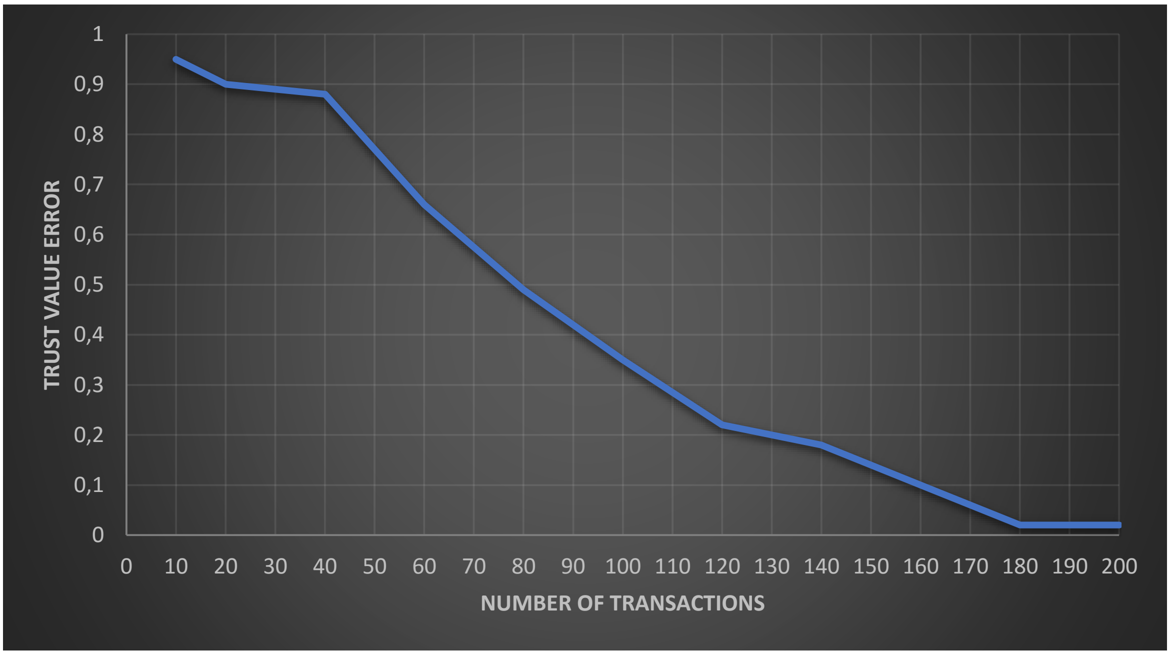


Figure 7-9: Average Trust Evaluation Error for Trustworthy Entities

7.3.4 Recommendation Accuracy

The accuracy of the recommendation provided by trustworthy entities was also measured. Figure 7-10 shows the average error of recommendation over 100 transactions. As expected, the error reduced as the number of transactions increases because the behaviour of the entities become more apparent as the number of transactions increases.

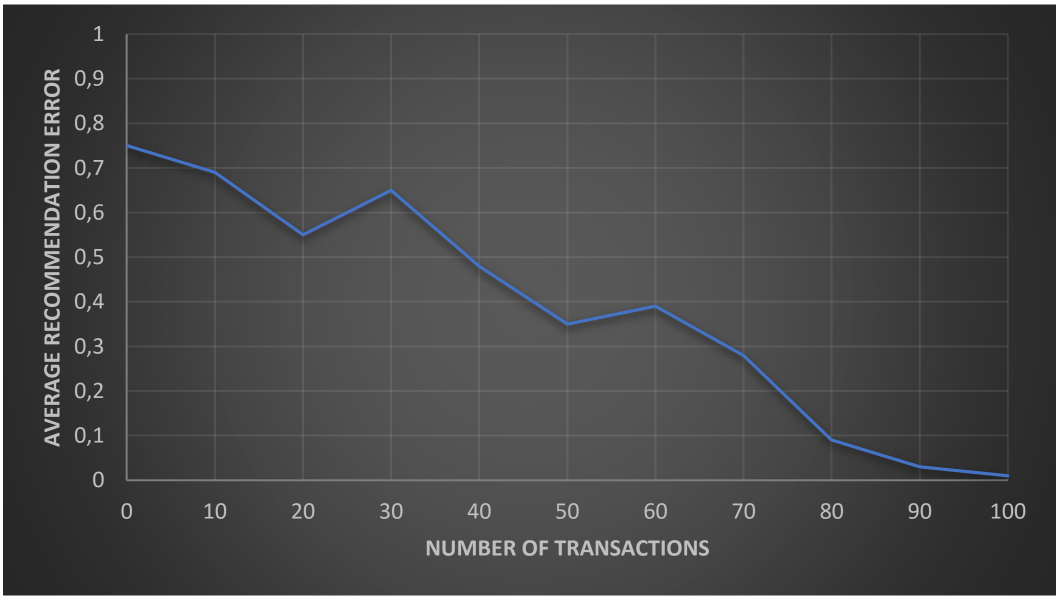


Figure 7-10: Recommendation Error

7.3.5 Convergence of Trust Values

During simulation, the convergence of trust values was also measured by taking into consideration the overall number of transactions among the entities. The experiment included both trust agents and entities. Trust values were considered to have converged if each trust value for all the trusters for the same trustee falls in the same linguistic variable (distrust, undistrust, ignorance, untrust and trust) even though they might not be exactly the same.

Figure 7-11 shows how the trust values converged over 200 transactions. At the beginning, the convergence is 100% because all entities were awarded the same trust value. The continuous drop of convergence percentage between 0 and 60 transactions can be attributed to computation error and oscillating behaviour of malicious entities.

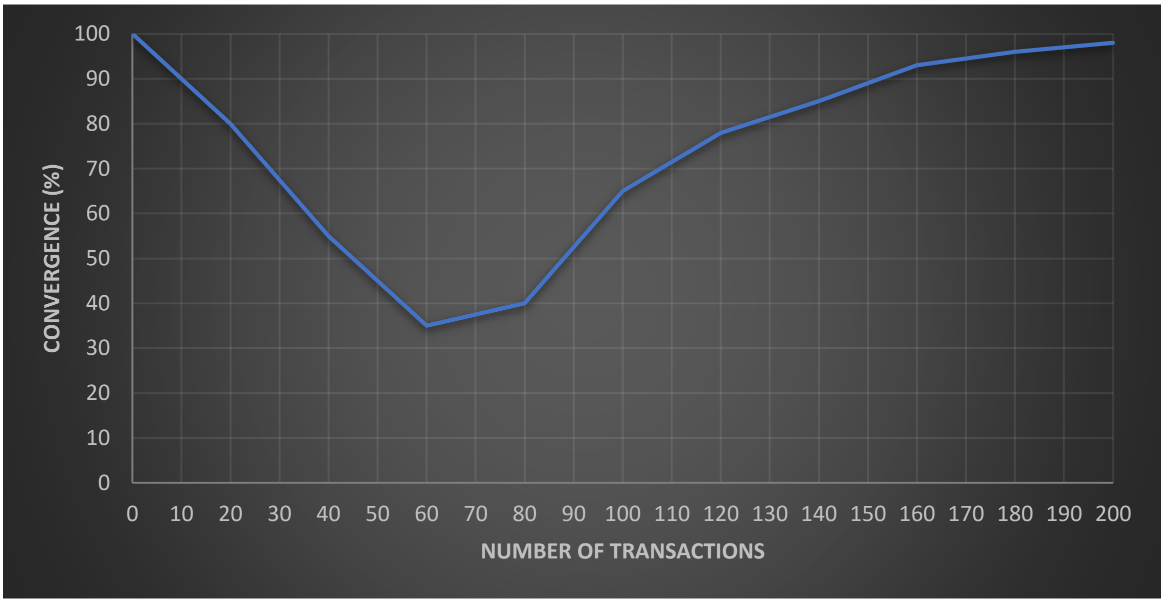


Figure 7-11: Trust Convergence Percentage

7.3.6 Unstable Behaviour

The last experiment in this section focused on a malicious node with oscillating behaviour. This experiment tested the benefits that an entity might get when it mixes good and bad behaviour. The entity behaves honestly until it is trusted and then behaves badly but not enough to be distrusted. The experiment recorded the trust values of the malicious entity over 200 transactions. Figure 7-12 shows the oscillating behaviour of the entity.

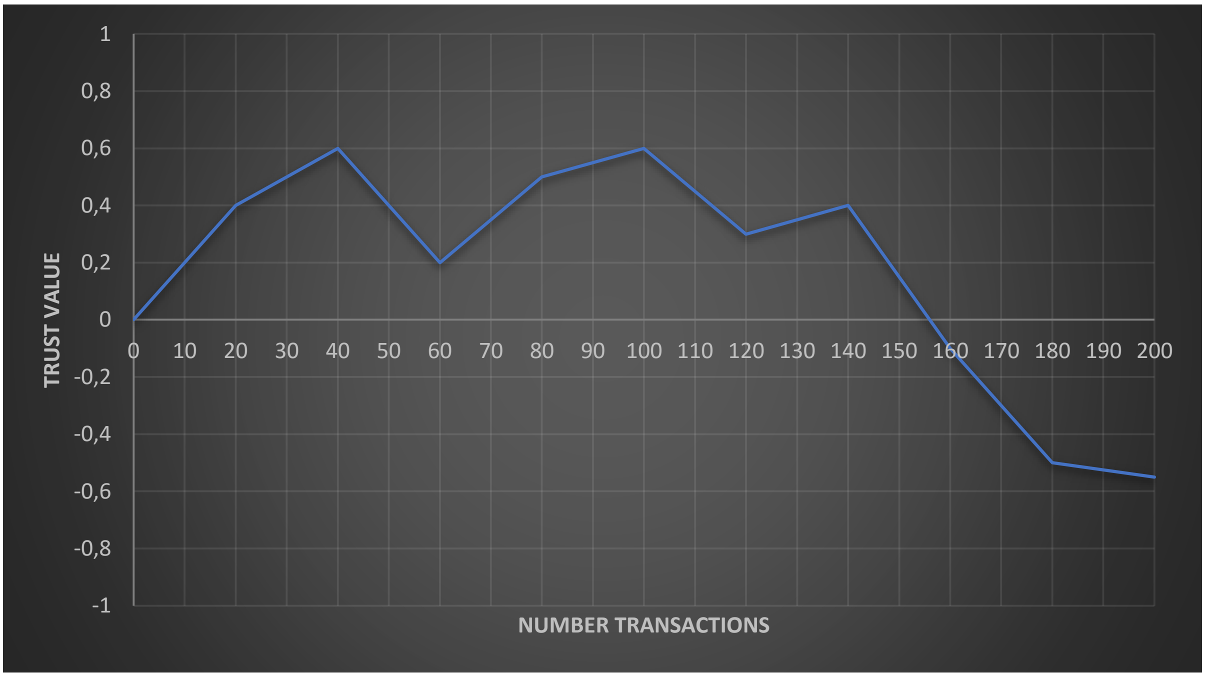


Figure 7-12: Oscillating Behaviour of a Malicious Entity

The results show that it took an average of approximately 180 transactions for the entity to be identified as malicious. The delay prevents entities from being mistrusted. However, malicious entities may take advantage of this window period.

7.3.7 Reputation Component Evaluation

All entities begin with a reputation value of zero. The trust agents keep track of the reputation of all the entities in their respective domains. Of all the entities in the simulation environment, 25% were not honest. The first test evaluated the performance of the environment with and without the reputation component. The results are shown in Figure 7-13. The results show that the rate of successful transactions continues to decrease when the reputation was not included. The fluctuation of the values can be attributed to the oscillating behaviour of malicious entities.

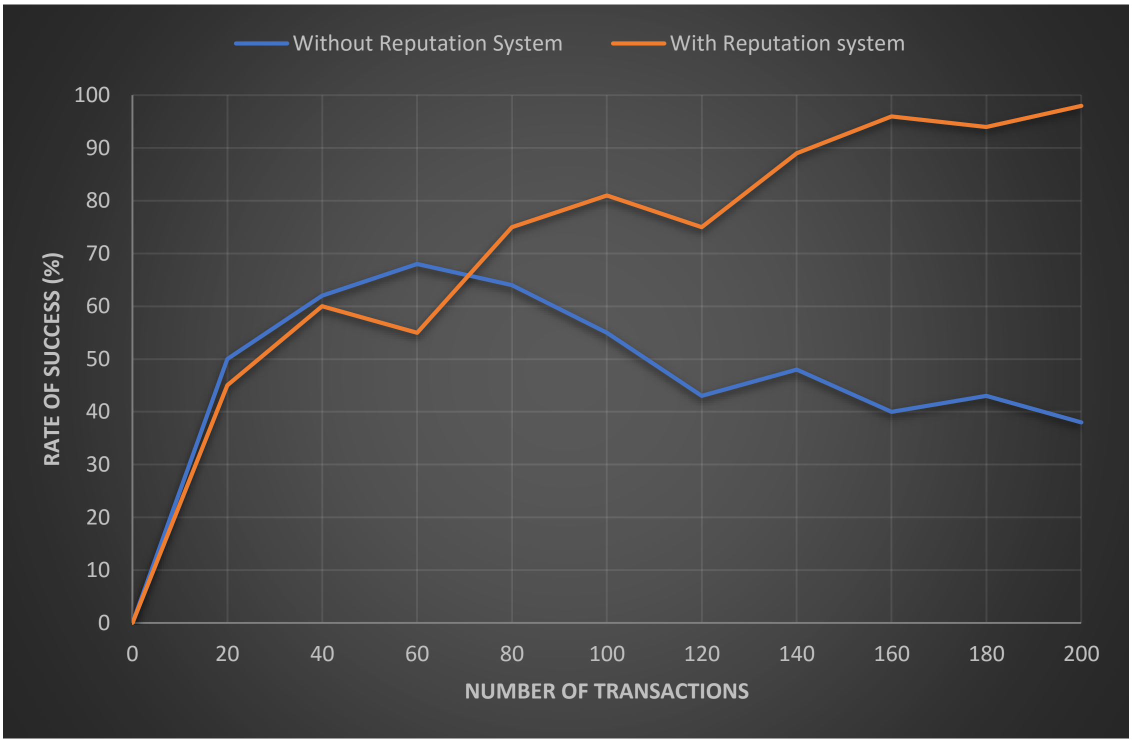


Figure 7-13: Effects of reputation on the environment

Figure 7-14 shows the error that occurred when computing reputation values over 200 transactions. As expected the error continued to decrease as the number of transactions increased.

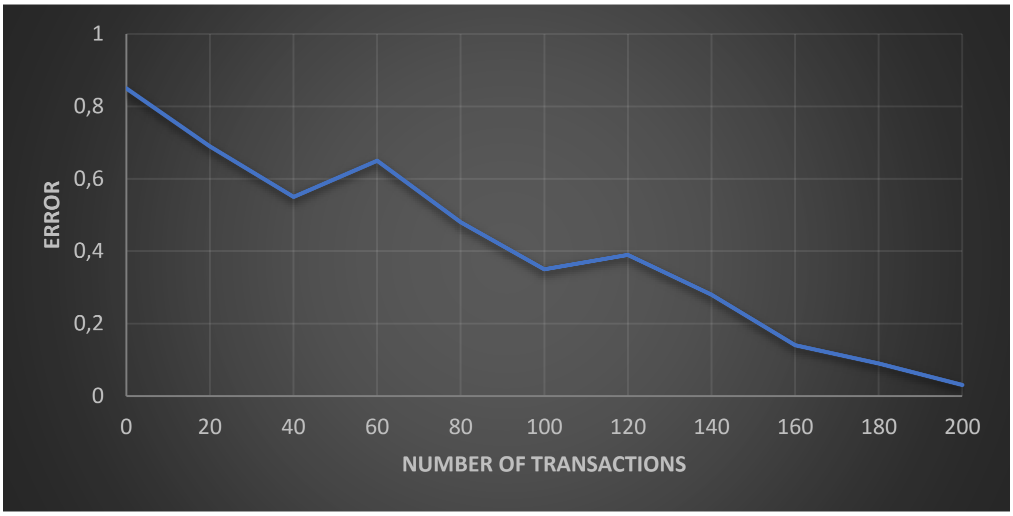


Figure 7-14: Reputation computation error

The convergence of the reputation values among trust agents was also taken into consideration during testing. Figure 7-15 shows the convergence graph over 200 transactions. The initial drop of the convergence can be attributed to both computation error and oscillating behaviour of malicious entities.

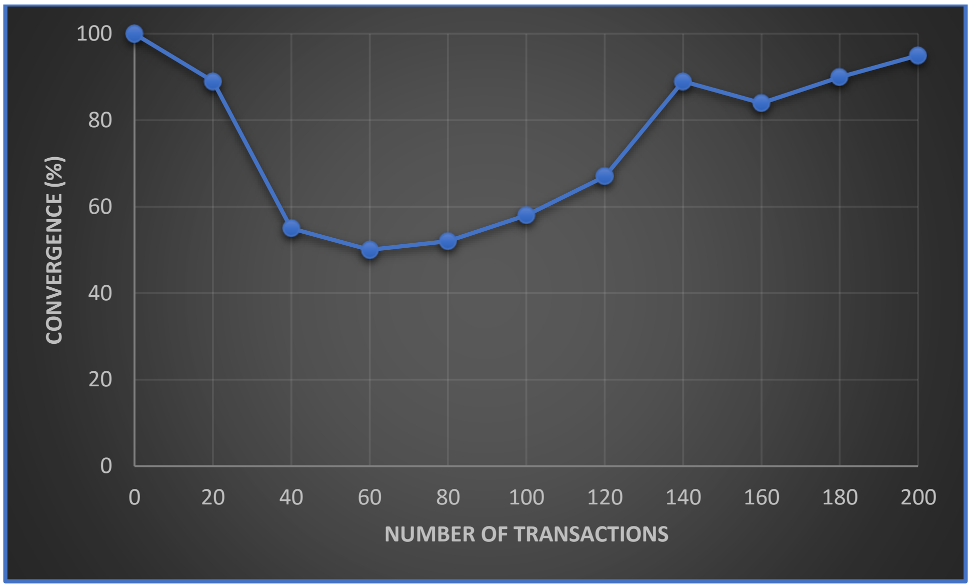


Figure 7-15: Reputation convergence percentage

All the results from the simulation show that the proposed trust model can effectively identify malicious entities in the simulated environment. This concluded the first part of the testing of the proposed model. The following section experiments the effects of distrust on the trust mode. As mentioned in Chapter 5, the effects of distrust will be tested by treating distrust as the negation of trust.

7.4 Testing the effect of Distrust on the trust model

This section tests the effects of distrust on the proposed trust model by treating distrust as a negation of trust. Distrust is the result of a consistent unacceptable behaviour. It follows that if distrust is the negation of trust then it inherits all the characteristics of trust. This led to a few modifications of the FIS to suit these new changes. The following section discusses the major changes that were made to the FIS.

7.4.1 Modification of the fuzzy inference System

The FIS was modified to treat distrust as the negation of trust in order to test the effects of distrust. (Nafi, Kar, Hossain, & Hashem, 2012, 2013) proposed a trust model that took into consideration trust without distrust. In their proposed model, they proposed linguistic variables as very low, low, average, high and very high (Nafi et al., 2012, 2013). For this evaluation, I adopt these variables with the exception of average which I replaced with ignorance. Therefore, the linguistic variables for trustworthiness were modified to VeryLow, Low, Ignorance, High and VeryHigh. Table 7-2 gives the description of each of the new linguistic variable.

Table 7-2: Description of Linguistic Variables for Trustworthiness

Trust Level	***Description***
VeryLow	Very untrustworthy
Low	Untrustworthy
Ignorance	Unknown
High	Trustworthy
VeryHigh	Very trustworthy

The linguistic variables of all the trust properties and their membership functions remained the same. The membership function of trustworthiness change to match the description in Table 7-2. Figure 7-16 shows the updated membership functions for trustworthiness.

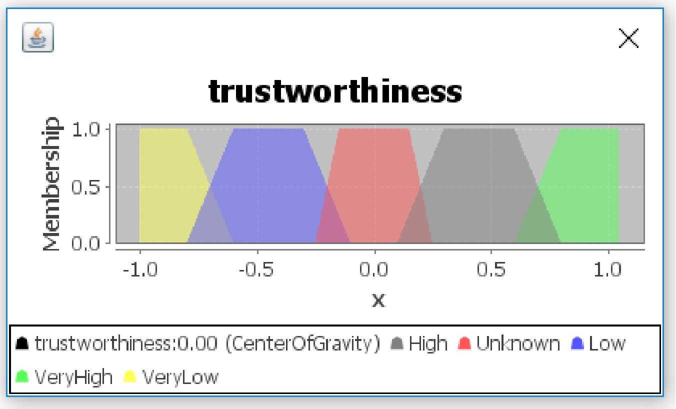


Figure 7-16: Membership functions for Trustworthiness

Changes were also made to the rule base to suit the changes made to trustworthiness. However, the defuzzification process remained unchanged.

7.4.2 Effects of Distrust on malicious and selfish entities

This section tests the effects that distrust has on malicious and selfish entities over 200 transactions. The experiment tested the prediction of distrust based on trust only. Figure 7-17 shows the fluctuations of the malicious and selfish entity.

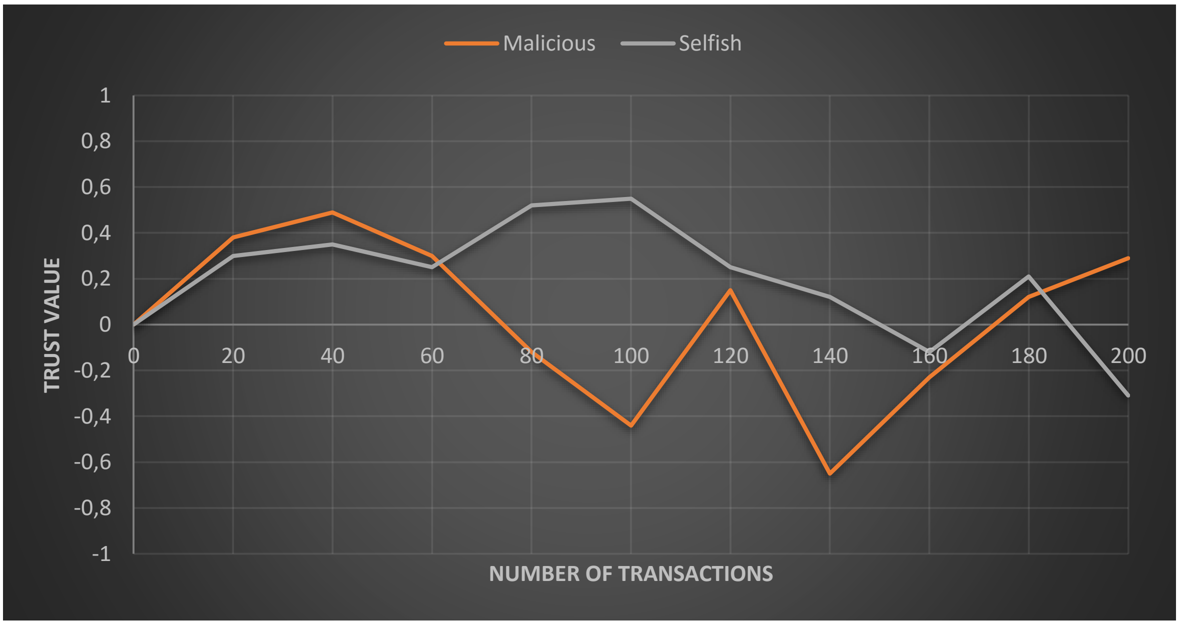


Figure 7-17: Trust values for malicious and selfish entities

Figure 7-17 shows that the malicious entity was able to regain trust after it had been identified as malicious. This could be attributed to the fact that all the linguistic variables of trustworthiness are context specific. This allows the entity to be selected for other transactions in a different context. This means that if the transaction is successful the malicious entity can receive a positive feedback which can raise its trust value.

7.4.3 Trust evaluation error

Trust evaluation error is the error that occurs after the trust value of an entity has been computed. This error measures the accuracy of the trust model. The error margin is supposed to continue to drop to zero as the number of transaction increases. Figure 7-18 shows the trust evaluation error over 200 transactions.

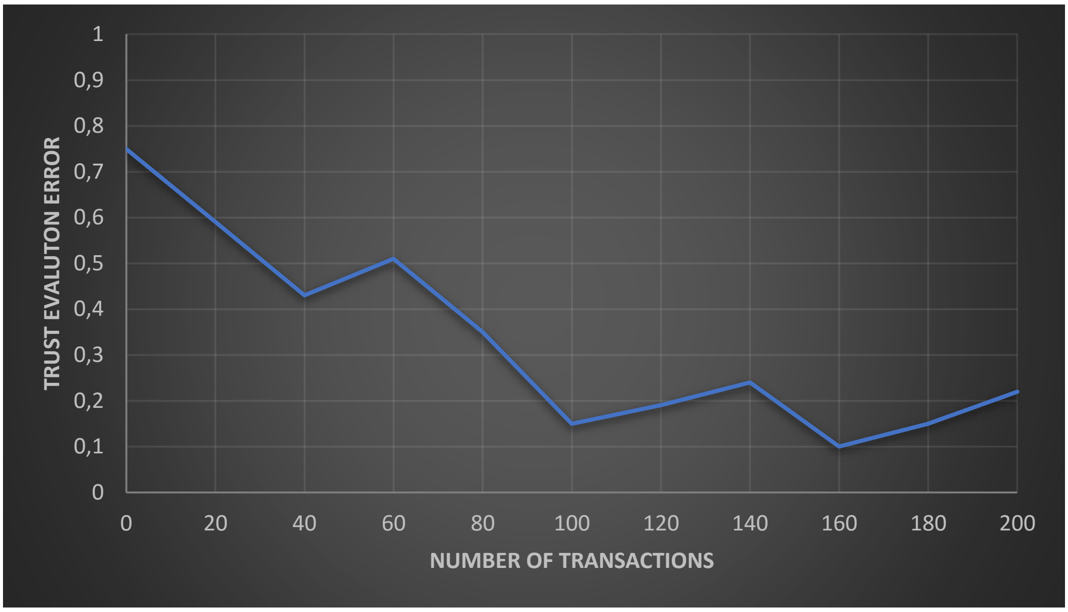


Figure 7-18: Trust Evaluation Error

The error was expected to continue to drop because the total number of entities in the simulation environment remained constant. However, the trust evaluation error continued to fluctuate. This shows that distrust plays a vital role in the trust model. Based on these results the research concludes that distrust is a form of negative trust but not the negation of trust.

7.5 Discussion

The experiments above show that distrust is separate from trust and treating it as a negation of trust may cause malicious entities to be trusted. Distrust and trust exist simultaneously together in the same environment. Distrust, unlike trust, is not context specific, a distrusted entity will approach all situations in a malicious manner. The results from the testing show that distrust enables entities to avoid malicious entities automatically by excluding distrusted entities in transactions and recommendation. I conclude that distrust excluding trust has a negative effect on a trust model.

Trust is only one element of trustworthiness. In order to accurately estimate trust, it is imperative to also take into consideration distrust. In addition, untrust and undistrust enable sufficient information to be collected before the decision to either trust or distrust is taken.

The number of observations signifies the amount of evidence collected. During the initial stages of evidence collection, the model is more sensitive. As the number of interaction increases, the trust value will begin to represent the true behaviour of the entity.

Uncertainty and vagueness of trust need to always be taken into consideration during trust computation. Such considerations are shown in trust computation by the progression of the trust value. The proposed model takes into consideration the behaviour of both the trustee and the truster. It is also important for the model to handle imprecise data.

7.6 Conclusion

This chapter analysed numerical results obtained during the simulation of the proposed model and the IoT environment. Simulation for agents was done using JADE and the fuzzy logic system was implemented using the jFuzzyLogic library. The simulation results showed that the proposed trust model can effectively identify malicious entities. The model was able to deal with different types of behaviours of entities. The testing proved that the proposed trust model can support decision making in IoT based on trust.

I am are in agreement with (Guha, Kumar, Raghavan, & Tomkins, 2004) that distrust significantly affects the propagation of trust. There is a need to collect ample evidence until an entity is distrusted because the decision is not reversible. In conclusion, the results from all the experiments show that the proposed trust model can reduce malicious activities in the IoT environment.

8

Conclusion and Future Work

8.1 Introduction

This chapter concludes the research by summarizing and highlighting the accomplishments of this research. The chapter also gives details of research areas that are still open.

8.2 Discussion

This research proposed a trust model for the IoT that uses fuzzy logic to compute trust. The evaluation of trust in the proposed trust model emanates from the human experience of trust. The proposed model is a hybrid of both the centralised and the distributed systems. This ensures the scalability of the model while enabling entities that lack computational capability to obtain trust values from the centralised component. Trust computation in IoT assists entities to estimate the likelihood of either a trustee's good intentions or malicious intentions. This is done through the use of personal past experiences in similar situations or recommendations. The proposed trust model has the following advantages:

- It is lightweight
- It is scalable, flexible and dynamic
- Supports the multi-dimensional element of trust
- Enables the building of trust networks
- Makes trust values for each entity in the IoT environment available for all the entities in the network

Trustworthiness is based on vague past experiences, opinions, and reputation. The model uses the fuzzy logic to compute the trustworthiness of the entities. Fuzzy Logic can tolerate imprecise inputs which makes it capable of handling uncertainty. Since trust is a human notion, it follows that is should be computed by mathematical approaches that are able to handle natural languages. Also, as each human is capable of estimating trust based on individual requirements, the proposed model support calculation for trust based on individual requirements and experiences. This section summarizes each of the chapters in this research.

Chapter 1 gives the description of the aims and the objectives of the research. Chapter 2 discussed the research design and research methodology. The chapter began by discussing the philosophy of computer science in order to highlight the position of computer science in the field of research and describe the views of the researchers on research. In this discussion, the researcher highlighted that computer science borrows from different fields. This leads computer science research to combine design, practice, and theories from different disciplines. This means research in computer science needs to have a theoretical basis and an experimental design. The discussion went on to highlight computer science research paradigms and stated that this research falls under the scientific paradigm. The chapter concluded by discussing the methodology followed in this research.

This research proposed a trust management model specifically for the IoT environment. The designing of the model began by discussing the concept of trust in chapter 3. The chapter began by discussing definitions of trust relevant to this research. This research adopted the definitions by (Gambetta, 2000) and (Dasgupta, 1988). The chapter also discussed the properties and the characteristics of trust relevant to IoT. The most important properties for IoT were identified to be integrity, competence, predictability, QoS, CoI, reliability, risk, consistency, sincerity and commitment. Chapter 3 also discussed the relationship between trust and reputation. The chapter highlighted that reputation can be used to enhance trust and aids in the creation of trust relationships.

I concluded the chapter by discussing the relationships between trust and distrust. The discussion included the details of the relations of trust and distrust. In terms of relations, I concluded that trust can only be considered to be partially transitive when taking into consideration recommendation. Otherwise, trust is not transitive. The importance of distrust in a trust model was highlighted in chapter 3. In this research, the researcher agrees with (S. Marsh & Dibben, 2005) that distrust is not the negation of trust but a form of negative trust. Even though a couple of years has passed, I still agree with (Tang et al., 2014) that there is still little research on the effects of distrust on trust models. Therefore, there is still a need to carry out more research on distrust in modelling trust.

The current state of the IoT was reviewed in chapter 4. The chapter highlighted the objectives of a trust management model, the limitations of the current models and the components which are required in an IoT trust model. Taking into consideration the objectives highlighted in chapter 4, it was discovered that trust models could be built incrementally and the objectives could transform into components in the trust model. The researcher identified the following objectives as the basis of a trust model:

- Trust Relationship and Decision
- Data transmission and communication trust
- Generality

Therefore, these are the main objectives that were included in the proposed model. QoS was identified as part of the properties of trust computation and we added it to the model as one of the trust composition elements. Of the highlighted components, the trust specification language and authentication were left out in the designing of the proposed model. The researcher is planning on adding the trust specification language as part of the future work. The researcher proposed that the authentication of entities be carried out by trusted third parties. Limitations addressed in this research include lack of consideration for distrust and lack of relationship evaluation methods.

Chapter 5 discussed the computation methods that were considered for the proposed trust model. The reviewed methods included game theory, Bayesian networks, fuzzy logic, weighted summation and neuro-fuzzy system. It was highlighted that weighted summation has the challenge when it comes to selecting adequate weight factors. For game theory, the researcher is in agreement with (S. P. Marsh, 1994) that it is not adequate to handle trust. The problem with Bayesian networks is that the probability values used in the network are based on repeatable experiments (Abdul-Rahman & Hailes, 2000). Also in probability, an entity can only belong to a particular set. These problems that were raised in regard to Bayesian networks can be solved by fuzzy logic. Fuzzy logic is suitable for trust computation because it imitates the human mind. Unlike probability, fuzzy logic enables entities to belong to multiple sets at the same time. Fuzzy logic estimates the degree of truth while probability estimates the probability of truth based on repeatable experiments. In fuzzy logic the transition from of membership is gradual and this enables the trust values for entities to transitions between membership sets. Neuro-fuzzy was deemed not suitable because of lack of sufficient data for training the neurons. Therefore, ultimately fuzzy inference system was selected for the proposed model.

After selecting the appropriate computation method, a model suitable for the IoT was designed in chapter 6. Chapter 6 gives a detailed description of the proposed model which includes the description of the FIS. The FIS is responsible for the calculation of the trust value. The FIS takes in trust properties as input and output a single output which is trustworthiness. The membership functions of the input values depend on the trust property. The membership functions of the output value were defined as distrust, undistrust, ignorance, untrust and trust. Distrust was defined as a form of negative trust in the proposed model. The description of how trust relationships can be created was included in chapter 6. Chapter 6 also included the description of how reputation and recommendation are computed.

Chapter 7 details the evaluation of the feasibility of the proposed trust model. The evaluation was carried out in a simulated environment. The trust model was implemented in Java. The FIS was implemented using the jfuzzyLogic library which is an open source library created by (Cingolani & Alcalá-Fdez, 2013). The trust agents and the root agents were also implemented using JADE. The simulation environment was managed by a Java application. The application was responsible for managing the environment and selecting the truster for each transaction.

The simulated environment consisted of honest entities, malicious entities, and selfish entities. The initial tests that were carried out tested the effects of competence and availability on the trust model. The results that were obtained showed that these two properties need to be validated before any request for transaction is made. In the simulated environment, 75% of the entities were trustworthy and the remaining 25% included both malicious and selfish entities. The second tests carried out on the model monitored the fluctuation of trust values of three random entities (one was malicious, the other one was selfish and the last one was trustworthy) over 200 transactions. The tests showed that the model took an average of 130 transactions to identify an entity as either malicious or honest. The results from these tests showed that the model was able to accurately identify malicious and honest entities.

The next tests done on the model tested the accuracy of the model over 200 transactions as well. The results from the tests showed the trust evaluation decreased as the number of transactions increased. The accuracy of the recommendations was also tested over 100 transactions. The results showed that even though the recommendation fluctuated, it continued to drop as the number of transactions increased. The fluctuations could be attributed to the contradicting behaviour of the malicious entities and selfish entities. The convergence of the trust values was also evaluated. Trust values were considered to have converged if each trust value for all the trusters for the same trustee falls in the same linguistic variable. The results showed that it took an average of 190 transactions for the convergence percentage to rise to about 96%.

One of the malicious entities in the simulated environment had an oscillating behaviour. The trust values for this entity were recorded over 200 transactions. The results showed that it took the model an average of 180 transactions to identify the entity as malicious. These results proved that the model is able to identify malicious entities even when they have oscillating behaviours.

The last evaluations done on the model tested the effects of distrust on the trust model. This was done by treating distrust as the negation of trust. The effects of distrust on a selfish entity and on a malicious entity over 200 transactions. The results show that the selfish was identified after an average of 190 transactions but the trust value of the malicious entity continued to fluctuate. The fluctuations may be attributed to the fact that trust is context specific. The trust evaluation error of the modified model was evaluated. The results showed that the trust error continued to fluctuate. This was attributed to the fluctuation of the trust values of malicious entities. The results also showed that it is important to include distrust a trust model. However, there is still a need to carry out more research on how distrust can be modelled. In conclusion, the results of the simulation showed the proposed model is able to effectively identify malicious entities/nodes.

Table 8-1 gives a summary of how each of the research questions and objectives were addressed in this research.

Table 8-1: Research Outcome

Research Question	**Objective**	**Research Outcome**
What are the main trust properties suitable for IoT?	Identifying trust properties suitable for IoT	Chapter 3 highlights all the properties of trust that are suitable for IoT
How can trust be best managed in an IoT environment?	Identifying limitations of current models	The limitations of the available trust models were identified in Chapter 4. Chapter 4 also highlighted the objectives and main components of a trust management model for IoT.
What are suitable trust computation methods for IoT?	Identifying computation method suitable for IoT	Chapter 5 reviewed different computational methods that are relevant for trust computation. The chapter concludes by justifying the method which was selected the proposed model.
What are the trust relationships available in the IoT environment? *How can new trust relationships be created in IoT?*	Creating new trust relationships that are suitable for the IoT environment	A description of how new trust relationships can be forged in IoT is given in Chapter 6.
How can trust be managed for IoT things with limited resources in terms of power	Designing of an architecture that is suitable for the IoT environment	Chapter 6 gives a detailed description of a trust model architecture that is suitable for IoT.

and computational capability?		
What is the suitable propagation method for IoT?	Identifying trust propagation method suitable for IoT	Chapter 6 also includes a description of the proposed propagation method for IoT.
What is the value of distrust in the IoT environment?	Investigation the importance of distrust in IoT	Chapter 7 details the evaluation of the feasibility of the proposed trust model. It also includes the evaluation of the effects of distrust in the model

This research demonstrated that trust management in IoT can be provided as an aid to other security measures. In trust management, it is important for the model to be able to collect and validate trust data. This data can be used to modify the model. After the collection of significant amounts of trust data over time, I propose that the computation model for the proposed trust model be implemented using the Neuro-Fuzzy method. This will enable the model to adapt to the ever-changing IoT environment. The Neuro-Fuzzy method will equip the model with learning mechanisms. The learning mechanisms can be used in determining the weights of the properties in trust calculations. In the IoT environment, the following need to be guaranteed:

- Data enormity
- Confidentiality
- Authentication and authorization

Components that guarantee these elements can be added to the trust model. The following section gives a brief discussion on the effects of distrust in trust management.

8.3 Effects of Distrust on a trust model

The absence of trust does not necessarily mean distrust. Lack of information or evidence may cause the absence of trust but this is not distrust. In this research, I came to the conclusion that distrust means being convinced that an entity is malicious based on evidence. The researcher also identified that distrust can be used to separate selfish nodes from malicious nodes. The researcher concludes that distrust is not context specific. Once an entity is identified to be malicious, it is malicious in every other context.

8.4 Future Work

Future work includes testing the model in a real IoT environment in order to validate the results from the simulated environment. In the IoT environment, there are some factors that are unpredictable such as an increase in network traffic that can affect the transaction among IoT entities. The researcher acknowledges that such factors can affect the outcome of trust computation. Such factors may be identified and be taken into consideration when updating trust.

Policies are important for organisations and are included in determining trust in businesses. Policies govern transactions among business entities. Some of these policies need to be included in defining trust requirements and making decisions. Therefore, as part of the future work; there is a need to include policies that affect trust computation in the model. Policies can be included as rules in the trust model.

The performance of the model largely depends on the linguistic variable of the fuzzy sets and the rules. As part of the future work, the researcher plans to explore the effects of different rules and fuzzy sets. The researcher also plans to add machine learning to the model with the aim to enhance trust computation and the performance of the model.

Lastly, time consumption and accuracy of trust evaluation are very important in a trust model. The testing carried out for the proposed trust model did not take into account time consumption due to the setup of the simulation environment. Therefore, there is still a need to carry out time consumption in the future testing of the proposed model.

8.5 Conclusion

When humans make the decision to trust, they don't explicitly assign a numerical value to the trust measure. This makes trust to be fuzzy. Policies that may be enforced when calculating trust may also be fuzzy. Therefore, when modelling trust it is important to take this into consideration. This makes fuzzy logic suitable for computational trust. Fuzzy logic helps trust reasoning by IoT devices to be closer to the human way of reasoning.

To demonstrate the feasibility of the proposed model, the researcher simulated an IoT environment consisting of devices with different capability and analysed the trustworthiness values obtained. The testing of the model proved that the model can effectively identify malicious entities and selfish entities. The evaluation of the proposed trust model also proved that trust can support decision making in IoT and enable the creation of trust networks. The notion of trust networks in IoT will enable entities to easily forge trust relationships and this will make trust evaluation more accurate.

References

Abdul-Rahman, A., & Hailes, S. (1997). A distributed trust model. In *New Security Paradigms Workshop* (pp. 48–60). ACM Press. https://doi.org/10.1145/283699.283739

Abdul-Rahman, A., & Hailes, S. (2000). Supporting trust in virtual communities. In *33th Hawaii International Conference on System Sciences, Hawaii, USA*.

Abraham, A. (2001). Beyond Integrated Neuro-Fuzzy Systems: Reviews, Prospects, Perspectives and Directions. In *Seventh International Mendel Conference on Soft Computiong* (pp. 376–372). Victoria, Australia: Brno, MENDEL 2001, Matousek Radek et al (Eds.).

Alam, S., Chowdhury, M. M. R., & Noll, J. (2011). Virtualizing Sensor for the Enablement of Semantic-aware Internet of Things Ecosystem. *International Journal of Design, Analysis and Tools for Cicuits and Systems*, *2*(1), 41–51.

Alnasser, A., & Sun, H. (2017). A Fuzzy Logic Trust Model for Secure Routing in Smart Grid Networks. *IEEE Access*, *5*(August), 17896–17903. https://doi.org/10.1109/ACCESS.2017.2740219

Ashton, K. (1999). That “Internet of Things” Thing. *RFID Journal*.

Atzori, L., Iera, A., & Morabito, G. (2010). The Internet of Things: A survey. *Computer Networks*, *54*(15), 2787–2805. https://doi.org/10.1016/j.comnet.2010.05.010

Avison, D. E., & Fitzgerald, G. (2006). *Information Systems Development: Methodologies, Techniques and Tools* (4th Editio). McGraw-Hill, Maidenhead.

Azzedin, F., & Maheswaran, M. (2002). Evolving and Managing Trust in Grid Computing Systems. In *IEEE Canadian Conference on Electrical & Computer Engineering (CCECE ’02)*. Winnipeg, Manitoba, Canada: IEEE.

Bao, F. (2013). *Dynamic Trust Management for Mobile Networks and Its Applications*. Virginia Polytechnic Institute and State University. Retrieved from http://vtechworks.lib.vt.edu/handle/10919/23157

Bao, F., & Chen, I.-R. (2012a). Dynamic trust management for internet of things applications.

In *2012 international workshop on Self-aware internet of things - Self-IoT '12* (pp. 1–6). San Jose, USA. https://doi.org/10.1145/2378023.2378025

Bao, F., & Chen, I.-R. (2012b). Trust management for the internet of things and its application to service composition. In *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2012 IEEE International Symposium on a* (pp. 1–6).

Barber, B. (1983). *The Logic and Limits of Trust*. New-Brunswick: Rutgers University Press.

Barron, E. N. (2013). *Game Theory*. https://doi.org/10.1002/9781118547168

Bassi, A., & Horn, G. (2008). *Internet of Things in 2020: A roadmap for the future. INFSO D.4 Networked Enterprise and RFID INFSO G.2 Micro & Nanosystems inco-operation with the RFID Working Group of the European Technology Platform on Smart Systems Integration (EPoSS)*. Brussels.

Bellifemine, F., Poggi, A., & Rimassa, G. (2001). Developing Multi-agent Systems with JADE. In C. Castelfranchi & Y. Lespérance (Eds.), *Intelligent Agents VII Agent Theories Architectures and Languages* (pp. 89–103). Berlin, Heidelberg: Springer Berlin Heidelberg.

Blaze, M., Feigenbaum, J., & Lacy, J. (1996). Decentralized trust management. In *IEEE Symposium on Security and Privacy* (Vol. 4, pp. 164–173). Oakland. https://doi.org/10.1109/SECPRI.1996.502679

Brey, P., & Søraker, J. (2009). Philosophy of Computing and Information Technology. *Philosophy of Technology and Engineering Sciences*, *14*, 1–77.

Brooks, F. P. (1996). The Computer Scientist as Toolsmith II. *Communications of the ACM*, *39*(3), 61–68. https://doi.org/10.1145/227234.227243

Cahill, V., Shand, B., Gray, E., Dimmock, N., Twigg, A., Bacon, J., … Nielsen, M. (2003). Using Trust for Secure Collaboration in Uncertain Environments. *IEEE P Ervasive Computing*, *2*(3), 52–61. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.111.3857&rep=rep1&type=pdf

Carroll, J. M., & Swatman, P. A. (2000). Structured-case: A methodological framework for building theory in information systems research. In *European Conference on Information Systems* (pp. 116–123). Vienna.

Cerf, V. G. (2012). Where is the Science in Computer Science? *Communications of the ACM*, *55*(10), 5. Retrieved from https://cacm.acm.org/magazines/2012/10/155530-where-is-the-science-in-computer-science/fulltext

Chang, E., Dillon, T. S., & Hussain, F. K. (2005). Trust and reputation relationships in service-oriented environments. *Proceedings - 3rd International Conference on Information Technology and Applications, ICITA 2005*, *I*(0), 4–14. https://doi.org/10.1109/ICITA.2005.168

Chen, D., Chang, G., Sun, D., Li, J., Jia, J., & Wang, X. (2011). TRM-IoT: A trust management model based on fuzzy reputation for internet of things. *Computer Science and Information Systems*, *8*(4), 1207–1228. https://doi.org/10.2298/CSIS110303056C

Chen, I.-R., Bao, F., & Guo, J. (2015). Trust-based service management for social internet of things systems. *Dependable Secure Computing*, *PP*(99).

Chen, I., Guo, J., & Bao, F. (2014). Trust Management for Service Composition in SOA- based IoT Systems, (4), 3486–3491. https://doi.org/10.1109/WCNC.2014.6953138

Cingolani, P., & Alcalá-Fdez, J. (2013). jFuzzyLogic: A Java Library to Design Fuzzy Logic Controllers According to the Standard for Fuzzy Control Programming. *International Journal of Computational Intelligence Systems*, *6*(SUPPL1), 61–75. https://doi.org/10.1080/18756891.2013.818190

Cofta, P. (2006). Distrust. In *The International Conference on Electronic Commerce (ICEC)*. Fredericton, Canada: ACM.

Colman, A. M. (2005). Game Theory. *Encyclopedia of Statistics in Behavioral Science*, *2*, 688–694. https://doi.org/10.3390/s111009327

Computer Science and Telecommunications Board National Research Council. (1994). *Academic Careers for Experimental Computer Scientist and Engineers*. Washington DC, USA: National Academy Press.

Dasgupta, P. (1988). Trust as a commodity. In D. Gambetta (Ed.), *Trust: Making and Breaking*

Cooperative Relations (pp. 49–72). New York: Basil Blackwell.

De Meo, P., Nocera, A., Quattrone, G., Rosaci, D., Ursino, D., Meo, P. De, … Calabria, R. (2009). Finding reliable users and social networks in a social internetworking system. In *Proceedings of the 2009 International Database Engineering & Applications Symposium* (pp. 173–181). https://doi.org/http://doi.acm.org/10.1145/1620432.1620450

De Oliveira Albuquerque, R., García-Villalba, L. ., & Kim, T. H. (2014). GTrust: Group extension for trust models in distributed systems. *International Journal of Distributed Sensor Networks*, *2014*. https://doi.org/10.1155/2014/872842

Denicolo, P., & Becker, L. (2012). *Developing research proposals*. Sage.

Denning, P. J. (1981). Performance Modeling: Experimental Computer Science at its Best. *Communications of the ACM*, *24*(November), 725–727.

Denning, P. J., Comer, D. E., Gries, D., Mulder, M. C., Tucker, A., Turner, J. A., & Young, P. R. (1989). Computing as a Discipline. *Communication of the ACM*, *32*(1), 9–23. Retrieved from http://delivery.acm.org/10.1145/70000/63239/p9-denning.pdf?ip=196.21.104.253&id=63239&acc=ACTIVE SERVICE&key=646D7B17E601A2A5.E6425C6EE37FBE0E.4D4702B0C3E38B35.4D4702B0C3E38B35&CFID=743777738&CFTOKEN=27284803&__acm__=1490601409_775eb0ca0d1bd7c303e

Dixit, A. K., & Nalebuff, B. J. (1993). *Game Theory*. Retrieved from http://www.econlib.org/library/Enc1/GameTheory.html

Dodig-crnkovic, G. (2002). Scientific Methods in Computer Science. *Computer (Long. Beach. Calif).*, 126–130. Retrieved from http://www.mrtc.mdh.se/~gdc/work/cs_method.pdf

Dodig-Crnkovic, G. (2002). Scientific Methods in Computer Science. In *Proceedings Conference for the Promotion of Research in IT at New Universities and at University Colleges*. Sweden.

Dong, P., Guan, J., Xue, X., & Wang, X. (2012). Attack-resistant trust management model based on beta function for distributed routing in internet of things. *China Communications*, *9*(4), 89–98.

Du, W., & Atallah, M. J. (2001). Secure Multi-Party Computation Problems and Their Applications : A Review and Open Problems. In *the 2001 workshop on New security paradigms (NSPW '01)* (pp. 13–23).

Duan, J., Gao, D., Foh, C. H., & Zhang, H. (2013). TC-BAC: A trust and centrality degree based access control model in wireless sensor networks. *Ad Hoc Networks*, *11*(8), 2675–2692. https://doi.org/10.1016/j.adhoc.2013.05.005

Eden, A. H. (2007). Three paradigms of computer science. *Minds and Machines*, *17*(2), 135–167. https://doi.org/10.1007/s11023-007-9060-8

Elkhodr, M., Shahrestani, S., & Cheung, H. (2016). The Internet of Things : New Interoperability, Management and Security Challenges. *International Journal of Network Security & Its Applications*, *8*(2), 85–102. https://doi.org/10.5121/ijnsa.2016.8206

Feitelson, D. G. (2006). Experimental computer science: The need for a cultural change.

Firdhous, M., Ghazali, O., & Hassan, S. (2011). Trust Management in Cloud Computing : A Critical Review. *International Journal on Advances in ICT for Emerging Regions*, *04*(02), 24–36. Retrieved from https://arxiv.org/ftp/arxiv/papers/1211/1211.3979.pdf

Fullam, K. K., Klos, T. B., Muller, G., Sabater, J., Schlosser, A., Topol, Z., … Voss, M. (2005). A specification of the Agent Reputation and Trust (ART) testbed: experimentation and competition for trust in agent societies. In *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems* (pp. 512–518). https://doi.org/10.1145/1082473.1082551

Fullér, R. (1995). *Neural Fuzzy Systems*. *Abo Akademi University*. https://doi.org/951-650-624-0

Gambetta, D. (2000). 'Can We Trust Trust? In D. Gambetta (Ed.), *Trust: Making and Breaking Cooperative Relations* (Electronic, pp. 213–237). Department of Sociology, University of Oxford. Retrieved from http://www.nuffield.ox.ac.uk/users/gambetta/Trust_making and breaking cooperative relations.pdf

Gangal, V., Narwekar, A., Ravindran, B., & Narayanam, R. (2016). Trust and Distrust Across Coalitions: Shapley Value Based Centrality Measures for Signed Networks (Student

Abstract Version). In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence* (pp. 4210–4211). AAAI Press. Retrieved from http://dl.acm.org/citation.cfm?id=3016387.3016521

Golbeck, J., & Hendler, J. (2006). Inferring Trust Relationships in Web-based Social Networks. *ACM Transactions on Internet Technology (TOIT)*, *6*(4), 497–529.

Grandison, T., & Sloman, M. (2003). *Trust Management for Internet Applications. iTrust*. Retrieved from http://www.doc.ic.ac.uk/~tgrand/iTrust.pdf

Griffiths, N., Chao, K. M., & Younas, M. (2006). Fuzzy trust for peer-to-peer systems. In *26th IEEE International Conference on Distributed Computing Systems Workshops (ICDCSW'06)* (pp. 73–79). https://doi.org/10.1109/ICDCSW.2006.57

Gu, L., Wang, J., & Sun, B. (2014). Trust management mechanism for Internet of Things. *China Communications*, *11*(2), 148–156. https://doi.org/10.1109/CC.2014.6821746

Guha, R., Kumar, R., Raghavan, P., & Tomkins, A. (2004). Propagation of Trust and Distrust. In *Proceedings of the 13th International Conference on World Wide Web* (pp. 403–412). New York, NY, USA: ACM. https://doi.org/10.1145/988672.988727

Guo, J., Chen, I., & Tsai, J. J. P. (2017). A survey of trust computation models for service management in internet of things systems. *Computer Communications*, *97*, 1–14. https://doi.org/10.1016/j.comcom.2016.10.012

Gutscher, A. (2007). A Trust Model for an Open , Decentralized Reputation System. In S. Etalle & S. Marsh (Eds.), *Joint iTrust and PST Conferences on Privacy Trust Management and Security, IFIPTM 2007* (pp. 285–300). Moncton, New Brunswick.

Gutscher, A., Heesen, J., & Siemoneit, O. (2008). Possibilities and limitations of modeling trust and reputation. In *CEUR Workshop Proceedings* (Vol. 332, pp. 50–61).

Hajek, P., Godo, L., & Esteva, F. (1995). Fuzzy Logic and Probability. In *The Eleventh Conference on Uncertainty in Artificial Intelligence* (pp. 237–244). Montréal, Qué, Canada: Morgan Kaufmann Publishers Inc. San Francisco, CA, USA.

Hassani, H. (2017). Research Methods in Computer Science: The Challenges and Issues. *CoRR*, *abs/1703.0*. Retrieved from http://arxiv.org/abs/1703.04080

Heckerman, D. (1995). *A tutorial on learning with bayesian networks*. Redmond. https://doi.org/Technical Report MSR-TR-95-06

Hoffman, L. J., Lawson-Jenkins, K., & Blum, J. (2006). Trust beyond security. *Communications of the ACM*, *49*(7), 94–101. https://doi.org/10.1145/1139922.1139924

IEC 61131. (2000). *International Electrotechnical Commission Technical Committee Industrial Process Measurement and Control. IEC 61131 - Programmable Controllers - Part 7: Fuzzy Control Programming.*

Jayasinghe, U., Lee, H.-W., & Lee, G. M. (2017). A computational model to evaluate honesty in social internet of things. In *Proceedings of the 32nd ACM SIGAPP Symposium on Applied Computing - SAC '17* (pp. 1830–1835). Marrakesh, Morocco. https://doi.org/10.1145/3019612.3019840

Jayasinghe, U., Otebolaku, A., Um, T.-W., & Lee, G. M. (2017). Data Centric Trust Evaluation and Prediction Framework for IoT. In *ITU Kaleidoscope Academic Conference* (pp. 147–153). Nanjing, China.

Johnson, N. L., Kemp, A. W., & Kotz, S. (1999). Univariate Discrete Distributions. In *Univariate Discrete Distributions* (2nd Editio, pp. 221–235). New Jersey: John Wiley & Sons.

Jøsang, A. (1996). The right type of trust for distributed systems. *Proceeding NSPW '96 Proceedings of the 1996 Workshop on New Security Paradigms*, 119–131. https://doi.org/10.1145/304851.304877

Kamvar, S. D., Schlosser, M. T., & Garcia-Molina, H. (2003). The Eigentrust algorithm for reputation management in P2P networks. In *12th International Conference on World Wide Web (WWW)* (pp. 640–651). https://doi.org/10.1145/775240.775242

Kim, I. Y., & de Weck, O. (2005). Adaptive Weighted Sum Method for Multiobjective Optimization. *Structural Multidisciplinary Optimization*, *29*, 149–158. Retrieved from http://web.mit.edu/deweck/www/PDF_archive/2 Refereed Journal/2_12_SMO_AWSMOO1_deWeck_Kim.pdf

Kothari, C. R. (2004). *Research Methodology Methods and Techniques* (2nd Editio). New Age International (P) Limited, Publishers.

Lewicki, R. J., Mcallister, D. J., & Bies, R. J. (1998). TRUST AND DISTRUST: NEW RELATIONSHIPS AND REALITIES. *Academy of Management Review*, *23*(3), 438–458.

Lewis, J. D., & Weigert, A. J. (1985). Trust as a Social Reality. *Social Forces*, *63*, 967–985.

Liang, Z., Shi, W., & Group, I. S. (2004). Enforce Cooperative Resource Sharing in Untrusted Peer-to-Peer Computing Environment 1 Introduction Overview. *Management*.

Liu, Y. B., Gong, X. H., & Feng, Y. F. (2014). Trust system based on node behavior detection in internet of things. *Journal of Communication*, *35*(5), 8–15.

Liu, Y., Chen, Z., Xia, F., Lv, X., & Bu, F. (2010). A Trust Model Based on Service Classification in Mobile Services. In *2010 IEEE/ACM International Conference on Green Computing and Communications, GreenCom 2010, 2010 IEEE/ACM International Conference on Cyber, Physical and Social Computing, CPSCom 2010* (pp. 572–576). Hangzhou, China. https://doi.org/10.1109/GreenCom-CPSCom.2010.19

Longman, W. (2003). ON THE INTERACTION BETWEEN THEORY , EXPERIMENTS , AND SIMULATION IN DEVELOPING PRACTICAL LEARNING CONTROL ALGORITHMS. *International Journal of Applied Mathematics and Computer Science*, *13*(1), 101–111.

Mahalle, P. N., Thakre, P. A., Prasad, N. R., & Prasad, R. (2013). A fuzzy approach to trust based access control in internet of things. *Wireless Communications, Vehicular Technology, Information Theory and Aerospace & Electronic Systems (VITAE)*, 2–6. https://doi.org/10.1109/VITAE.2013.6617083

Mäkinen, J. (2015). Data quality, sensitive data and joint controllership as examples of grey areas in the existing data protection framework for the Internet of Things. *Information and Communications Technology Law*, *24*(3), 262–277. https://doi.org/10.1080/13600834.2015.1091128

Marsh, S., Basu, A., & Dwyer, N. (2012). Rendering unto Cæsar the things that are Cæsar's: Complex trust models and human understanding. *IFIP Advances in Information and Communication Technology*, *374 AICT*, 191–200. https://doi.org/10.1007/978-3-642-29852-3_13

Marsh, S., & Dibben, M. R. (2005). Trust, Untrust, Distrust and Mistrust – An Exploration of the Dark(er) Side. In *Proc. 3rd International Conference, iTrust* (pp. 17–33). https://doi.org/10.1007/11429760_2

Marsh, S. P. (1994). *Formalising Trust as a Computational Concept*. University of Stirling. https://doi.org/10.2165/00128413-199409230-00010

Mc Kelvey, N., Kevin, C., & Subaginy, N. (2014). Internet of Things. In M. Khosrow-Pour (Ed.), *Encyclopedia of Information Science and Technology* (Third Edit, pp. 366–372). Hershey: IGI Global. https://doi.org/10.4018/978-1-60566-026-4

Mcknight, D. H., & Chervany, N. L. (2000). Trust and Distrust Definitions : One Bite at a Time. 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* (pp. 27–54). https://doi.org/10.1007/3-540-45547-7_3

Meghdadi, A. H., & Akbarzadeh-T, M.-R. (2001). Probabilistic Fuzzy Logic and Probabilistic Fuzzy System. In *IEEE International Fuzzy Systems Conference* (pp. 1127–1130).

Mora, M., Gelman, O., Steenkamp, A., & Raisinghani, M. (2012). Models for Interpretive Information Systems Research, Part 1: IS Research, Action Research, Grounded Theory – A Meta-Study and Examples. In *Research Methodologies, Innovations and Philosophies in Software Systems Engineering and Information Systems* (1st ed., pp. 222–237). Herrshey: IGI Global. https://doi.org/10.4018/978-1-4666-0179-6.ch011

Nafi, K. W., Kar, T. S., Hossain, A., & Hashem, M. M. A. (2012). An Advanced Certain Trust Model Using Fuzzy Logic and Probabilistic Logic theory. *(IJACSA) International Journal of Advanced Computer Science and Applications*, *3*(12), 164–173.

Nafi, K. W., Kar, T. S., Hossain, M. A., & Hashem, M. M. A. (2013). A fuzzy logic based certain trust model for E-commerce. *2013 International Conference on Informatics, Electronics and Vision, ICIEV 2013*. https://doi.org/10.1109/ICIEV.2013.6572693

Nauck, D. (1997). Neuro-Fuzzy Systems : Review and Prospects. In *Fifth European Congress on Intelligent Techniques and Soft Computing (EUFIT'97), Aachen* (pp. 1044–1053). Aachen, Germany: Aachen: ELITE-Foundation.

Negnevitsky, M. (2005). *Artificial Intelligence* (Second Edi). Harlow, England: Pearson Education Limited. Retrieved from www.pearsoned.co.uk

Nitti, M., Girau, R., & Atzori, L. (2014). Trustworthiness Management in the Social Internet of Things. *Knowledge and Data Engineering, IEEE Transactions On*. https://doi.org/10.1109/TKDE.2013.105

Nitti, M., Girau, R., Atzori, L., Iera, A., & Morabito, G. (2012). A subjective model for trustworthiness evaluation in the social Internet of Things. In *Personal Indoor and Mobile Radio Communications (PIMRC), 2012 IEEE 23rd International Symposium* (pp. 18–23). IEEE.

O'Leary, Z. (2004). *The Essential Guide To Doing Research*. London: SAGE Publications. Retrieved from http://books.google.com/books?hl=en&lr=&id=ItKeqNfgNW0C&oi=fnd&pg=PA1&dq=The+Essential+Guide+to+doing+research&ots=pJijdy6w9N&sig=sC4eomiaX2AQR_4ZcxvHBlfja80

Odejobi, T. (2012). *Research Methodology in Computer Science and Engineering*.

Oteafy, S. M. a, & Hassanein, H. S. (2012). Resource re-use in wireless sensor networks: Realizing a synergetic internet of things. *Journal of Communications*, *7*(7), 484–493. https://doi.org/10.4304/jcm.7.7.484-493

Pescatore, J. (2014). Securing the " Internet of Things " Survey. *Sans*, *44*(9), 22.

Ponce-Cruz, P., & Ramirez-Figueroa, F. D. (2010). Fuzzy Logic. In *Intelligent Control Systems with LabVIEW* (pp. 9–46). Springer-Verlag London Limited 2010. https://doi.org/10.1109/2.53

Rapaport, W. J. (2005). Philosophy of Computer Science: An Introductory Course Philosophy of Computer Science. *Teaching Philosophy*, *28*(4), 319–341. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.69.4787&amp;rep=rep1&amp;type=pdf

Ries, S., Kangasharju, J., & Mühlhäuser, M. (2006). A Classification of Trust Systems. In Z. Tari (Ed.), *On the Move to Meaningful Internet Systems 2006* (pp. 894–903). Montpellier, France: Springer-Verlag Berlin Heidelberg 2006. https://doi.org/10.1007/11915034_114

Roman, R., Najera, P., & Lopez, J. (2011). Securing the Internet of Things. *IEEE Computer*, *44*(January), 51–58.

Sabater, J., & Sierra, C. (2005). Review on Computational Trust and Reputation Models. *Artificial Intelligence Review*, *24*, 33–60. Retrieved from http://www.emse.fr/~vercouter/tutorial/EASSS09Trust.pdf

Saied, Y. Ben, Olivereau, A., Zeghlache, D., & Laurent, M. (2013). Trust management system design for the Internet of Things: A context-aware and multi-service approach. *Computers and Security*, *39*, 351–365. https://doi.org/10.1016/j.cose.2013.09.001

Saunders, M., Lewis, P., & Thornhill, A. (2008). *Research Methods for Business Students*. *Research methods for business students* (5th Editio). https://doi.org/10.1007/s13398-014-0173-7.2

Saunders, M., Lewis, P., & Thornhill, A. (2009). *Research Methods for Business Students* (Fifth edit). Pearson Education.

Schmidt, S., Steele, R., Dillon, T. S., & Chang, E. (2007). Fuzzy trust evaluation and credibility development in multi-agent systems. *Applied Soft Computing Journal*, *7*(2), 492–505. https://doi.org/10.1016/j.asoc.2006.11.002

Selcuk, A. ., Uzun, E., & Pariente, M. . (2004). A Reputation-Based Trust Management for P2P Networks. *International Journal of Network Security*, *6*(3), 235–245.

Sicari, S., Rizzardi, a., Grieco, L. a., & Coen-Porisini, a. (2015). Security, privacy and trust in Internet of Things: The road ahead. *Computer Networks*, *76*, 146–164. https://doi.org/10.1016/j.comnet.2014.11.008

Smith, P. N. (1994). Applications of Fuzzy Sets in the Environmental Evaluation of Projects. *Journal of Environmental Management*, *42*, 365–388.

Suo, H., Wan, J., Zou, C., & Liu, J. (2012). Security in the internet of things: a review. In *International Conference on Computer Science and Electronics Engineering (ICCSEE)* (pp. 648–651).

Suparta, W., & Alhasa, K. M. (2016). Adaptive Neuro-Fuzzy Interference System. In *Modeling* *of Tropospheric Delays Using ANFIS* (pp. 5–19). SpringerBriefs in Meteorology. https://doi.org/10.1007/978-3-319-28437-8

Sushmita, M., & Sankar, P. K. (1996). Neuro-Fuzzy Expert Systems: Relevence, Features and Methodologies. *Journal of the IETE*, *42*(4 & 5), 335–347.

Swinth, R. L. (1967). Journal of Conflict Resoltion. In *Journal of Conflict Resolution* (Vol. 11, pp. 334–344).

Tang, J., Hu, X., & Liu, H. (2014). Is distrust the negation of trust? In *Proceedings of the 25th ACM conference on Hypertext and social media - HT '14* (pp. 148–157). ACM. https://doi.org/10.1145/2631775.2631793

Tedre, M. (2007). Know your discipline: teaching the philosophy of Computer Science. *Journal of Information Technology Education*, *6*(1), 105–122.

The Mathwork Inc. (2017). *Fuzzy Logic Toolbox TM User ' s Guide*.

Thomborson, C. (2010). Axiomatic and Behavioural Trust. In *International Conference on Trust and Trustworthy Computing* (pp. 352–366). TRUST'10 Proceedings of the 3rd international conference on Trust and trustworthy computing. https://doi.org/10.1007/978-3-642-13869-0

Tong, X., Zhang, W., Long, Y., & Huang, H. (2013). Subjectivity and objectivity of trust. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *7607 LNAI*, 105–114. https://doi.org/10.1007/978-3-642-36288-0_10

Truong, N. B., Lee, H., Askwith, B., & Lee, G. M. (2017). Toward a trust evaluation mechanism in the social internet of things. *Sensors (Switzerland)*, *17*(6), 1–24. https://doi.org/10.3390/s17061346

Turner, R., & Angius, N. (2017). The Philosophy of Computer Science. In *The Stanford Encyclopedia of Philosophy* (Spring 201). Retrieved from https://plato.stanford.edu/archives/spr2017/entries/computer-science/

Umarani, V., & Sundaram, K. S. (2013). Survey of Various Trust Models and Their Behavior in Wireless Sensor Networks. *International Journal of Emerging Technology and Advanced*

Engineering, *3*(10), 180–188.

Vermesan, O., Friess, P., Guillemin, P., Gusmeroli, S., Sundmaeker, H., Bassi, A., … Doody, P. (2011). *Internet of Things Strategic Research Roadmap*.

Viharos, Z. J., & Kis, K. B. (2014). Survey on Neuro-Fuzzy systems and their applications in technical diagnostics and measurement. In *13th IMEKO TC10 Workshop on Technical Diagnostics Advanced measurement tools in technical diagnostics for systems' reliability and safety*. Warsaw, Poland. https://doi.org/10.1016/j.measurement.2015.02.001

Wang, J., Bin, S., Yu, Y., & Nui, X. (2013). Distributed Trust Management Mechanism for the Internet of Things. *Applied Mechanics and Materials*, *347–350*(4), 2197–2200. https://doi.org/10.4028/www.scientific.net/AMM.347-350.2463

Wang, Y. (2005). Trust 2 : Developing Trust in Peer-to-Peer Environments. In *2005 IEEE International Conference on Services Computing (SCC'05)*. Orlando, FL, USA: IEEE.

Wang, Y., Cahill, V., Gray, E., Harris, C., & Liao, L. (2006). Bayesian network based trust management. *Autonomic and Trusted …*, (2006), 1–13. Retrieved from http://www.springerlink.com/index/g10g777pt4311253.pdf

Wang, Y., & Vassileva, J. (2003a). Bayesian Network-Based Trust Model in Peer-to-Peer Networks. In *The Sixth International Workshop on Trust, Privacy, Deception and Fraud in Agent Systems, 200* (pp. 23–34).

Wang, Y., & Vassileva, J. (2003b). Trust and Reputation Model in Peer-to-Peer Networks. In *Third In'l Conf. on Peer-to-Peer Computing* (pp. 150–157). Zurich.

Wegner, P. (1976). Research paradigms in computer science. In *2nd International Conference of Software Engineering* (pp. 322–330). San Francisco, CA.

Wen-Mao, L., Li-Hua, Y., Bin-Xing, F., & Hong-Li, Z. (2012). A hierarchical trust model for the internet of things. *Chinese Journal of Computers*, *5*, 846–855.

Woody, C. (1927). The Values of Educational Research to the Classroom Teacher. *The Journal of Educational Research*, *16*(3), 172–178. https://doi.org/10.1080/00220671.1927.10879779

Xia, H., Jia, Z., Ju, L., Li, X., & Zhu, Y. (2011). A subjective trust management model with multiple decision factors for MANET based on AHP and fuzzy logic rules. In *2011 IEEE/ACM International Conference on Green Computing and Communications, GreenCom 2011* (pp. 124–130). https://doi.org/10.1109/GreenCom.2011.30

Xiong, L., & Liu, L. (2003). Peer Trust: Supporting Reputation-Based Trust for Peer-to-Peer Electronic Communities. *Conference on E-Commerce*, (0).

Yan, Z., & Holtmanns, S. (2007). Trust modeling and management: from social trust to digital trust. In R. Subramanian (Ed.), *Computer Security, Privacy and Politics: Current Issues, Challenges and Solutions* (pp. 290–323). IGI Global. https://doi.org/10.4018/978-1-59904-804-8.ch013

Yan, Z., Zhang, P., & Vasilakos, a. V. (2014). A survey on trust management for Internet of Things. *Journal of Network and Computer Applications*, *42*(0), 120–134. https://doi.org/doi:10.1016/j.jnca.2014.01.014

Yu, B. Y. Bin, Singh, M. P., & Sycara, K. (2004). Developing trust in large-scale peer-to-peer systems. *IEEE First Symposium OnMulti-Agent Security and Survivability, 2004*. https://doi.org/10.1109/MASSUR.2004.1368412

Yu, Y., Li, K., Zhou, W., & Li, P. (2012). Trust mechanisms in wireless sensor networks: Attack analysis and countermeasures. *Journal of Network and Computer Applications*, *35*(3), 867–880. https://doi.org/10.1016/j.jnca.2011.03.005

Zadeh, L. A. (1965). Fuzzy Sets. *Information and Control*, *8*, 338–353. https://doi.org/10.1109/2.53

Zadeh, L. A. (2003). *Probability theory & fuzzy logic*. Retrieved from http://kmh-lanl.hansonhub.com/uncertainty/meetings/zadeh03vgr.pdf

Zhang, Q., Yu, T., & Irwin, K. (2004). A Classification Scheme for Trust Functions in Reputation-Based Trust Management. In *International Workshop on Trust, Security, and Re putation on the Semantic Web*. Hiroshima, Japan.