Trust Evaluation in the IoT Environment

By

Upul Jayasinghe

A thesis submitted in partial fulfilment of the requirements of Liverpool John Moores University for the degree of Doctor of Philosophy

July 2018

DECLARATION

The work presented in this thesis was carried out at the Department of Computer Science, Liverpool John Moores University. Unless otherwise stated, it is the original work of the author.

While registered as a candidate for the degree of Doctor of Philosophy, for which submission is now made, the author has not been registered as a candidate for any other award. This thesis has not been submitted in whole, or in part, for any other degree.

Upul Jayasinghe,

Department of Computer Science,
James Parsons Building,
Liverpool John Moores University,
3 Byrom Street, L3 3AF,
Liverpool, UK.

SUPERVISORS CERTIFICATION

We certify that the thesis entitled "**Trust Evaluation in the IoT Environment**" was prepared under our supervision at the Department of Computer Science, Faculty of Engineering and Technology, Liverpool John Moores University as a partial of fulfilment of the requirements of Liverpool John Moores University for the degree of Doctor of Philosophy.

Signature :

Name : Dr. Gyu Myoung Lee

Title : Director of study

Address : 6.25 James Parsons Building, 3 Byrom St, Liverpool L3 3AF, UK.

Date :

Signature :

Name : Prof. Qi Shi

Title : Second Supervisor

Address : 6.33 James Parsons Building, 3 Byrom St, Liverpool L3 3AF, UK

Date:

Signature :

Name : Dr. Nathan Shone

Title : Third Supervisor

Address : 6.47 James Parsons Building, 3 Byrom St, Liverpool L3 3AF, UK

Date :

EXAMINING COMMITTEE CERTIFICATION

We certify that we have read the thesis entitled "**Trust Evaluation in the IoT Environment"** and as Examining Committee has examined **Mr. Upul Jayasinghe** in its content and that in our opinion, it is adequate for the award of the degree of Doctor of Philosophy.

Signature :

Name :

Title :

Address :

Date :

Signature :

Name :

Title :

Address :

Date :

Approved by the Department of Computer Science

Signature :

Name :

Title :

Address :

Date :

ACKNOWLEDGMENT

First, I would like to thank the Department of Computer Science, LJMU for giving me the opportunity to undertake this research degree. Not only has this opportunity allowed me to develop many new skills but also it has given an invaluable chance to extend my career ambitions with international exposure.

The support and guidance I have received have been essential to the completion of my research, and for this, I would like to thank my supervisory team. My sincerest thanks go to my director of studies, Dr. Gyu Myoung Lee, for his guidance, advice, critique, motivation and above all, his invaluable expertise during the course of this research. He helped me to continually progress and build upon my ideas, ultimately resulting in the completion of this thesis with a number of publications. I would also like to thank my co-supervisors Professor Qi Shi and Dr. Nathan Shone for their continued advice, support, and effort taken to improve the quality of work.

I would also like to show my appreciation for the tremendous support given by the research administrators especially Tricia Waterson, Ian Fitzpatrick, and Carol Oliver. I would like to thank all my friends and colleagues at LJMU for their invaluable advice, help, and support with special thanks going to Nguyen Binh Truong, Ali Alfoudi, Mohammed Daghriri, and Abayomi Otebolaku.

I thank my parents and my brother, who have supported me not just throughout my research but also throughout my life. I would also love to thank my wife. I do not believe I can finish this dissertation without her generous support. Without them, I would never have reached this far. Finally, I take this opportunity to express my gratitude to everyone who supported me throughout this journey.

ABSTRACT

Along with the many benefits of IoT, its heterogeneity brings a new challenge to establish a trustworthy environment among the objects due to the absence of proper enforcement mechanisms. Further, it can be observed that often these encounters are addressed only concerning the security and privacy matters involved. However, such common network security measures are not adequate to preserve the integrity of information and services exchanged over the internet. Hence, they remain vulnerable to threats ranging from the risks of data management at the cyber-physical layers, to the potential discrimination at the social layer. Therefore, trust in IoT can be considered as a key property to enforce trust among objects to guarantee trustworthy services.

Typically, trust revolves around assurance and confidence that people, data, entities, information, or processes will function or behave in expected ways. However, trust enforcement in an artificial society like IoT is far more difficult, as the things do not have an inherited judgmental ability to assess risks and other influencing factors to evaluate trust as humans do. Hence, it is important to quantify the perception of trust such that it can be understood by the artificial agents. In computer science, trust is considered as a computational value depicted by a relationship between trustor and trustee, described in a specific context, measured by trust metrics, and evaluated by a mechanism.

Several mechanisms about trust evaluation can be found in the literature. Among them, most of the work has deviated towards security and privacy issues instead of considering the universal meaning of trust and its dynamic nature. Furthermore, they lack a proper trust evaluation model and management platform that addresses all aspects of trust establishment. Hence, it is almost impossible to bring all these solutions to one place and develop a common platform that resolves end-to-end trust issues in a digital environment.

Therefore, this thesis takes an attempt to fill these spaces through the following research work. First, this work proposes concrete definitions to formally identify trust as a computational concept and its characteristics. Next, a well-defined trust evaluation model is proposed to identify, evaluate and create trust relationships among objects for calculating trust. Then a trust management platform is presented identifying the major

tasks of trust enforcement process including trust data collection, trust data management, trust information analysis, dissemination of trust information and trust information lifecycle management. Next, the thesis proposes several approaches to assess trust attributes and thereby the trust metrics of the above model for trust evaluation. Further, to minimize dependencies with human interactions in evaluating trust, an adaptive trust evaluation model is presented based on the machine learning techniques.

From a standardization point of view, the scope of the current standards on network security and cybersecurity needs to be expanded to take trust issues into consideration. Hence, this thesis has provided several inputs towards standardization on trust, including a computational definition of trust, a trust evaluation model targeting both object and data trust, and platform to manage the trust evaluation process.

PUBLICATIONS AND CONTRIBUTIONS

JOURNAL PAPERS

1. **U. Jayasinghe**, G. M. Lee, T.-W. Um, and Q. Shi, "Machine-Learning-based Trust Computational Model for IoT Services," *IEEE Transactions on Sustainable Computing*, Accepted for Publication, 2018.
2. N. B. Truong, **U. Jayasinghe**, T.-W. Um, and G. M. Lee, "A Survey on Trust Computation in the Internet of Things," *The Journal of Korean Institute of Communications and Information Sciences (J-KICS)*, vol. 33, no. 2, pp. 10-27, 2016.

CONFERENCE PAPERS

1. **U. Jayasinghe**, A. Otebolaku, T.-W. Um, and G. M. Lee, "Data centric trust evaluation and prediction framework for IOT," *in ITU Kaleidoscope: Challenges for a Data-Driven Society*, Nanjing, China, 2017, pp. 1-7.
2. **U. Jayasinghe**, H. Lee, and G.M. Lee. "A computational model to evaluate honesty in social internet of things." *in Proceedings of the Symposium on Applied Computing (ACM)*, Marrakesh, Morocco, 2017, pp. 1830-1835.
3. **U. Jayasinghe**, N. B. Truong, G. M. Lee, and T.-W. Um, "RpR: A Trust Computation Model for Social Internet of Things," *in Intl. IEEE Conferences on Ubiquitous Intelligence & Computing*, 2016, Toulouse, France, pp. 930-937.
4. G. M. Lee, **U. Jayasinghe**, N. B. Truong, and C.-h. Cho, "Features, Challenges and Technical Issues," *in Second Bright ICT Annual Workshop on Bright ICT 2016*, Dublin, Ireland, 2016.
5. **U. Jayasinghe**, and G. M. Lee, "Adaptive Trust Computation Algorithm for Internet of Things", *Faculty Research* Conference*: Liverpool John Moores University*, 2018, Liverpool, UK.

TECHNICAL MEETINGS

1. Verified trustworthy software systems, Royal Society, London, and Imperial College, London (A Physical Meeting), 2016.
2. Worldwide interoperability for semantics IoT (WISE-IoT) Project (Weekly Online Meetings), 2016-2018.

STANDARDIZATION

1. "Overview of trust provisioning for ICT infrastructures and services," in Recommendation Y.trust-provision vol. Q16/13, ed: ITU-T, 2016.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

CFD	Cooperativeness, Frequency and Duration
CLR	Co-Location Relationship
CoI	Community of Interest
CPS	Cyber-Physical System
CPSS	Cyber-Physical-Social System
CRAWDAD	Community Resource for Archiving Wireless Data At Dartmouth
CWR	Co-work Relationship
DTA	Data Trust Attribute
DTM	Data Trust Metric
FSM	Finite State Machines
ICT	Information and Communication Technology
IoT	Internet of Things
ITU	International Telecommunications Union
ITU-T	ITU Telecommunication Standardization Sector
KNN	K-Nearest Neighbors
MANET	Mobile Ad-hoc Networks
MSE	Mean Square Error
OOR	Ownership Object Relationship
OSN	Online Social Networks
P2P	Peer-to-Peer
PCA	Principal Component Analysis
PKI	Public Key Infrastructure
POR	Parental Object Relationship
QoS	Quality of Service
RBFK	Radial Basis Function Kernel
RpR	Recommendations plus Reputations
SIoT	Social IoT
SPOF	Single Point of Failure
SVM	Support Vector Machine
TAG	Trust Agent
TCPD	Trust Computation, Prediction and Decision Making
TII	Trusted Information Infrastructure
WSN	Wireless Sensor Networks

CHAPTER 1: INTRODUCTION

1.1. Background

In the past decade, the concept of the IoT has attracted plenty of research and produced many smart services, where large numbers of heterogeneous objects collaborate to solve complex problems. Nevertheless, the scope of the current IoT ecosystem is being further expanded due to the novel integration of social network paradigms within the conventional IoT model, breeding more innovative business models for a futuristic society [1], [2].

However, along with the many benefits of these findings, the diversity and sensitivity of the information, which is being shared, brings a completely new challenge to establish a trustworthy environment among objects connected. But, it can be observed that often these challenges are addressed only by considering security matters involved and do not evaluate the social and subjective risks among IoT objects and services [3]. If knowledge is exploited for malicious intentions, it could result in irreparable damage and uncertain dangers. Therefore, the concept of trust in SIoT can be considered as a key property to establishing trustworthy and reliable service provisioning among various objects.

The trust concept is an abstract notion, with different meanings depending on both participants and scenarios; and influenced by both measurable and non-measurable factors. Moreover, inconsistency in trust definitions is leading to difficulty in establishing a common, general notation that holds regardless of personal dispositions or differing situations. Generally, trust can be defined as a qualitative or quantitative property of a trustee measured by a trustor for a given task in a specific context and in a specific time period. Trust is a fundamental fact that affects the appetite of an object to consume a particular service or product offered by another. This can be observed in everyday life, where trust decisions are made. When purchasing a specific product, we may favor certain brands due to our trust that these brands will provide excellent quality compared to the unknown brands. Trust in these brands may come from our past experience of using their products (termed “belief”) or from their reputations, which are perceived from other people who bought items and left their opinions about

those products (termed “reputation”), or recommendations from close peers such as families and friends (termed “recommendation”).

Although the significance of trust in our physical world is as important as it is in the digital environment, building trust and confidence in the latter is much more difficult. This is due to our inability to have a physical view of an object, unlike in our physical world, where we can view the building of the bank, observe its safe deposits, meet the bank personnel, etc. Another issue with trust is that it is difficult to quantify the exact trustworthiness value of an object. This is even harder when each object has different interpretations and perceptions of the term “trustworthy”. Therefore, they may assign different trustworthiness values to the same provider or the service. For example, a service consumer assigns “very trustworthy” to the provider for a transaction that he has performed. However, another consumer assigns “untrustworthy” for a similar transaction from the same provider. These differences further increase the difficulty in determining the exact trustworthiness of an object.

Currently, no distinguishable solution can provide a generic trust modeling mechanism, trust management platform, and quantifiable solutions for assessing, and aggregating trust metrics (TM) based on numerical and Artificial Intelligence (AI) methodologies for the IoT. As such, the motivation of this research is to address these substantial gaps in the domain of IoT and present a platform that can support establishing trustworthy services and applications. Hence, this chapter will introduce the research within this thesis, along with the research problem, research issues, gap of knowledge, research motivation, research aims and objectives, research novelty, and lastly the overall structure of this thesis is outlined.

1.2. Research Problem

As humans, we often make decisions based on trust that we have on other party and for that inherited judgmental knowledge helps us to analyze risks and related factors to build such perception of trust. Similarly, in an artificial society like IoT, things must take necessary decisions at right time to achieve the goals set by humans and avoid any threats due to malicious counterparts. However, the inability of things to build up a judgmental knowledge like humans, make them vulnerable to risks associated with the environment. Therefore, it is important to formalize trust as a quantifiable concept

to be used in an artificial society to mitigate such risks and realize perception of trust for autonomous decision making, which leads to the research question of this work;

Is it possible to quantify trust as a computational property to be used by objects in a digitized society to combat the misbehaviors' and establish a trustworthy environment for its users?

1.3. Research Issues and the Gap of Knowledge

Trust is a broad concept with application across many disciplines and subject areas but with no commonly agreed definition. A review of the economic literature on trust found that the existence of uncertainty was one factor present in most definitions of trust. It is a critical factor that highly influences the likelihood of entities to interact and transact in digital environments. However, when it comes to quantitative modeling of trust in IoT, there is no commonly agreed definition for trust. Due to this issue, most of the trust based solutions presented in the literature address quite isolated problems and have become a major drawback in the field to come up with more generic solutions to mitigate trust related issues in IoT.

Moreover, the existing mechanisms for trust management are quite limited to Cyber-Physical Systems (CPS) like Peer-to-Peer (P2P) and Mobile Ad-hoc Networks (MANET). There is no comprehensive management platform or evaluation model available to extract TMs from CPSS [3]. One of the most important gaps that this thesis intends to discuss here is the lack of using environment information for trust evaluation. In an IoT environment, physical devices are owned by human-related factors and inherently socially connected by a physical-cyber-social system. Moreover, current trust evaluation methods also lack concerns about trustors' subjective properties, in other words, the trust results in a personalized expectation. Furthermore, a most common method of assessing trust in IoT applications is to estimate trust level of the objects and trust level of the data is assumed to be the same as the trust level of the data source. However, there are situations where it is impossible to evaluate trust of the objects due to lack of information. Besides, most of the services in IoT (e.g. clouded based services) often worry about the trustworthiness of data instead of the data source who is generating them. Therefore, it is also important to address the challenge of evaluating trust of data while preserving the traditional form of trust evaluation which is for objects.

On the other hand, trust aggregation methods play an important role in trust evaluation. This is vital in a situation where there are several distinct trust attributes (TA) needed to combine into one final TM; and where there are, several TMs need to be combined into the overall trust value for evaluation. However, only limited work is available in this area and the simplest method to deal with trust aggregation currently is to use a static weighted sum method. However, this solution is not smart enough considering the complexity of an IoT environment and the varied influence of each factor toward trust evaluation. Hence, schemes that are more intelligent are required to find these weighting factors and thresholds that define a trustworthy boundary, for example, using machine learning and data analytics techniques.

1.4. Research Motivation

Trust evaluation is currently an emerging concept especially in the digitized environment as it shows quite a significant potential towards eliminating risks related to privacy and preserving the integrity of the interactions. Despite the many positives that the concept of trust could bring to establish a trustworthy environment for its users, it still lacks several important facts as discussed in Section 1.3 which need to be solved immediately before deploying it in a real-world environment to avoid unnecessary threats which can be initiated within the system and outside.

Therefore, this work is motivated by three main research challenges, which are:

1. Formalizing concept of computational trust in the IoT environment:

   The concept of computational trust in different communities varies in how it is represented, computed, and used. Due to these inherited differences, trust is hard to formalize in a general setting, and up to now, no commonly accepted definition appears. However, it is important to quantify the level of trust in IoT for the reasons discussed in Section 1.3. Hence, motivations to overcome these issues are the formulation of a generic definition for trust irrespective of the environment.

2. Design and Develop trust evaluation mechanism targeting IoT environment:

   To evaluate trust, there are many properties that need to be considered. These properties could be trust-related attributes as well as environment-related

attributes. Moreover, depending on services and applications, the required attributes of trust may vary. For example, for a particular application, technical attributes may consist of security, reliability, and availability. Whereas, for other applications, security and reliability might be enough.

Therefore, motivation to solve this gap can be resolved in two steps: The first one is to develop a trust evaluation model, that basically identify TAs which influence the particular service, evaluate the identified TAs and the creation of trust relationships based on the trust values. The second one is to explore a trust management platform, that helps on managing the trust evaluation process from data collection to trust information lifecycle management.

3. Development of intelligent trust aggregation technique:

   Once all the TAs are computed, it is important to combine them accordingly to observe the influence of a particular TA on final trust. Current methods mostly ignore this fact and simply use linear equations with random weighting factors to aggregate all the TAs and TMs. But the assessment of a proper weight is a complex task since trust is a varying quantity which depends on many factors, e.g. expectations of a trustor, time, context, etc. Thus, schemes that are more intelligent are required to find these weighting factors and a threshold that defines a trustworthy boundary. This gap motivates to investigate more effective trust accumulation methods such as the application of ML techniques.

1.5. Research Aims and Objectives

The aims of this research are to identify the benefit of establishing trust in an IoT ecosystem in addition to privacy and security, recognize limitations of existing solutions, and then to design a trust management platform that negotiates with applications and services to autonomously establish trust among them. Furthermore, the proposed solution must be able to fill the gap created by existing methods and correctly identify and mitigate misconducts through the application of intelligent algorithms.

The thesis work towards these aims by addressing the following objectives:

1. Conduct a thorough survey and investigation on; the necessity of trust as a computational concept and recognizing positives and negatives of existing systems (Chapter 2).

2. Design a generalized trust definition, trust evaluation model, and a trust management platform in which trust can be formalized and produced for any service objects or data objects in IoT (Chapter 3).

3. Intelligent algorithms and assessment approaches to identify the trustworthy objects in a given service scenario and predict the trustworthiness of each object in future transactions (Chapter 4, 5, 6, and 7).

4. Demonstrate the effectiveness of devised solutions and techniques based on real-world use cases (Chapter 4, 5, 6, and 7).

5. Collaborate with standardization bodies to stimulate standardization activities on trust based on the novel findings (Chapter 8).

1.6. Research Contributions

This thesis presents several novel contributions based on the aims and objectives of our research. The major contributions of this thesis are to formulate the concept of computational trust, trust evaluation model based on knowledge, experience, and reputation, trust management platform and their novelties are discussed at the beginning of each chapter where relevant. The most important contributions are:

1. Development of a formalized definition for computational trust, trust evaluation model and a trust management platform, which defines a complete trust provisioning lifecycle (Section 3.2, 3.3 and 3.4) and its extended version towards to evaluate trust of data in Chapter 7 (Section 7.2) along with a possible implementation scenario based on crowd sensing use case in a distributed environment (Section 7.3),

   Until now, most of the trust profiling models and platforms have been developed to target specific application areas such as MANETs, P2P, etc. or security and privacy aspects of them. None of them addresses a generic trust model that can represent all aspects of trust evaluation based on direct interactions, experience, and global opinions. In contrast, the thesis develops a

prototype that can evaluate trust in any given scenario based on knowledge, experience, and reputation. Then, based on the application or service area, the TAs, which define the major TMs, can be extracted accordingly. Furthermore, we elaborate on the components that any trust management platform must have to evaluate trust according to the proposed model.

2. Reputation based trust evaluation (Chapter 4).

   According to the proposed trust evaluation model, reputation plays a vital role in trust evaluation, as it converges millions of opinions to one single quantity that can be used as a metric to make a judgment before the prospective actions. Hence, this work proposes a novel reputation and experience based trust assessment algorithm for IoT that can calculate relationships among objects accordingly before making any associations with them. The basic underlying principle in this algorithm is to collect feedback from neighbor objects as experiences or reputations based on the previous interactions with them. The numerical model adapts the concepts from the PR algorithm and improves it in such a way that it generates a trust network based on recommendations and reputations instead a number of incoming and outgoing links from an object. Further, it shows a significant improvement in terms of detecting most trustworthy objects, compared to PR or ID algorithms.

3. Knowledge based trust evalaution (Chapter 5).

   Even though IoT environment produces large amounts of data, it is questionable how much of them can be directly useful for a trustworthy evaluation process. Therefore, it is vital to extract trust features by scanning social and system level interaction logs and storing them in a Data Repository (DR) for further analysis. Hence, a novel numerical model that can extract such features is proposed here. First, critical TAs, which represent features of direct interactions between trustor and trustee, are identified. Then, a quantifiable model is formulated for each of these properties and trustworthiness is evaluated based on these equations. In contrast to current methods in which they consider only dependable features like delay, bandwidth, QoS, etc., the proposed model is more concentrated on realizing the true meaning of social trust and hence each of these models is defined and estimated based on social

properties like relationships, credibility, social and temporal proximity. Further, to improve the practical significance of each model and its meaningfulness, the numerical model is developed based on an actual data set that in fact represents a real-world use case.

4. Machine learning based trust evaluation (Chapter 6); and an extended version towards data trust prediction based on CF (Chapter 7),

   One of the main motivations to carry out this research is the lack of an effective mechanism to determine weighting factors to combine several TAs together to estimate final trust value. To date, this has been done using trial and error methods or assigning predetermined thresholds depending on the application area in the context of trust evaluation. Furthermore, unavailability of a mechanism that can correctly label an interaction depending on its trustworthiness significantly affects the progression of trust related research work and its developments. To solve these two issues, first, an unsupervised learning algorithm based on the k-means algorithm that in fact have the intelligence to differentiate the trustworthiness of each interaction is developed. Based on the results of this algorithm, two more algorithms are designed to predict future interactions. The first algorithm is based on Support Vector Machines (SVM) and the second one is based on the Collaborative Filtering (CF) concepts.

   Moreover, the proposed clustering model in combination with a classification algorithm shows 2% improvement in contrast to previous algorithms in terms of trust aggregation. In addition, this algorithm can adapt to the changes of the interactions over time and gaining a more powerful insight compared to the traditional methods like linear aggregation in which behaviors of the objects are believed in such a way that they would act in the same manner in the future, as before. This shows a prominent feature of our algorithm towards designing an autonomous system that can assess trust dynamically without external interventions and predicting future misbehaviors intelligently. Furthermore, the proposed data trust prediction algorithm based on CF suggests a solution to estimate trustworthiness of objects who do not have prior encounters. This has never been addressed before in the literature and it further enhances the

accuracy of the proposed model in Section 3.3 when there is only limited information available to calculate trust.

5. Standardization work on trust provisioning (Chapter 8)

   As existing standardization activities on trust are still limited, the thesis has extended the results explained in Section 3.2, Section 3.3, and Section 3.4 towards creating a standard for trust evaluation and management with the collaboration with ITU-T.

1.7. Thesis Organization

The remainder of the thesis is organized into the following seven chapters and the structure of this thesis is shown in Figure 1-1.

Chapter 2: Literature Review

This chapter outlines a literature review on existing research on trust to investigate its positives and negatives. This, in fact, provides the motivation to carry out following research by realizing some of the gaps including unavailability of a generic trust evaluation model and a trust management platform for IoT; estimating only dependable trust in contrast to social trust; absence of autonomous trust aggregation and prediction methods; and limited work on data trust estimation and prediction. For that, previous work related to trust based on their trust evaluation model, trust management, and their novelty are thoroughly investigated.

Chapter 3: Trust Evaluation and Trust Management

This chapter first describes the trust concept and key characteristics of trust. Then it proposes a possible evaluation model for trust and a trust management platform that explains all the steps in the trust evaluation process in an IoT environment starting from data collection to decision making.

Chapter 4: Reputation based Trust Evaluation

This chapter proposes a algorithm called RpR based on recommendations and reputations provided by the objects as an aggregated result of feedback in a distributed IoT environment. First, it explains a novel numerical model to aggregate all the experiences and opinions from the network based on graph theory concepts. Then a

novel algorithm is proposed to evaluate the trustworthiness of each object autonomously in a distributed environment. Finally, the effectiveness and performance of the algorithm over other similar systems is demonstrated in a simulation environment.

Chapter 5: Knowledge based Trust Evaluation

This chapter proposes a direct trust evaluation model, which is a vital TM in the process of trust provisioning, especially when there are no previous encounters. First, important TAs that can define the knowledge in a particular use case, are identified. Then based on their definitions, a quantifiable model for each of these properties is formulated to evaluate the trustworthiness. Finally, the effectiveness of this model is tested based on a real-world data set obtained from a real-world use case.

Chapter 6: Machine Learning based Trust Evaluation

This chapter proposes an innovative approach to trust labeling, evaluation, and prediction based on ML concepts. One of the issues on applying intelligent learning methods in trust evaluation is that the unavailability of labeled data sets based on their trustworthiness. Hence, an algorithm based on unsupervised learning is discussed first to label any data set with respect to trustworthiness. Then, based on this information, a trust prediction algorithm is presented. Finally, the effectiveness of the proposed algorithm is verified through a simulation but using a real-world data set.

Chapter 7: Data Trust Evaluation

This chapter proposes a hybrid trust management platform, which is capable of evaluating both data trust as well as traditional object based trust. As data trust is quite different from the object based trust, a unique trust evaluation model that represents unique features of data is presented first. Afterward, a model to compute individual DT attributes and the Data TMs (DTM) is discussed. Additionally, a trust prediction algorithm based on CF is proposed to find the DT between trustors and data sources who have not had prior encounters. Finally, a possible implementation scenario is discussed based on a crowd sensing use case.

Chapter 8: Conclusion and Future Work

This chapter summarizes the findings of this thesis and describes the extent of the limitations overcome by this research. Furthermore, it reveals future directions of this work; how the research will be continued.

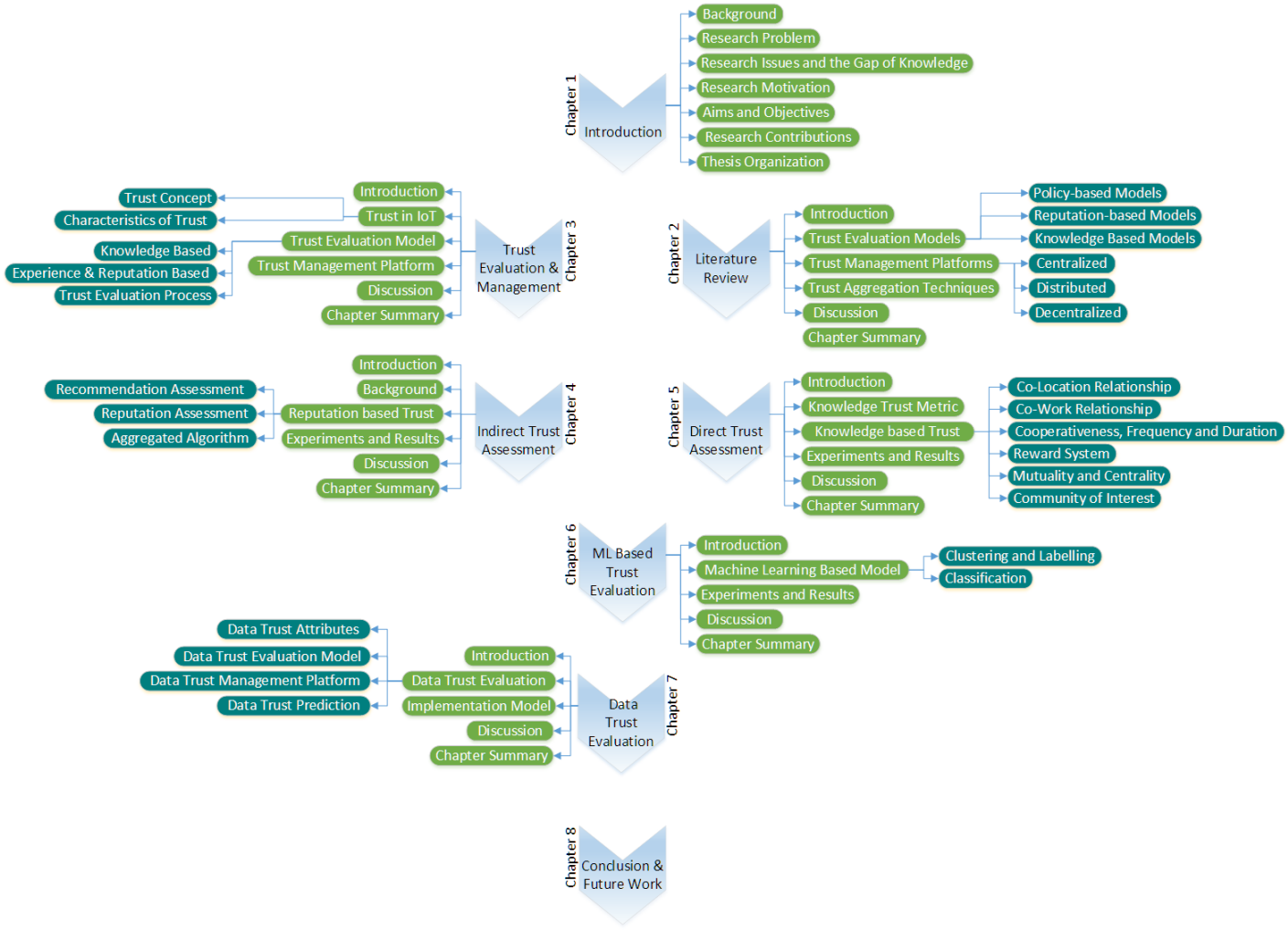


Figure 1-1: The Structure of the Thesis.

CHAPTER 2: LITERATURE REVIEW

2.1. Introduction

This chapter analyzes the existing work and solutions for all aspects of trust evaluation in an IoT ecosystem, mainly considering trust evaluation models and trust management platforms as shown in Figure 2-1. First, we demonstrate the merits, and demerits of the existing systems, and then provide evidence to stress the shortages of existing approaches in the context of providing futuristic trust-based solutions, which justify the incentive behind this research.

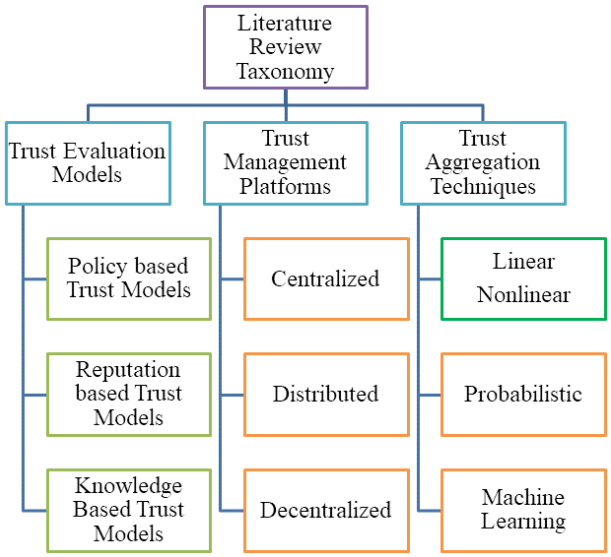


Figure 2-1. Taxonomy of the literature review.

2.2. Trust Evaluation Models

There are several common trust evaluation models, which can be observed in the literature like policy based (or rule-based approach), reputation based, knowledge based, based on aggregation method, application specific and based on intelligent learning algorithms. These trust evaluation models have been investigated under the context of different network environment including IoT with different purposes and goals.

2.2.1. **Policy-based Trust Evaluation Models**

Traditionally, policy mechanisms manage the decisions of a system by describing a pre-defined set of conditions (rules) and a specific set of actions in accordance with

each condition. In this manner, a policy can assist in making a decision for trust evaluation when a certain ambiguity level occurs while assessing trust. As a result, policy-based trust models normally involve the exchange or verification of trust-related credentials known as a trust negotiation process.

A high-level view of generic policy-based trust model is shown in Figure 2-2. To establish and calculate trust, trust management needs to integrate trust negotiation protocols for creating, exchanging, and managing the credentials of a network object. The policy-based trust methods generally assume that the trust evaluator would obtain a sufficient amount of credentials from a trustee and from other sources like the policy database for trust evaluation after several processes of credential creation and exchange. First, trustor would make a request from trust evaluator to access the trustworthiness of trustee. Upon receiving the request, trust evaluator might query trustee, policy base and other sources to evaluate the trust based on the already defined policies at each source as shown in step 2 of Figure 2-2. Once the evaluation process is completed, the decision is informed towards trustor by the trust manager as well as any update towards policy base. There is an issue called “recursive problem” which is related to the trust of the credentials in this approach. This problem can be solved by introducing a trusted authority (a third-party object) for issuing and verifying these credentials.

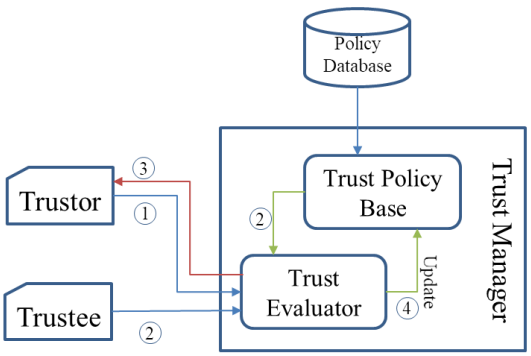


Figure 2-2. A high-level view of the policy-based trust model.

The policy-based trust mechanism is usually used in the context of distributed network systems as a solution for access control and authorization [4],[5], [6],[7]. The goal is simply to judge whether a user is trustful or not based on a set of credentials and predefined rules, before granting rights to access network resources. The focus in this

situation is how to apply policy languages for specifying and producing additional rules and trust knowledge for trust evaluation procedures.

For the summary of the research related to policy-based mechanisms, we organize the research work into sub-categories of trust evaluation procedures: trust credentials establishment, trust negotiation process, and policy/rules trust languages.

3.3.1.1. Trust Credentials Establishment

Conventionally, a credential is information about an object and context of the environment needed to evaluate trust. Although the word "credential" is frequently used in many research works, there is no common definition or standard to specify and determine it. Policies should rely on credential information and other context properties in order to judge trust. An obvious example of credentials in trust is the use of username and password to gain access control when logging into a computer. According to the system policy, having a correct username and password proves that the user is trusted by that computer system. In a more complicated example, credentials are also automatically generated during a negotiation process by leveraging security certificates with digital signatures or using Public Key Infrastructure (PKI). Note that only certificates that include trust-related information of an object or context can be used as credentials. For example, TrustBuilder [8] dealt with trust by establishing trust credentials using traditional security techniques such as authentication and encryption which is called "hard security" trust.

A well-known method of credential exchange is the Kerberos protocol [9]. The protocol considers a user as the trustee and a computer as the trustor and enables them to securely exchange their own verifiable credentials. For that, the Kerberos system needs to use a third party, in this case, another computer, to facilitate the credentials exchange process. However, this approach is no longer used since the current network systems like the IoT are much more complex and are facing many intelligent attacks.

Recently, many researchers have considered "credentials" in a broader perspective, and have used the term "trust metrics", and "trust attributes" instead of

"credentials". This approach allows us to develop more flexible, scalable, and effective trust models.

3.3.1.2. Trust Negotiation Process

An important issue when exchanging and generating credentials is unintentional information disclosure, which can result in a loss of security and privacy. The question raised is - to what extent an object trusts other objects to see its own credential information in exchange for earning their credentials. There are many research works dealing with this trade-off between gaining trust and sacrificing privacy such as in [10],[11],[12]. These researchers considered several particular contexts in accordance with the types and number of credentials. They analyzed the loss of privacy once any credentials are revealed to other objects. This trade-off approach has motivated some researchers to develop a trust platform by developing architecture systems based on that trade-off principle.

TrustBuilder is a typical example in which a mechanism is implemented for analyzing and choosing the reasonable solution for the trade-off in the context of web services [8]. The trustor needs to understand the risk of losing privacy information when revealing credentials in exchange for earning trust. Based on this mechanism, trust is gained when a successful trade-off is made, sufficient credentials are revealed while some level of privacy is still maintained. The concept of a trust transitivity property is also characterized in TrustBuilder in the form of a "credentials chain". For example, if object A trusts B's credentials and B trusts C's credentials, then A trusts the credentials of C to some degree.

Based on the credentials chain concept, some research works designed and developed trust frameworks that perform credential chaining and credential exchange such as in PeerTrust [13], PROTUNE [7] and RT10 [14]. Ontologies and Context-aware mechanisms are also soon introduced when developing credentials in the context of client-server systems [15] and Semantic Web [16].

3.3.1.3. Policy Languages and Trust Languages

It is necessary to design formalism for trust-related information, e.g. credentials and TMs in order to develop a trust system. This objective can be achieved by incorporating findings from logic to automate various kinds of reasoning, such as

the application of rules and policies or the relations of sets and subsets for the trust evaluation processes. Most researchers have used the Semantic Webs techniques such as semantic representation, policy languages, ontologies, and reasoning mechanisms to the trust evaluation. The issue is how to represent and express, trust information and trust knowledge. Some efforts have been made to create policy languages for trust as described in Table 2-1.

Table 2-1: A Comparison of Policy based models.

Research	Trust Context	Policy/Trust Language Features
KAoS [17]	Access Control for KAoS services	KAoS Policy language with the ability of dynamic policy changes.
Semantic Webs [18]	For Security and Privacy Issues	Use semantic representation and model for dynamic policy manipulation. Allow each object to set their own policy,
Global Computing system [19]	To replace key-based security	Include observation of trustee, recommendations from others and reference to other sources of the trustee. Use a formal policy language. Trust can be proved
Web services [20]	Specification and OASIS standard providing extensions to WS-Security	Security Assertion Markup Language (SAML). Trust is gained through proof of identity, authorization, and performance. To validate the security token.
Global Computing system [21]	For trust-based security mechanism	Policy language that uses lattices of relative trust values. Allows fine-tuned control over trust decisions
Cassandra [22]	Role-based access control and Context-based system for authorization	Use a policy specification language based on Datalog with constraints from five special predicates. Trust is obtained after credentials exchanged.
Open Distributed System, WWW [23]	Trust-based access control for web resources	Use ontology for representing trust negotiation policies. Rules are used to negotiate trust.
Policy Maker [24]	Trust-based authorization	Ability to provide “proof of compliance” for request, credentials, and policies. Allow individual systems to have different trust policies.

KeyNote [25] [26]	Trust-based authorization	Same principles as PolicyMaker[24]: directly authorize actions (in accordance with credentials) instead of processing both authentication and access control. Require credentials and policies be written in a specific assertion language to work with KeyNote compliance checker.

Policy based trust models often follow twofold approach for making trust decisions i.e. an object will be allowed or deny based on the credentials provided. Therefore, this type of models is more suitable for specifying who is allowed to access specific resources or services. One of the noticeable advantages of such mechanism is the absence of third parties to support the decision-making process. Hence, by design policy-based trust models show good resilient over privacy and credential leakage due to the involvement of third parties.

In contrast, it is practically impossible to discover and implement every possible rule and hence, the trustworthiness of the trust model totally depend on the accuracy and completeness of the policy database. The process becomes more complex if there is an involvement of either context, time and task. Further, in policy-based systems, trustees don't have the choice to select which information is allowed to disclose. That is, if the trustee cannot provide or not willing to provide the required credentials as they are not directly relevant to the expected service, policy-based trust model would still deny the request as the logic behind the model is based on the hard-coded rules.

2.2.2. **Reputation-based Trust Evaluation Models**

A reputation-based trust model is used in trust evaluation for assessing trust score or trust level based on the history of interactions of related objects. The reputation information in this scenario could be either directly with the evaluator (direct reputation) or a a recommendation by other objects (indirect reputation, recommendation, or third-party information). As defined by [27] reputation is a measure that is derived from direct or indirect knowledge on earlier interactions of objects and is used to assess the level of trust an entity puts into another entity. For example, Figure 2-3 shows the reputation-based trust evaluation process between a trustee and trustee. After each interaction, each trustor records his satisfaction over the interaction and transmit it towards reputation system as a feedback. Then, the

reputation system accumulates all such feedbacks and generate one reputation value for the specific trustee and send it towards trustor for decision making.

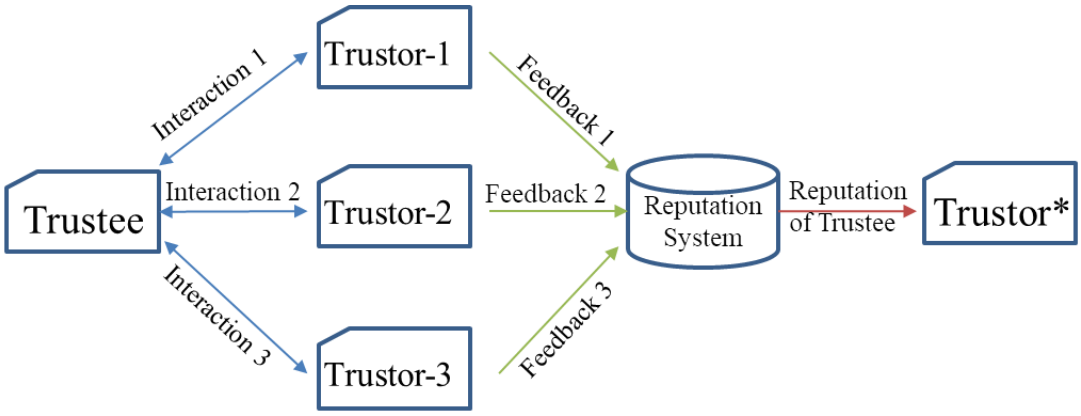


Figure 2-3. A high-level view of the reputation-based trust model.

In recent years, most researchers have accepted that reputation is one important factor of trust, resulting in the dominance of reputation-based trust models compared to policy-based models. Some have tried to integrate both approaches in their trust models in order to leverage the advantages of them. Nevertheless, both credentials and reputations are the important information involving in the trust transitivity among objects, and each of them has its own pros and cons that have motivated researchers to work on.

There are many parallel research works on both the reputation-based trust model and the reputation model. The confusion between a reputation system and a trust system should be clarified. Basically, a reputation system collects feedback from objects after an interaction incurs. This feedback will be combined and calculated using several mathematical models to get a reputed score. This reputed score is sometimes misunderstood as a trust level. Several reputation systems have been developed in the context of e-commerce systems and web services such as eBay [28] and Keynote [25], [26]. These systems use a centralized authority to get ratings and feedback from users after each transaction and then update the overall reputed score by using several mathematical models as mentioned above. In some distributed systems, each object establishes and maintains reputed scores of its neighbor s by using several heuristic algorithms like [29], [30].

Reputation-based trust systems can be considered as a step forward compared to a reputation system in which a trust evaluation mechanism combines not only ratings or

feedbacks from objects but also trustor and trustee properties and preferences; and context information to calculate trust level. In this sense, the reputation system is a part of the trust system. There has been a large amount of effort to investigate the reputation-based trust model and to develop reputation-based trust systems in many types of network environments such as distributed systems, P2P networks, sensor networks, and grids. There are also some research works seeking to build a network of trust in which trust is established and maintained between any two objects over time, resulting in creating a "web of trust".

3.3.2.1. Reputation-based Trust in Distributed and P2P Networks

The trust models in this category are capable of establishing trust, calculating trust levels, and making trust decisions in the absence of a centralized authority. The contribution in this approach is how to create appropriate credentials, TMs, and TAs that are provided to each object to produce trust. Depending on the different purposes of applications in each network environment, reputation-based trust systems are utilized accordingly. For example, in a distributed system, many research works focus on the detection of malicious objects and the prevention of network attacks, while trust systems in P2P networks focus on guaranteeing the quality of data transfers.

3.3.2.2. Reputation-based Web of Trust

Reputation, in this scenario, is defined as a TM, and each object maintains reputation information on other objects, thus creating a "trust network" or "web of trust". There are two approaches to trust systems in the web of trust. The first approach assumes that trust credentials and TMs already exist, and the trust systems are trying to propagate trust among objects, which may not have been evaluated for trust. The latter supposes that a web of trust is given, in which a link between two objects means the trust decision with a trust value. It does not matter how these links are made, just as long as the trust can be quantified. If there is no link between two objects, it means no trust decision has been made, and trust transitivity should be applied in this scenario. The summary and comparisons of reputation-based trust evaluation are briefly described in Table 2-2.

Table 2-2: A Comparison of Reputation based models.

Research	Trust Context	Reputation-Related Features
Distributed System [31], [32]	Malicious Node detection	Define Agent, Trust Relationships, Trust Value, and Trust Categories. Define first-hand knowledge as direct reputation and second-hand knowledge as a recommendation. Proposes a Recommendation protocol for trust propagation.
Distributed System Social Network [33], [34], [35]	Reputation Management	Reputation information is obtained from external sources. Local objects calculate, maintains, and make trust decisions without a centralized trust management system. Sources, who provide reliable reputation information, are weighted accordingly.
Social Networks Multi-agents system [36]	Reputation System	Analyze the reputation information by characterizing the indirect and direct information. Considers the social relation in calculating reputation score. Put the context information into account.
Open Networks [37]	Trust-based authentication	The algorithm is capable of estimating the degree of trust in the presence of conflicting information.
P2P Networks [38], [39]	Webpages ranking	PageRank algorithm is used to rank the web pages by the authority. As an example, the EigenTrust algorithm uses PageRank concepts to calculate a global reputation value for each object.
P2P Networks [40]	Reputation System	Best object for a given resource is determined by a protocol called XRep, which is governed by the users' feedbacks.
Web of Trust [41], [42]	TrustMail application	Trust and reputation information is first categorized by ontologies and then this knowledge is used to quantify trust to make a trust decision about any two objects. Trust transitivity is considered as a credentials chain. Local reputation and Global reputation is also taken into account.
Web of Trust P2P Network [43], [44]	Trusted applications in Open Network	The algorithm calculates a global reputation value considering both global opinions as well as local reports. Furthermore, it discriminates each object's reputation reports based on their applicability to the context and reliability.

In reputation-based trust models, trustees' trustworthiness will be evaluated based on the opinions from the objects who had previous interaction with the same trustee. Hence trust values or the rating scan have multiple levels in contrast to binary decision making in policy-based trust models. As the decision making is based on the accumulated trust value, honest and trustworthy objects will have more opportunities in the network and dishonest objects will be discouraged from future interactions. Further, it is not required to disclose credential information of the trustees to evaluate the trustworthiness, which prevents the risk of compromising private information.

However, the whole system depends on the honesty of the feedbacks and fake reputations or threats discussed in Section 1.1.1.3.b can jeopardize the objective of the model completely. Hence it is important to validate the feedbacks before using them and encourage feedbacks from most trustworthy objects only. However, existing work lacks such mechanisms and have completely ignored unique properties of trust like subjective and context-dependent nature of trust.

2.2.3. **Knowledge Based Trust Evaluation Models**

To understand trust, it is required to analyze the collected data from objects, extract the necessary information for trust; understand the information and then create the trust-related knowledge for the trust evaluation. Knowledge is the first party information provided by the trustee to evaluate its trustworthiness and composed by some TAs depending on services and objects. It leverages the direct trust evaluation and is comprised of two major tasks: (i) specify a set of TAs for the trustee's trustworthiness that reflects the trustor's propensity and the environmental factors; and (ii) an aggregation mechanism to combine these TAs for deriving the direct trust as the Knowledge TM value. Notably, Husted [45]and [46] defines knowledge-based trust models as "Deals with the ability to predict the behavior of the trustee based on prior performance" and "The trust developed through repeated interactions that allow an individual to collect information about the other and develop an expectation that the other's behavior is predictable" respectively.

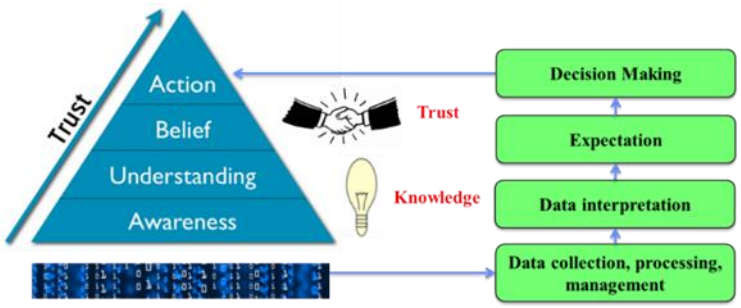


Figure 2-4. A high-level view of the knowledge-based trust model.

According to the above definitions, a high-level view of such model is shown in Figure 2-4 [47]. The knowledge is accumulated by individuals or objects through data analytics over time. So far data processing, management and interpretation for awareness and understanding have been considered as fundamental processes for obtaining the knowledge. Upon obtaining knowledge, trust can be evaluated by connecting knowledge and expectations of the trustor as shown in Figure 2-4.

Social interactions among objects disclose the valuable information of trust in relation to the sociological concept of human interactions based on trust relationships. In this regard, authors in [48] and [49] have developed a social model of cyber objects corresponding to their owner's social behavior. In such models, objects interact with each other based on their trust relationships and reveal information in terms of trust as described in [50], [51]. Moreover, [52] and [53] discuss the trust assessment of a social network based on concepts such as a CoI, friendship, followers, as well as frequency, duration and behavior of the objects. In a similar manner, authors in [54] and [55] present a computational model for trust based on similarity, information reliability, and social opinions.

In the aspects of knowledge extraction from web sources, [56] propose a new approach based on endogenous signals, which essentially describe the correctness of the factual information provided by the source; sources with few or no false facts are recognized as trustworthy sources. These endogenous signals are characterized by knowledge triples in the format of (subject, predicate, object) from web pages. A subject represents a real-world object, identified by an ID, and predicate defines a particular attribute of the object. Authors in [57] and [58] further expand upon the above research and compare several knowledge triples before estimating the final trust values.

On the other hand, Finite State Machines (FSM) are used to represent and store the knowledge about the present and past behaviors of the systems. Knowledge triple in an FSM is represented by a set of states, transitions, and actions. The advantage of using this method is the ability to detect abnormal behaviors without any training data or signatures. In this regard, authors in [59], [60] propose a protocol to detect attacks in ad hoc networks and [61] propose an algorithm to detect abnormalities in system calls.

As already identified through the literature, knowledge-based trust models are work based on the actual knowledge of the facts that constantly obtaining from ongoing or immediately completed interactions. That makes such a model is less prone to fluctuations of biased and discriminative efforts by the malicious objects in the process of trust evaluation for example as in reputation systems. This motivated to carry out most of the work in this thesis based on the knowledge-based trust.

However, irrespective of its capability to be a good candidate for trust evaluation, in the abundance of knowledge pool, current mechanisms are mot intelligently enough to filter correct and appropriate facts to evaluate trust while considering the trustor's subjective perception as well. Moreover, evaluation of trust based on knowledge has already become a technical bottleneck, particularly in the distributed systems due to limited resources at the edge of things to process big amount of knowledge. Moreover, knowledge bases often contain privacy related information and accidental or intentional exposure of such knowledge may lead unexpected outcomes, ranging from uninvited advertisements to identity theft. Hence it is a must to implement mechanisms and regulations to up rise the trustworthiness of trust management platforms.

2.3. Trust Management Platforms

There have been many proposed trust management platforms for different types of networks such as mobile ad-hoc networks, wireless sensor networks, peer to peer networks and social internet of things. Authors in [62] first introduced the term “Trust Management” and identified it as a separate component of security services in networks and clarified that “Trust management provides a unified approach for specifying and interpreting security policies, credentials, and relationships”. Moreover, authors in [63] define trust management as “Collecting the information required to make a trust relationship decision; evaluating the criteria related to the trust

relationship as well as monitoring and evaluating existing trust relationships; and automating the process" which is one of the inspiration to design and implement trust management platform for IoT environment in Section 3.5. Furthermore, authors in [64] recognize several specific goals that a trust management in IoT should achieve as below:

- Trust relationship and decision (TRD): trust management provides an effective way to evaluate trust relationships of any two entities and assist them to make a wise decision to communicate and collaborate with each other.
- Data perception trust (DPT): data sensing and collection should be reliable in the trust management system.
- Privacy preservation (PP): user privacy including user data and personal information should be flexibly preserved according to the policy and expectation of IoT users. This objective relates to the IoT system objective properties in general.
- Data fusion and mining trust (DFMT): the huge amount of data collected in IoT should be processed and analyzed in a trustworthy way with regard to reliability, holographic data process, privacy preservation, and accuracy.
- Data transmission and communication trust (DTCT): data should be transmitted and communicated securely in the IoT system. Unauthorized system entities cannot access private data of others in data communications and transmission.
- Quality of services (QoS): Quality of services should be ensured.
- System security and robustness (SSR): trust management should effectively counter system attacks to gain sufficient confidence of system users.
- Generality (G): trust management for various systems and services is preferred to be generic that can be widely applied, which is a system objective property.
- Human-Computer Trust Interaction (HCTI): trust management provides sound usability and supports human-computer interaction in a trustworthy way, thus can be easily accepted by its users.

- Identity trust (IT): The identifiers of system entities are well managed for the purpose of trustworthy. Scalable and efficient identity management in is expected.

However, most of the trust management mechanisms discussed in the literature including the ones stated in [64], [65], [62], [66], [67], [68], [69], [70], and[71]are basically explain how trust can be modeled but not the actual process of trust management from data collection to trust information lifecycle management as stated in Section 3.5. Yet, depending on the nature of trust propagation trust management platforms can be classified as below.

With heterogeneous applications and services in the IoT, one must give special attention to the architecture of the trust model with respect to trust propagation. According to the literature, studies on trust architectures can be mainly divided into three groups; centralized, decentralized and distributed approaches. Some properties of each approach are described in Table 2-3.

Table 2-3: Comparison of Trust Propagation Methods.

Property	**Centralized**	**Decentralized**	**Distributed**
Points of failure	Single point of failure	Finite number of failures	Infinite
Maintenance	Easy	Moderate	Difficult
Stability	Highly unstable	Recovery possible	Quite stable
Scalability	Low Scalability	Low Scalability	Infinite
Development	Less Complex	Moderate	Complex
Diversity	Low	High	High

In the centralized approach, a centralized platform manages all aspects of trust management including the information about TMs, TAs, protocols, algorithms and mathematical models, and provides the service on demand as shown in Figure 2-5(b). On the other hand, in the distributed approach Figure 2-5(a), TAGs do all the necessary computation locally.

(a) Distributed (b) Centralized (c) Decentralized

Figure 2-5: Centralized vs Decentralized vs Distributed Networks.

However, for IoT applications, sticking to only one approach will not be sufficient as sometimes calculations have to be done locally and some remotely, depending on the resources' availability. Therefore, the fully distributed model or the fully centralized version will not give satisfactory results and hence an alternative approach, which is in between the centralized and distributed approaches should be considered. In this regard, the decentralized model is shown in Figure 2-5(c) can be considered as an optimum model for the trust evaluation with the complexity of IoT services.

2.3.1. **Centralized**

In this approach, each trust request and service will go through a centralized platform, which can be accessed by all other nodes in the domain. This platform is responsible for managing trust information including trust negotiation, calculation, and decision making and/or assisting users by providing the initial information required for trust evaluation.

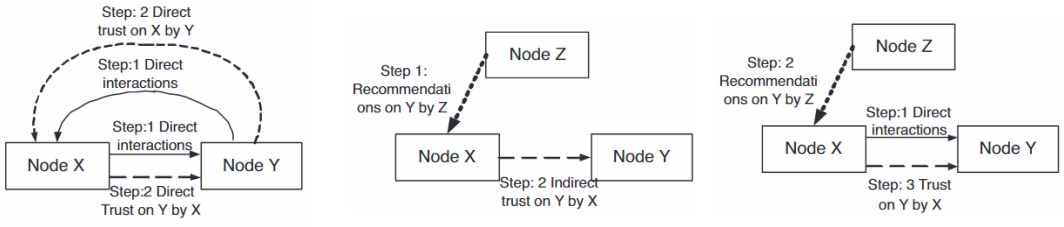
In general, centrality based rating systems are global rating systems. One of the most prominent areas where centralized trust evaluation has been deployed is in the social networks like Facebook™ and e-markets like Amazon™ and eBay™ [72], [73]. In these, reputation is a function of the cumulative ratings of users by others. Furthermore, [74] explains how the reputation system works in social networks using a mathematical model. It introduces the adjacency matrix, which represents ratings from node "i" to node "j". Then a recursive based method is followed to solve this matrix and obtain the reputation score for each reputed user.

A more evolved version of a reputation model called SPORAS, compared to eBay™ is developed by [75] in which only the most recent recommendations have

been taken into the consideration. Here, the mechanism is built in such a way that the reputation update will affect low reputed users significantly and high reputed users minimally. The underlying core principle is based on the standard deviation of reputation values. In addition, the authors suggest a method to incorporate reputation mechanisms in online communities, to make it more reliable and more effective in the ways that users contribute to the community. In [76], trust evaluation based on a centralized cluster head is proposed. Initially, the cluster head is responsible for delivering trust values for every node in its domain. After that, each local node will combine locally calculated trust with the initially learned trust value from the cluster head.

2.3.2. **Distributed**

In distributed trust evaluation methods, every node is supposed to calculate trust locally by observing and exchanging reports with the neighboring nodes. For example, a trustor node might estimate the final trust value of the trustee by comparing its own reports, based on either direct trust measurements, reports from the trustee or reports from other global peers. This is illustrated in Figure 2-6 as neighbor sensing, recommendation based trust, and hybrid trust.



(a) Direct Trust : Neighbor Sensing

(b) In-direct Trust: Recommendations

(c) Hybrid Methods

Figure 2-6: Distributed trust evaluation methods.

Trust estimation based on direct trust as well as from neighbor reports is proposed in [77], in which a mathematical model is formed based on probability theory to determine optimum percentages from both objects. A direct trust evaluation method for wireless sensor nodes is proposed in [78] based on the confidence interval concept. Final trust value will be decided after observing the behavior of adjacent node over a considerable time. Here, trust is represented as the mean trust value and a confidence interval about the mean. Then, based on the confidence interval, the trustor will proceed with the decision-making process, i.e. if the

confidence interval is sufficiently narrow enough. If not, the trustor will acquire more knowledge from the trustee before calculating the final trust value.

In a situation where direct observation is not possible with a trustee node, trustworthiness can be calculated based on recommendations from the peer users, which have records about the trustee. However, relying on others' recommendations involves a high level of risk compared to the direct trust method. This is because recommenders can falsely provide dishonest information, which can lead to a reduction in the trust value of honest users and improve the trust of malicious nodes. Therefore, other than calculating trust, validating them is also a key research area when direct trust information is not available.

Concerning dishonest users, [79], [80] propose trust credibility evaluation methods based on threshold values and assigning lesser weights for the dishonest users in future transactions. After filtering out the false recommendation, the next step is to calculate the effectiveness of each honest recommendation. Authors in [81] proposed several methods to determine the credibility of trust by using fuzzy logic. A trust calculation method based on threat reports for MANETs is proposed in [82]. In this method, an alarm system is included in each node. Then every node listens to its adjacent nodes and generates a trust report based on their behavior. This will be broadcast to every node so that if any node generates a false report it can be detected by the alarm system.

2.3.3. **Decentralized**

A decentralized trust mechanism is an alternative model to both distributed and a centralized architecture that combines the positive points from both the previous designs. In the centralized approach, the trust information can be computed on demand, whenever an object needs to rely on its cooperative objects and delivered to the requesting object at that moment. On the other hand, the distributed approach can compute trust on a regular basis and propagate this throughout the topology. However, it is a concern that an object itself in a large-scale network such as the IoT possibly lacks the knowledge to evaluate trust. It certainly needs help from others such as trusted authorities. Moreover, a real-time trust data flow would result in a communication overhead, detrimental to network performance as well as to a constrained object's battery life. The traditional strategies for centralized

systems are unsuitable for solving trust issues in a large-scale distributed network like the IoT because of their poor scalability as well as center-dependence, leading to a single point of failure. Thus, we conclude that a decentralized system would be the best candidate for trust management in a diversified environment like the IoT. This is because it possesses desirable properties such as scalability, the ability to produce accurate global trust, efficient resource management, reduced network overhead, and dynamic adaptability.

In this regard, the authors in [83], [84] propose a trust model based on grouping several nodes together depending on their capability level. Then, each group selects a cluster head and all the cluster heads are connected to the base station. Then based on the reports from neighbor s, the cluster head will determine the trust value of other cluster heads based on interactions and then forward all the information to the base station. Finally, the base station compares all of the reports from all of the cluster heads and calculates the final trust value and sends it back to the cluster heads if needed.

Blockchain technology, which is one of the most famous decentralized systems, introduced by S. Nakamoto [85], is a strong candidate for implementing trustworthy decentralized systems. In this regard, Foutiou et al. [86] propose a decentralized secure access control mechanism with the help of blockchain concepts. Similarly, the authors in [87] propose a trust management system for authentication based on the blockchain technology and analyze its resilience over common network attacks. Therefore, it is certain that blockchain is very useful as a decentralized technology in the context of futuristic trust based applications in the IoT.

2.4. Trust Aggregation Techniques

Several research works have tried combining multiple models in order to leverage their advantage whilst attempting to mitigate their drawbacks. This idea has recently become more popular in the context of the IoT, where trust is more complex because many factors contribute to the trust establishment and trust evaluation. In such IoT environments, the history of interactions and behaviors of objects is not only for reputation information but also for trust-related knowledge extraction. The

combination of reputation information, knowledge, and relationships among objects in the IoT draws a very complicated picture of trust evaluation.

Table 2-4: Summary of Trust Aggregation Techniques.

Aggregation Techniques	Important Features
Weighted Sum [65], [88]	Higher weight is assigned to objects with higher reputation or transaction relevance. Thus, objects with strong relationships to trustor have a higher weight. Use credibility or similarity as a weight for indirect trust aggregation.
Fuzzy Logic-based [89], [90]	Fuzzy Logic provides trust level in the form of vague terms such as "low-high", "good-bad", and "acceptable-unacceptable" rather than providing an exact trust value.
Belief Theory [79], [91]	Belief theory (evidence theory or Dempster-Shafer theory (DST)) deals with reasoning using uncertainty and has connections to other techniques such as probability, possibility, and imprecise probability theories. Used in trust computational model to compute the trust of agents in autonomous systems by modeling the trust by belief, disbelief, and uncertainty of an object to other objects.
Bayesian Methods [92], [93]	Trust can be considered as Bayesian interference and can be modeled as a random variable in the range of [0, 1] following Beta distribution in which Belief discounting can be applied to defend against malicious objects such as bad-mouthing attacks ballot-stuffing attacks.
Machine Learning [94], [95]	Trust evaluation is formalized as a classification problem and a novel approach utilizing a machine learning method is presented. Firstly, the trust feature vector is constructed according to the trust related factors. Then by training with collected sample data which contains trust feature vectors and trust ratings, a trust classifier is established.

TMs can be gained from sufficient TAs by using trust aggregation techniques, for example, TMs can be computed by using Weighted Sum [65], [88], Fuzzy-based algorithms [89],[90], Belief Theory [91],[79], or Bayesian mechanisms [92],[93]. On the other hand, the authors of [94] and [95] outline the requirements for robust probabilistic trust assessments using supervised learning and apply a selection of

estimators to a real-world dataset, in order to show the effectiveness of supervised methods. Another interesting work that applies ML techniques is found in [96]. In this work, they propose to use neural networks in order to provide a global reputation model using the distributed reputation evaluations. The global reputation is determined by the neural network's output unit, a two-class classification in this case. Furthermore, authors in [97], [98] and [99] investigate more innovative models and solutions for privacy, security and data integrity based on statistical and Deep Learning (DL) concepts. Moreover, authors in [100] and [101] propose a regression based model which compares the variation of trustworthiness with respect to trust features in MANET and Wireless Sensor Networks (WSN). Recently, authors in [102], [103] and [104] have presented several trust management frameworks based on Reinforcement Learning (RL) and multiclass classification techniques. To calculate the overall trust score or trust level, a mechanism with one of the trust aggregation methods mentioned above is needed to combine those TMs.

Note that trust aggregation is a dynamic process that heavily depends on context-aware information, service requirements, and trustor's preferences. Each trustor needs appropriate trust data, context data, and aggregation methods for producing an overall trust score that reflects the trustor's perspective and context awareness. Specific trustors might use and define different trust aggregation techniques for dealing with their associated trust data. There is currently no complete trust aggregation mechanism that can deal with the personalized trust in a dynamic context-awareness environment, however, several researchers have proposed some solutions for particular contexts and services. The summary of such aggregation models is described in Table 2-4. The trust aggregation techniques are the crucial parts needed to investigate and develop in order to build a completed trust management platform in the IoT.

2.5. Discussion

Based on many existing works which have been analyzed above, there are many gaps that needed to be filled. Further, it can be observed that most of the existing solutions are concentrated on solving issues either in particular areas of applications like P2P networks, MANETs, WSNs, etc. or particular service aspects like authentication, security, access control, etc. Besides, some works incorrectly used the terms trust and

security interchangeably. In summary, Table 2-5 shows the gap of knowledge identified by the thesis.

Table 2-5. Gap of knowledge of the research area.

Research	Gap of Knowledge
▪ Trust definitions	
Security as Trust [4], [5] Privacy as Trust [10], [105] Policy as Trust [17], [18] Reputation as Trust [28], [29]	- Conflicting definitions and interpretations of trust
▪ Choose of metrics and attributes in trust modeling	- Trust models mostly design to identify dependability issues
Availability : [77], [106] Reliability : [107], [108] Safety : [109], [110], [81], [111] Security : [85], [86], [87]	- Trust estimation based only on limited features
▪ Target Environment	
P2P, WSN and MANET [112], [113], [38], [39], [40]	- Mostly aligned with cyber-physical systems (CPS)
Ad Hoc [100], [101] / M2M [102] / MANET [104]	- No comprehensive trust management platform defined for IoT
▪ Trust modeling and aggregation	- Trust aggregation based on fixed weighting factors - Identification of trust boundaries based on predefined thresholds - Unsuitability of using predefined probabilistic distributions to represent the dynamic property
Weighted sum [65], [88] Fuzzy logic [89], [90] Belief theory [79], [91] Bayesian [92], [93]	
▪ -AI in Trust prediction	- Often based on synthesized data sets as there are no real data sets based on trust - Mostly depend on privacy, security and reputation data - Absence of model validation
Support vector [94], [95] Neural networks [96] Deep learning [97], [98]	

One of the most important gaps that we intend to discuss and go for doing research is the lack of using environment information to trust evaluation. In the IoT environment, physical devices are owned by human-related factors and inherently socially connected by the physical-cyber-social system. Moreover, trust evaluation methods also lack concerns about trustor's subjective properties, in other words, the trust results are not reflected of personalized expectation. The solutions for this gap could be two-fold approaches: The first one is to develop the trust relationships among entities in the IoT, thus creating a reliability and readiness of the trust network, based on the existing social models in the network systems. The second one is to explore other social TMs such as trustor's similarity and friendship behaviors, centrality, community of interest, and more appropriate reputation TM.

Along with the two approaches, trustor preferences should be considered to reflect the personalized trust and to enhance the intelligence of trust. There are a large number of TMs depending on each context of IoT and services requirements such as honesty, cooperativeness, QoS, community of interest, etc. In order to explorer more TMs, it is needed to investigate the network environment ontologies and trust ontologies in which relationships among entities and the relationships' properties are represented and clarified. Consequently, by using a reasoning mechanism or a machine learning technique, new trust information and trust knowledge could be extracted and help to enhance the effectiveness of trust evaluation.

Another big gap in the area of trust evaluation is the trust aggregation methods and trust reasoning that have been stated in the chapter. This gap incurs in both situation in the trust evaluation procedure: when there are several distinct TAs needed to combine into one overall TM, and when there are several TMs needed to combine into the overall trust score or trust level. There are limited literature in this area as mentioned in Section 2.3.2 The most popular and simple method to deal with the trust aggregation and trust reasoning currently is to apply the use of static weighted sum for trust formation. However, this solution is not smart enough due to the complicated IoT environment. Thus, there is an urgent need for a novel research on the use of more effective trust formation methods including dynamic weighted sum, belief theory, fuzzy logic, and regression analysis. For example, an intelligent weighted sum method can dynamically adjust the weights associated with TA and TMs based on context awareness and user preferences. The weighted sum method can also use a regression

analysis that links context information with TA and TM and user preference so as to determine the best weight assignment

The scope of existing standards on security and privacy need to be expanded to include trust issues in future IoT Infrastructures. As existing research and standardization activities on trust are still limited to the social trust among humans, trust relationships among humans and things as well as among cross domains of social cyber-physical worlds should also be taken into account for trustworthy autonomous networking and services in IoT environments.

2.6. Chapter summary

This chapter summarized the latest advances and applications of trust solutions for the IoT through the various types of references including scientific publications, textbooks, and online articles. The thesis identifies and categorizes existing work on trust into three main categories based on their approaches on realizing computational trust in a digitized environment as (i) Trust evaluation models; (ii) Trust management platforms; and (iii) Trust aggregation techniques. After that, positives, negatives, and gap of knowledge of current methods are investigated.

CHAPTER 3: TRUST EVALUATION AND MANAGEMENT

3.1. Introduction

The concept of the IoT, which has made many unthinkable inventions possible, has been a breakthrough in the past decade and many more are expected in the years to come. In an IoT infrastructure, billions of electronic devices are connected to the Internet and these devices are equipped with sensors that observe or monitor various aspects of human life in the real world for supporting more ubiquitous and intelligent services. A modern-day IoT ecosystem involves the networking among physical devices and cyber components as well as the social interactions of them. This is essentially a leap forward of CPS and the formation of CPSS to connect the Cyber-Physical world with social world objects [1]. Based on the CPSS concept, the new IoT model, which incorporates social paradigms into the IoT ecosystem, is introduced to explain the social behavior of objects along with human interactions [2].

However, this integration introduces new concerns for risks, privacy, and security at both the system and social levels because of heterogeneous interactions among humans and objects. Consequently, managing risks and securing IoT is broader in scope and pose greater challenges than the traditional privacy and security triad of integrity, confidentiality, and availability in the physical and cyber world. The aim of future IoT services is to make decisions autonomously without human intervention. In this regard, trust has been recognized as a vital key for processing and handling data, and for complying with the services, business, and customer needs. Accordingly, ITU-T has been developing related standards for trust provisioning after publishing the first recommendation [3] based on the activities of the Correspondence Group (CG) on Trust. For supporting trust, it is crucial to minimize unexpected risks and maximize risk predictability using a trust platform. This platform should help the IoT infrastructure to operate in a controlled manner and to avoid unpredicted conditions and service failures.

There are several trust related frameworks that can be observed from the research literature, such as [97] and [114] based on privacy, [5] and [21] based on reputation, and [115] and [116] based on social relationships. On the other hand, there are some

frameworks which are aiming at a particular application area like ad-hoc networks [117], P2P [118] or social networks [48]. However, these approaches lack generality in terms of the application domain and target area. Therefore, it is essential that trust mechanisms are designed and developed to look ahead to the future where many individual objects are interconnected with new vulnerabilities possibly being introduced into heterogeneous systems and application domains.

Thus, this chapter first proposes a generic definition for trust to avoid any ambiguity on trust provisioning. Then it proposes a trust model and a trust management platform to find viable solutions to trust related problems in any environment including IoT.

3.2. Trust in IoT

3.2.1. Trust Concept

Typically, trust can be observed as a metric used to evaluate social actors in consideration of mutual benefits, coordination, and cooperation. Actors continuously update their trust on others in response to perception fluctuations due to direct interactions and based on beliefs and opinions of others who are around. Furthermore, trust also affects the decision of an object to transact with another object in an IoT ecosystem in which all participating objects must take decisions based on trust to provide/receive services to/from other objects.

However, building trust in IoT is much more difficult due to the inability of machine objects to generate perceptions about other objects around them like humans. Furthermore, it is difficult to quantify the exact trustworthiness value of an object with a high accuracy. This is even harder when each object has a different interpretation and perception of the term “trustworthy”. Therefore, they may assign different trustworthiness values to a provider or a service. As an example, a service consumer object assigns “very trustworthy” to the provider for a specific transaction that it has performed. However, another consumer object might assign “untrustworthy” for a similar transaction from the same provider. These differences further increase the difficulty to determine the exact trustworthiness of a provider.

Therefore, it is essential to establish a generic platform which defines the blueprint of a trust management process while keeping in mind the diversity of trust features and hence the flexibility given to objects to choose best and practical measures. To clarify

the ambiguities and definitions of trust, we use the following definition in the context of the IoT [119]:

Definition 1. *Trust*

It is a qualitative or quantitative property of a trustee, evaluated by a trustor as a measurable belief, in a subjective or objective manner, for a given task, in a specific context, for a specific time period.

The term "trustor" is used to represent an object that is expected to initiate an interaction with another object and "trustee" as the second object that provides necessary information to the trustor upon its request. The first thing that we want to emphasize in the definition of trust is the nature of the measurement that can take either a quantitative or a qualitative form. Apart from the well-known numerical measurements like similarity, accuracy, etc., qualitative properties like motivation, awareness, and commitment can also be used to judge certain situations in the process of trust based decision-making. In addition, it is important to recognize trust as a belief even in the cyber world. That means, trust is a relative phenomenon and 100% belief is neither practical nor achievable in a diverse environment like the IoT.

Moreover, the perception of trust can be either subjective or objective, depending on the requirement of the trustor and the availability of needed information. If the trustor wants TMs in a specific format that goes with the trustor's profile of interest, then the measurements can be characterized as subjective. On the other hand, objective measures can be described as TAs collected without any profile based filtering. Lastly, it is very important to define trust specifically for a particular task, context and time frame. For example, one might trust another for their cloud storage services but not for online streaming services, i.e. task dependent trust. Furthermore, one might need the service of a cloud only for a limited duration and not for a lifetime, i.e. time-dependent trust. Moreover, a client might use different cloud services in different countries, as he does not trust the same provider globally, i.e. context dependent trust. Hence, trust is variable in nature and hence cannot be assigned permanently to measure every task and every context of a specific actor or object. Further, we need to emphasize that trust is a relative quantity between two or more objects as opposed to a measurement of individual objects.

3.2.2. **Key Characteristics of Trust**

With the above understanding of trust, the following characteristics further explain the nature of trust in an IoT ecosystem [120-122].

Trust is Dynamic: Validity of trust is dependent upon time and may change as time goes by. For example, for the past one year, Alice highly trusts Bob. However, today Alice found that Bob lied to her, consequently, Alice no longer trusts Bob.

Trust is Context-dependent: Trust is context dependent and whenever the context change happens, trust must be recalculated accordingly. Additionally, the level of trust in diverse contexts is evaluated differently. For example, Alice may trust Bob to provide financial advice but not medical advice.

Trust is not Transitive: That is trust cannot be passed from one to another and must be recalculated at each object. For example, Alice cannot simply trust Charlie, just because Bob, who is a trustworthy friend of Alice, trusts Charlie.

Trust is Asymmetric: A trust relationship works only in one direction and it is not mutually reciprocal in nature. That means there is no guarantee that Alice trusts Bob, even though Bob trusts Alice.

Trust is Implicit: It is always hard to explicitly calculate the trust with 100% accuracy. For example, Alice may trust Bob normally because they have known each other for a long time. However, sometimes, Alice's estimated trust on Bob might not be 100% correct. Therefore, this might affect Alice in a negative way and would not be able to achieve expected results all the time.

Antonymy: The trust assessed by different objects on the same service might not be essentially the same, it differs depending on each one's perspective. For example, object Alice trusts Charlie in the context of teaching, however, Bob's view on the same matter might be different from Alice's.

Asynchrony: The time period of a trusting relationship may be defined differently between the objects. For example, Alice has trusted Bob for three years, but Bob may think that their trust relationship has lasted only one year.

Gravity: The degree of seriousness in trust relationships may differ between the objects. For example, Alice may think that their trust relationship with Bob is important for her, however, Bob might think otherwise.

3.3. Trust Evaluation Model

Having stated the generic definition of trust and its features, the next step is to define the course of trust acquisition, aggregation, and representation in a conceptual setting. In the human world, one's judgment on others is basically dependent on three factors; (i) How much the trustor knows about the trustee (i.e. Knowledge); (ii) Trustor's previous experience with the trustee (i.e. Experience); and (iii) Public opinion on the trustee (i.e. reputation). Analogy to the human world, this work adopt these facts on the IoT environment and formulate a trust evaluation model called REK based on Reputation, Experience, and Knowledge as defined below and shown in Figure 3-1.

Definition 2: *Trust Evaluation Model*

The method used to identify, evaluate and create trust relationships among objects for calculating trust. It comprises three TMs: Knowledge, Experience, and Reputation. Each TM is a collective representation of several TAs. Each TA represents the trustworthiness feature of a trustee.

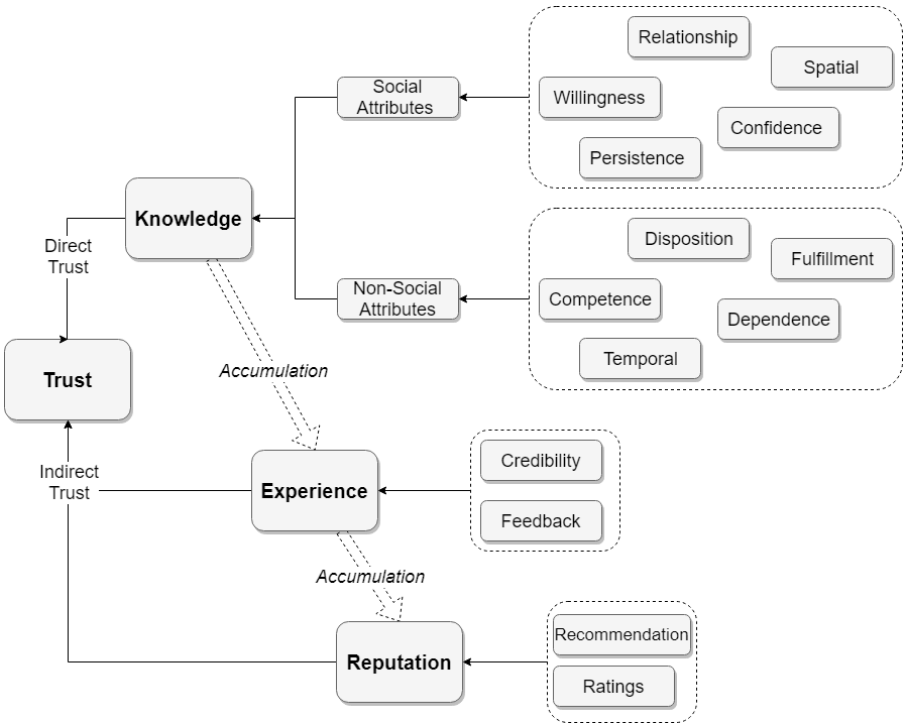


Figure 3-1: A Generic Trust Evaluation Model.

3.3.1. **Knowledge based Trust Metric**

The knowledge based TM covers all aspects of direct trust evaluations, which provide a perception about a trustee before and during an interaction. To make this possible, it must provide relevant data to the trustor for its assessment. If a data feature can be represented using a quantitative measurement, then the result is a numerical value in a certain range. As an example, social relationships like co-location and co-work, credibility factors like cooperativeness, time-dependent features like the frequency and duration of interactions, and spatial distribution of relevant trustees compared to the trustor can be used as direct trust measurements.

The main purposes of trust assessment are to facilitate more intelligent decision making and task delegation. In this regard, we further elaborate two more metrics, which come under the knowledge TM as non-social TMs and social TMs. In non-social trust, the idea is to find whether the trustor can rely on physical or cyber objects and social trust determines whether a trustor can depend on other social objects [14]. Let us consider a specific trustor *A* and a trustee *B* with respect to a particular goal *g* in the decision making process. Based on this setup the definitions of the attributes in Figure 3-1 are [123];

(a) Social Attributes

- Relationship: Mutual association on completing goal g
- Willingness Trust: *B* shows no resistance over accomplishing the goal *g*
- Persistence Trust: Consistency over time, conquering the task
- Confidence Trust: Confident about B himself towards realizing *g*
- Spatial Trust: Social proximity between A and B on realizing g

(b) Non-social Attributes

- Competence Trust: B is beneficial and capable of realizing *g*
- Disposition Trust: B actually performs the task
- Dependence Trust: Achievement of goal *g* relies upon B
- Fulfillment Trust: B's contribution is necessary to achieve the task
- Temporal Trust: Frequency and Duration of interactions between A and B

Under the social TAs, the relationship TAs in Figure 3-1 defines the mutual relationship between the trustor and a trustee. It is reasonable to assume that if two objects have a noble relationship between them, a higher trustworthiness can be expected between them. As an example, if the trustor and the trustee are operated in close proximity such as looking for a parking lot near a supermarket, then both benefit (e.g. getting a vacant, closest, easily navigable parking lot) from their relationship based on location similarity that we have identified as co-location TA. Likewise, if the two objects are in a working relationship like car sharing in which one needs to provide a service and other needs to get the service, both can support each other via their co-work association.

Furthermore, it is important to maintain knowledge about the consistency of trustworthy service provisioning. We discuss properties related to this issue under persistence. The features like cooperativeness under persistence represents the level of social cooperation from the trustee to the trustor. The higher cooperativeness means the higher trust level in an IoT ecosystem. A user can evaluate the cooperativeness of others based on social ties and select socially cooperative users. Additionally, we have introduced a rewarding system, which also comes under persistence, in order to track the history of misbehavior situations or unsuitable reactions originated by the trustee. A rewarding TA can be used to either encourage or discourage further interactions with a particular trustee based on its past character.

Moreover, in an IoT ecosystem, service provisioning (discover, manage and terminate) is based on its social relationships without solely depending on the underlying system level information. Therefore, it is vital to identify TAs, which determine the social proximity of the objects in collaboration. In this aspect, we identify three properties under spatial TAs in as mutuality, centrality, and CoI as governing features that define the social positioning of a trustee. Mutuality measures the degree of profile similarity between the trustee and trustor in resemblance to what is used in social networking. The community-interest represents whether the trustor and the trustee have a close relationship in terms of social communities, groups, and capabilities. Two objects with a high degree of community-interest have more opportunities for interacting with each other, and this can result in a higher trust level. Centrality measures the importance of a trustee among other objects with respect to a particular task and context.

To capture the significance of time-related information to trustworthiness evaluation, TAs like the frequency and duration of the interactions, which come under temporal TAs, can be used. It is logical to assume that the higher the frequency and duration of interactions, the more trust is built up among the associating objects. On the contrary, the shorter time spent on each other, the less knowledge is gathered on each other's behaviors and capabilities. As an example, in whitewashing attacks, a dishonest object can vanish for some time and rejoin the service in order to clear its reputation. However, if a trustor can keep a record of the consistency of the interested trustees then it can avoid such situations.

3.3.2. **Experience based and Reputation based Trust Metrics**

After acquiring enough evidence about trustees through the knowledge TM, the trustor can initiate collaborations with selected trustees based on the perception that the trustor has already obtained. However, the result of these interactions might differ from the perception and hence it is critical to keep a record of each individual experience to be used in future interactions. For instance, the experience might be feedback from consumers after each transaction (as used in many e-Commerce systems), just a Boolean value (0/1) indicating whether a service transaction successfully operates (as in some reputation-based trust systems), etc. Then, by accumulating these experiences over time in relation to the corresponding contexts, tasks and times, the trustor can build up additional intelligence compared to the knowledge TM.

To further enhance the perception of the trustor, other objects can share their experience in using the trustee, upon a request by the trustor, which we identify as reputation or the global opinion of the trustee. As an example, we have come up with a non-bias PageRank based model to calculate the reputation values of trustees in a distributed network as in [124]. In summary, the experience TM is a personal observation considering only interactions from a trustor to a trustee, whereas the reputation TM reflects the global opinion of the trustee.

According to the above definitions, the formation of trust according to the three TMs is shown in Figure 3-2 for the objects who have just entered the IoT ecosystem. The first step of the trust assessment is to build to up the knowledge about the trustee who is about to receive/provide the service from/to the trustor. At this level, no information

can be generated in relation to experience or reputation as there have been no previous interactions among them.

However, after just one transaction, both parties can start the process of obtaining experiences through their conversations. Generating trust scores based on experience is an iterative process and the more time spent with each other (Trustor and Trustee), the more they can learn about their own ability to provide accurate judgments. Likewise, a trustee might initiate or receive more interactions from other peers in the service domain. At this point onwards, peers can generate trust scores based on the reputation values for the trustee. Finally, accumulation of knowledge, experience, and reputation over time and context can be presented as the trust value of their relationship at this moment for applications like trust based decision-making or any other appropriate service as required.

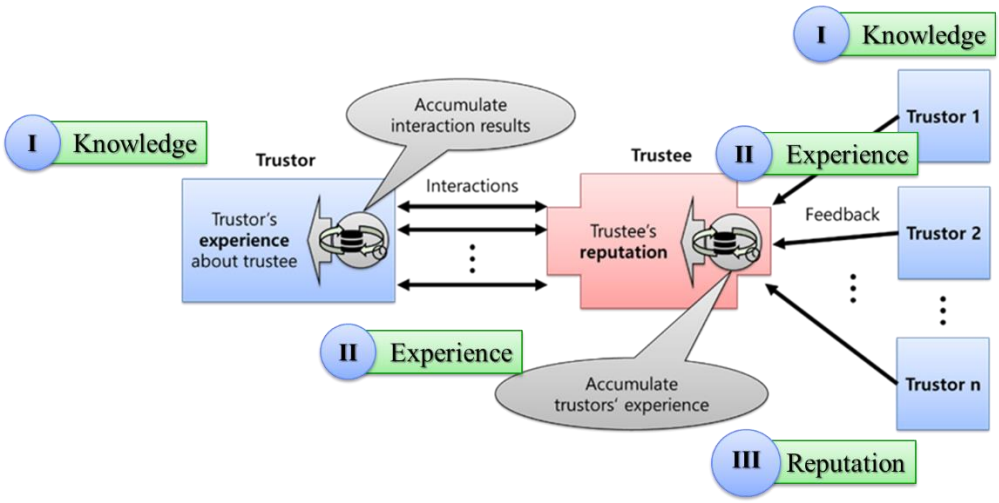


Figure 3-2: Direct and Indirect Trust Evaluation.

3.3.3. **Trust Evaluation Process**

In the context of computer science, trust evaluation can be defined as the process of obtaining a quantitative value for trust (see Section 3.2) between at least two parties; a trustor and a trustee. In general, obtaining a final trust value is a dynamic process and relies upon many properties of the trustor, the trustee, the environment, and the aggregation methods. According to our proposed trust model in Section 3.3, we find hierarchical relationships among these properties as shown in Figure 3-3.

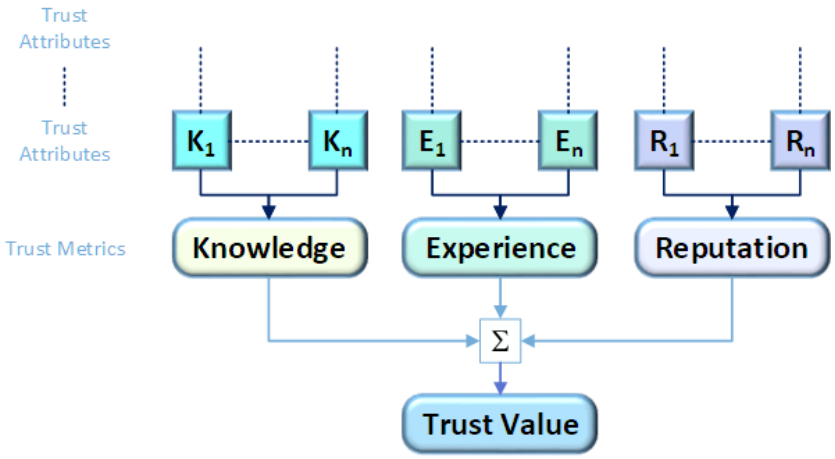


Figure 3-3: Aggregation of TAs and TMs towards Trust Value.

According to the model, the first step of the trust evaluation is to estimate relevant TMs depending on the application. However, this information is not readily available and hence attributes, which define these TMs, must be obtained. There are numerous methods available to estimate these TAs ranging from numerical methods, probabilistic methods, belief theory to ML methods. In simple terms, the mathematical approach to find the trust value trustor *i* and trustee *j* can be represented as below.

$$K_{ij} = \alpha_1 K_1 + \alpha_2 K_2 + \cdots + \alpha_n K_n \tag{3.1}$$

$$E_{ij} = \beta_1 E_1 + \beta_2 E_2 + \cdots + \beta_n E_n \tag{3.2}$$

$$R_{ij} = \gamma_1 R_1 + \gamma_2 R_2 + \cdots + \gamma_n R_n \tag{3.3}$$

$$Trust_{ij} = \theta_1 K_{ij} + \theta_2 E_{ij} + \theta_n R_{ij} \tag{3.4}$$

where α,β,γ, and θ, are weighting factors that normalize each metric in between 0 and 1. K_x, E_x and R_x represent the TMs-Knowledge, Experience, and Reputation, respectively.

Note that equations (3.1), (3.2), and (3.3) denote an iterative process. To determine one TM, it might be necessary to examine several levels deep in the hierarchy, until a sufficient number of attributes are assessed to represent the relevant TM. Therefore, the dimensions of matrix TA_x can vary from one to infinity depending on the use case. However, trust value does not need to be 100% correct, as we discussed in Section 3.2 and hence the optimum number of dimensions must be considered based on the criticality of the requirement.

However, equations (3.1) to (3.4) show the preliminary idea of trust evaluation and based on this, there are a number of trust evaluation methods found in the literature as discussed in Chapter 3, each with their positives and negatives. It is possible to characterize such methods into several categories as network architecture-based methods, policy based methods, reputation based methods, knowledge based methods, trust aggregation based methods, and those based on novel concepts like ML.

3.4. Trust Management Platform

Definition 2: *Trust Management Platform*

It responds to various requests from many service objects, analyses the level of trust by tracing the accumulated data from various sources and make suitable decisions in order to establish reliable communication among objects.

To achieve such goal, the required number of processes inside the platform can be vastly different from one application to other. However, it can be identified five must-have processes in any generic trust management platform as Trust Data Collection, Trust Data Management, Trust Information Analysis, Dissemination of trust Information and Trust Information Lifecycle Management [125].

- Trust Data Collection

  The trust data collection process mainly involves two sub-processes planning to collect the trust data as well as collecting the trust data. In planning trust data collection, it is necessary to elaborate what and how much trust data should be collected. Entries of trust data should be related to the purpose of trust evaluation and be uncorrelated with each other for usability on trust evaluation processes. Trust data should also be collected as infrequently as possible due to the associated time and resource costs. Moreover, excessive trust data collection may cause privacy concerns within ICT infrastructures and services. The amount of trust data, which needs to be collected, shall also be considered in this process. Further, it is important to monitor and verify the process of data collection with respect to the expected requirements.

- Trust Data Management

  To generate the information about trust, collected trust data must be processed in such a way that it gives a meaningful result. In this regard, the main concerns

in data management would be to consider the purpose of trust evaluation and the assurance that data collected will not cause any degradation of accuracy or waste resources (due to incorrect or polluted data) whilst creating trust information. Furthermore, this process is responsible for protecting collected data from abnormal access such as hacking or data leakage.

- Trust Information Analysis

  The trust information analysis process extracts meaningful trust information from trust data and other trust information for objects in ICT infrastructures and services. Because trust is generated on the relationship between the trustor and the trustee, trust information should explicitly reflect the trust relationship using the objective and subjective manner.

- Dissemination of Trust Information

  Trust dissemination is the mechanism used to distribute or broadcast trust information, which is created in the previous process. There are many ways of disseminating trust information in different domains. In the case of a social domain, recommendation and visualization methods are considered as the main approaches to disseminate trust information. Efficient, effective and suitable trust dissemination methods should be developed, that is, only trust information that concerns a trustor should be disseminated to trustors in ICT infrastructures and services. A trustor can determine the trustworthiness of a trustee with trust information with the subjective criteria.

- Trust Information Lifecycle Management

  In previous processes, trust information is created and disseminated to ICT infrastructure and services. Because of the dynamic characteristic of trust, trust information should be re-established, updated, and abolished. The re-establishment phase replaces components of trust information due to the change of an object or a service. The feedback of the trustor that receives trust information on the trustee also could be related to the replacement of components in the reestablishment phase. At the update phase, the value of trust information and trust data is updated, and trust information is re-evaluated. Finally, if trust information of an object or a service is dispensed with, trust information can be abolished or reset.

According to the discussion above, the proposed trust management platform, which manages all aspect of trust data collection to trust lifecycle management in an IoT ecosystem is shown in Figure 3-4. The platform consists of several modules like TAG, Trust Broker (TB), Trust Data Access Object (DAO), DR, Application Programming Interface (API), TM Extractor (TME), AI Engine (AIE), Trust Information Analysis (TIA), Trust Modelling Algorithm (TMA), Trust Service Enabler (TSE), and Trust Lifecycle Management module (TLM). These modules will perform one or several tasks at a time to achieve the objectives discussed above.

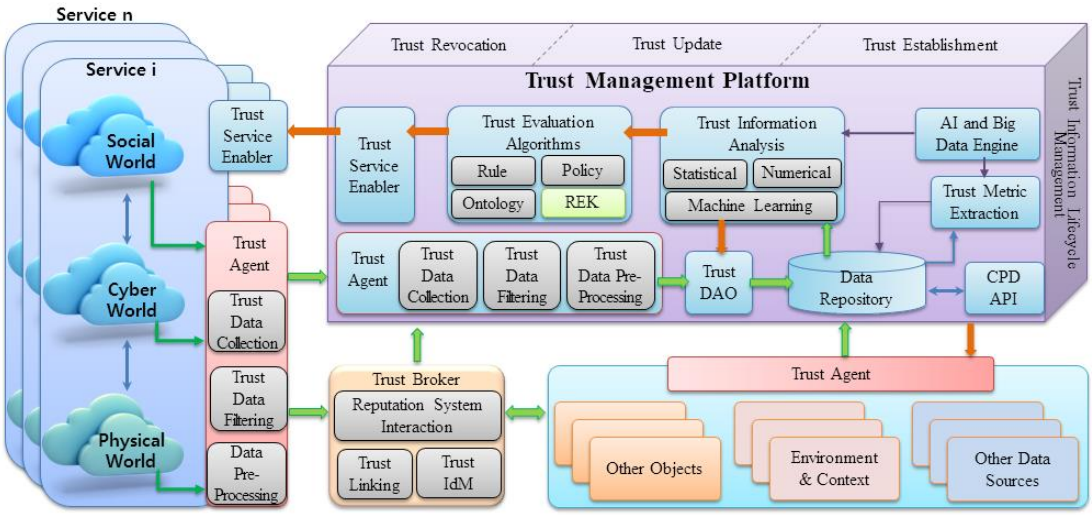


Figure 3-4: Trust Management Platform.

– Trust Agent (TAG)

  The first step of the trust evaluation process is to collect appropriate data from all the sources, where applicable, including cyber, physical and social objects of relevant services or applications. Generally, TAG works similarly to client-server application, in which objects and the central platform change their role depending on the direction of data flow. The data could be either information obtained directly from relevant parties, experience or opinions of objects as reputation or feedbacks from/to other objects, applications or services. Furthermore, depending on the available processing power, data filtering and data pre-processing will be carried out at the object itself, at the broker or at the centralized platform.

– Trust Broker (TB)

  The information received from TAGs, and from other various types of applications, and services in the IoT ecosystem will be managed by the TB.

Essentially, we propose to implement the broker as a publish-subscribe system such that any service or application, which wants data for trust evaluation, must subscribe to the system using proper credentials to receive information from the broker, whenever this information is available. In this particular scenario, the trust management platform uses the TB to get direct as well as indirect information from various sources.

- Data Repository (DR)

  Once the TAG, inside the platform acquired the data, it will be stored in the local repository, close to the platform to be used by other modules in the platform. Additionally, it stores information from the TME, TIA and through an API that connects to external trust data sources.

- Trust Information Analysis (TIA)

  All the information acquired so far is analyzed inside this module with the support from an external AI engine. Depending on the availability of data, the attributes, which were identified by the TME, and the requirement of the analysis, the TIA module will utilize numerical, statistical, ML or ensemble approach to calculate the metrics. To support ML and ensemble base calculation, an external AIE is implemented to provide more robust and autonomic execution.

- Trust Modelling Algorithm (TMA)

  The final trust value based on the calculated metrics from previous steps is generated inside based on the appropriate model for each application. For example, if the REK model is used, a trust value based on the knowledge, reputation and experience are generated. The TMA should essentially work together with the TIA to find the best possible metrics for each model in which some models will combine the metrics according to pre-defined rules or policies, while others will generate these rules and policies dynamically to suit the situation in the best possible way.

- Trust Service Enabler (TSE)

  This is one of the most important modules in the platform, which collects trust information from the platform and other knowledge bases to take appropriate decisions based on trust. The service enabler is different from the service

provider in that the service enabler is concerned with the extraction of information and knowledge from the gathered lifecycle data and packaging or integrating them into suitable service content, whilst the service provider is responsible for providing services to customers and stakeholders by using the service enabler.

The roles of the TSE can be identified as:

a) collecting trust information and making decisions

b) maintaining a persistent trust data/information database

c) attention to access control during the service deployment

d) presenting the results in an appropriate format to the customer (e.g. trust value, report, decision/action, consequences, etc.) [126].

3.5. Discussion

Firstly, this chapter formally defines the trust concept, its key characteristics, and features for a clear understanding of trust in the IoT. As discussed in Table 2-5, there are many definitions of trust resulting in difficulty in establishing a standard notation of trust in computer science. Such confusing definitions and blurred nature of trust itself make developers or researches to come up with stable and standardized trust evaluating mechanism similiter to well-known security mechanisms. Therefore, the concept of computational trust and definitions stated in this thesis are standardized with ITU-T and confidently it would help to avoid such ambiguities in the future.

Then, a novel trust evaluation model based on three metrics called knowledge, experience and reputation are then proposed to evaluate trust. As this thesis pointed out in Section 2.2, there are a number of existing works can be found in the literature. However, most of these approaches consider only one-dimensional model and often ignore the characteristics of trust including its subjective nature. Because of that, they failed to evaluate trust in any given situation making it unreliable to consider for decision making process. On the other hand, policy based trust evaluation models discussed in Section 2.2.1. is totally dependent on the policies defined and any interaction outside these policies will lead to categorize the trustee as untrustworthy. Similarly, reputation-based trust models show weak performance again fake trust

information and highly sensitive for perceptions of the reputation generators and knowledge based trust evaluation models are suffered from humongous amount of information and inability to estimate a correct number of attributes required for trust evaluation. However, the proposed trust evaluation model is based on three dimensions and hence it shows good resilient against hard-line decisions as it does not need to rely on one dimension like in policy based models, robust response for fake reputations as it can validate these responses through other dimensions like experience, and reduce complexity when evaluation TAs based on knowledge as the trust evaluation is supported by three dimensions which indeed helps to compromise some of the less relevant TAs.

On the other hand, a noticeable drawback of existing trust management platforms is the absence of a functional description of the modules that required to evaluate trust. Even though most of the current work uses the term "Trust Management", which is actually used it to emphasizes the trust evaluation techniques and there is only very limited work that describes the actual functions needed to evaluate trust from data collection trust information lifecycle management. Therefore, a trust management platform for trust evaluation and decision making for a trustworthy IoT Eco-system is proposed as a solution to such limitations, emphasizing key functionalities, requirements and standard interfaces for autonomic decision making based on trust.

3.6. Chapter summary

In this chapter, we discussed the importance of trust in achieving more futuristic, and trustworthy services in an IoT ecosystem. First, we identified the key challenges and necessity of trust in a hostile environment, where billions of objects from the social, cyber and physical worlds, with various ambitions, are interconnected. Then a formalized definition for computational trust is presented to avoid any misinterpretation of trust in future in the similar area of research. Then a trust evaluation model is presented to describe the process of TA identification, evaluation, and trust relationship entablement. Finally, a trust management platform is presented to address the issues from data collection to trust information lifecycle management in the trust evaluation process.

CHAPTER 4: REPUTATION BASED TRUST EVALUATION MODEL

4.1. Introduction

SIoT is a revolutionary idea, which combines traditional IoT models with social network paradigms. “Objects” in SIoT formulate social relationships with other “objects” according to the relationships, defined by their trustworthiness. Hence, in this section, thesis propose a reputation based TM assessment algorithm called RpR (Recommendations plus Reputations) that enables objects in SIoT to build associations in a trustworthy manner. Along with SIoT concepts, recommendations can be defined as the opinions of friends in the context of human social networks and reputations as the opinions of other global objects.

The proposed algorithm here is an extended version of the popular PageRank™ (PR) algorithm, which ranks web pages according to their importance [127]. The PR algorithm discusses how the incoming and outgoing links of a web page can be used to evaluate the importance of that particular page compared to its neighbor s. However, the PR algorithm is not capable of assessing objects other than those directly connected, and it shows extremely weak performance in the case of fake reputations. With these issues in mind, we propose an algorithm, which evaluates recommendations and reputations and generates a collective trust value, which will be identified as RpR trust scores.

The rest of the section is organized as follows. Section 4.2 describes the concept of trust corresponding to a SIoT environment. Section **Error! Reference source not f ound.** presents the formulation of RpR algorithm while meeting the properties of a real-world scenario. Section 4.4 provides simulation results in order to validate and compare the desirable attributes and the performance enhancements.

4.2. Background

In SIoT, objects are linked with services that they can deliver and acquire. The key objectives of such a network are to discover reputed services, active resources and publish this information over the network to be used by interested parties. To achieve this kind of behavior, navigating through a social network is done based on the

relationship of objects rather than depending on typical internet discovery tools. Social relationships can be considered as human-human, human-objects, and objects-objects. Relationship based routing is a far more proficient technique for SIoT compared to standard routing methods, due to its requirements such as context awareness service delivery, trustworthiness, and scalability.

Accordingly, we categorize relationships into four main categories depending on their trust level: namely Friendship, Ownership, Interaction, and Community interest. In Figure 4-1, we have shown an example scenario of this classification using a car sharing use case. As demonstrated in the example, each object has at least one owner and which may be a friend, a part of a transaction and/or a member of a specific community. Also, we identify the trustor and trustee relationship in which the trustor is responsible for evaluating the trust and trustee is responsible for providing the necessary information requested by the trustor.

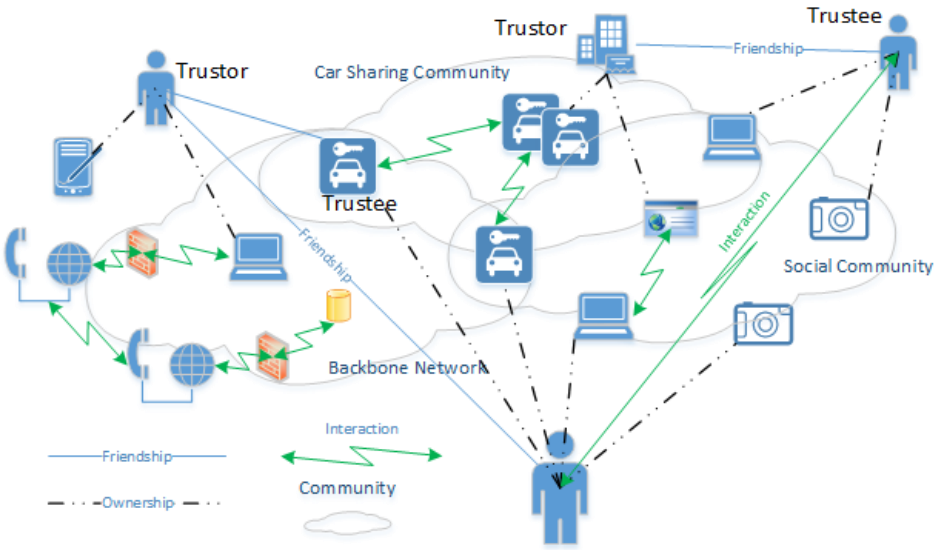


Figure 4-1: Associations in a SIoT system based on a car sharing use case.

Being a friend is one of the strongest relationships in SIoT and it allows devices connected to friends to collaborate with each other more reliably compared to others. Likewise, there can be devices, which are operated to achieve certain goals or policies set by the community. Some examples are social networking, backbone services, security measures, and the communities who interfere with the goodness of the operation like cybercrime, spam, and any other rule violators. Furthermore, in a setup like SIoT, objects belonging to the same community are more likely to share information with their members as opposed to those who are not in the community. As

an example, a spam community might share information or mechanisms relevant to self-promoting, bad-mouthing, good mouthing within their community and work collaboratively to alter the normal behavior of the target system. Our use case example presented in Figure 4-1 is a car sharing system where people can rent a car for a period of time. Normally, a customer wants a reliable car for a reasonable service level and cost. At the same time, a car provider and a broker need to ensure that the customer is trustworthy. To meet these criteria, it is essential to have a system that provides assurance for every party who participates in the transaction, which is essentially establishing a view of trust among objects in SIoT.

4.3. Evaluation of Reputation Based Trust

The algorithm is specifically suitable for a distributed environment where every object keeps a record of its own trust value based on a particular set of friendly and third-party estimations. We assume that if a particular object (Trustor) is already connected or in contact with another object (Trustee), there is a relationship with these two objects regardless of trustworthiness. We apply this property to generate a weighted directed graph where the vertices represent objects and edges represent the relationships between them. Figure 4-2 shows an example Car Sharing use case, where User 1 (“A”) has a friendly relationship with Service Provider 2 (“D”) and User 2 (“E”) provides some reputation on object “A” through object “B” based on their social relationship.

Moreover, it is reasonable to believe that an object would have a higher trust score if many objects were directed towards it. However, if a particular object provides an excessively high number of opinions about its neighbor, one may suspect that this is a dishonest object trying to achieve some undesirable objective. Therefore, filtering out opinions is also critical to have a more trustworthy score. Keeping these factors in mind, we develop Recommendation and Reputation assessment algorithms and they will be combined later to formulate the final algorithm to calculate RpR trust scores.

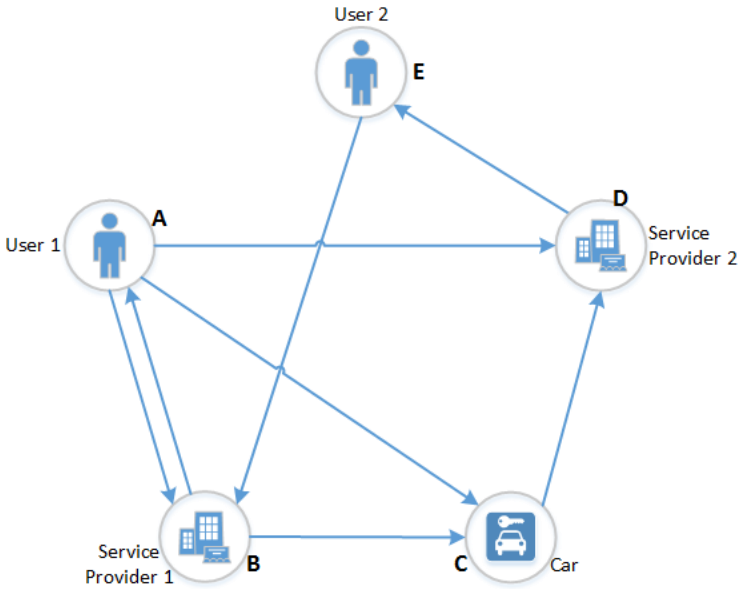


Figure 4-2: A graph representation of the SIoT Model.

4.3.1. **Recommendation Assessment**

In this section, we formulate an algorithm to assess recommendations for SIoT in comparison to PR for the web. Since our goal is to build the algorithm based on relationships but not on web links, multi-graphs and self-promoting links are ignored. Let us consider an object v_i which has a friendly relationship with object v_j as shown in Figure 4-3. Since the number of incoming relationships corresponds to the recommendation level of a target object, we can express the recommendation value of v_i as $R_{\text{rec}}(v_i) = \sum_{j=1}^{N} 1$, for N number of total directly connected objects of relevance to a particular context. Note that the value of each relationship between i and j is assumed as one at the initial stage. However, if the recommender has many outgoing links, it is an indication that it is a friend of many other objects and hence a recommendation score for each target object must be equally distributed along the links as shown in equation (4.1). In order to simplify the computational overhead and the algorithm, the factor $1/O(v_j)$ is represented in matrix form as in equation (4.2)[127]. Here the function O(-) represents the number of outgoing links.

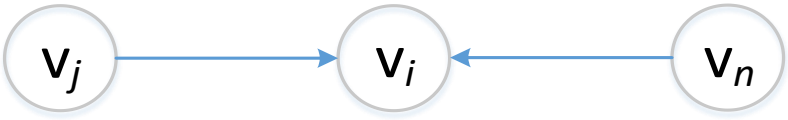


Figure 4-3: An example of recommendation flow.

$$R_{\text{rec}}(v_i) = \sum_{j=1}^{N} \frac{1}{O(v_j)} \tag{4.1}$$

where $O(v_j)$ is equivalent to a number of direct links from v_j.

$$T_{ij} = \begin{cases} 0 & Otherwise \\ \dfrac{1}{O(v_j)} & if(v_j, v_i) \in \epsilon \end{cases} \tag{4.2}$$

where T_{ij} represents the transition probability from object "*j*" towards "*i*" in the directed graph.

$$R_{\text{rec}}(v_i) = \sum_{j=1}^{N} T_{ij} \tag{4.3}$$

As the objects are distributed, the final recommendation score of each object can be calculated recursively as in (4.4).

$$R_{\text{rec}}(v_i) = \sum_{j=1}^{N} T_{ij}\, R_{\text{rec}}(v_j) \tag{4.4}$$

The equivalent matrix form is:

$$\boldsymbol{R}_{\text{rec}}^{t+1} = \boldsymbol{T}\, \boldsymbol{R}_{\text{rec}}^{\boldsymbol{t}} \tag{4.5}$$

where $\boldsymbol{R}_{\text{rec}}^{t+1}$ is the predicted recommendation score, T is the transition matrix, $\boldsymbol{R}_{\text{rec}}^{\boldsymbol{t}}$ is the current score and ***t+1*** denotes the current time stamp.

Equation (4.5) represents the numerical model for the SIoT environment comparable to PR for web links. Moreover, this is developed based on the assumption that there are no dishonest objects present and the initial recommendation values are uniformly distributed over the entire network. In a real environment, this is not the case and we discuss a solution for these issues in this section. First, it is essential to filter out untrustworthy objects from good objects as we assume that good objects often recommend reliable objects and dishonest objects recommend unreliable objects. This is in relation to human behavior and this thesis adopt that concept here to develop our model discussed in equation (4.5) furthermore as described below.

Let us consider object A in Figure 4-2. It can be observed that object "A" provides more outgoing links to other objects, i.e. "A" has a good relationship with many other

objects. If a third party user is connected to object A, that user can reach three other friends of "A" easily in order to get some information or services. Following this approach, if we can distinguish objects like "A", which is well connected with other objects, we can reasonably filter out dishonest objects from the environment. To identify such objects we adopt the inverse version of the PR algorithm discussed in [128]. In the PR model, a rank score depends on how many inwards links there are from adjacent objects. The higher the number of inward links, the greater the rank score of the target object will be.

Accordingly, an inverse graph of Figure 4-2 is shown in Figure 4-4. Now that the graph is inverted, PR gives the ranking scores based on most outgoing links in contrast to the original PR algorithm, which is based on inward links. We define inverse transition matrix $\boldsymbol{U}$ as in equation (4.6) where $I(v_j)$ are the input relationships.

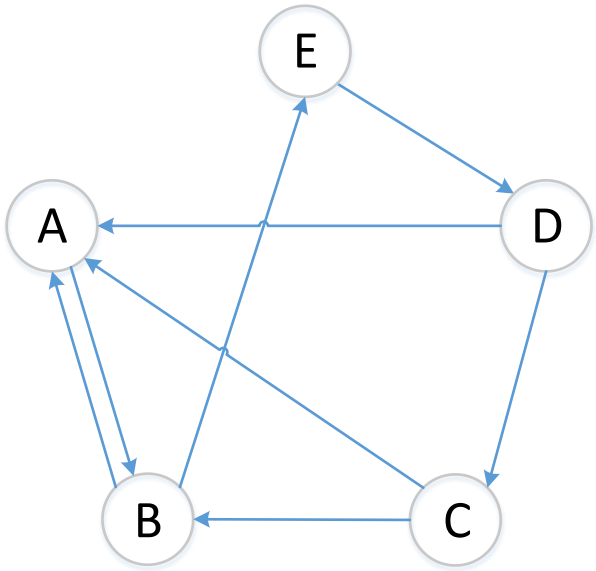


Figure 4-4: An Inverse Graph of the SIoT Model.

$$U_{ij} = \begin{cases} 0 & Otherwise \\ \dfrac{1}{I(v_j)} & if(v_i, v_j) \in \epsilon \end{cases} \tag{4.6}$$

Note that $\mathbf{T}^{\mathrm{T}} \neq \mathbf{U}$. Then the inverse rank scores of each object can be calculated by substituting (4.6) in to (4.5) as shown in (4.7).

$$\boldsymbol{t_{r+1}} = \mathbf{U}\, \boldsymbol{t_r} \tag{4.7}$$

Equation (4.7) provides an idea about which object has the highest number of outgoing links. As this equation is obtained from the inverse matrix, it also implies the highest

number of incoming links in the original matrix. According to our assumption, more incoming links means the object is well worth visiting and hence the most trustworthy. However, it is required to define a threshold value (δ) to select the most trustworthy objects "k" from total "N" objects. After identifying the most trustworthy objects, we can distribute initial rank values among these specific objects, making others zero.

Let us say that the modified vector is $\boldsymbol{t}_r$ where k number of objects have a positive value and N-k number of objects are zero. Then, we combine the trust vector $\boldsymbol{t}_r$ with (4.5) as in (4.8).

$$\boldsymbol{R}_{\text{rec}}^{t+1} = \alpha \boldsymbol{T} \boldsymbol{R}_{\text{rec}}^{\boldsymbol{t}} + \beta \mathbf{t}_r \tag{4.9}$$

Now, the model is biased and hence the trustor can get more updates from most trustworthy objects instead of uniformly as before. Here, α and β are decay parameters which bring the final scores in between zero and one.

4.3.2. **Reputation Assessment**

Reputations are the opinions from objects other than friends in our proposed SIoT model. One such illustration is shown in Figure 4-5, where object v_k gives its opinion of v_i through object v_j. More clearly, this can be a situation where User 1 in Figure 4-2 has a relation with D through C when the direct link from A-D is not present.

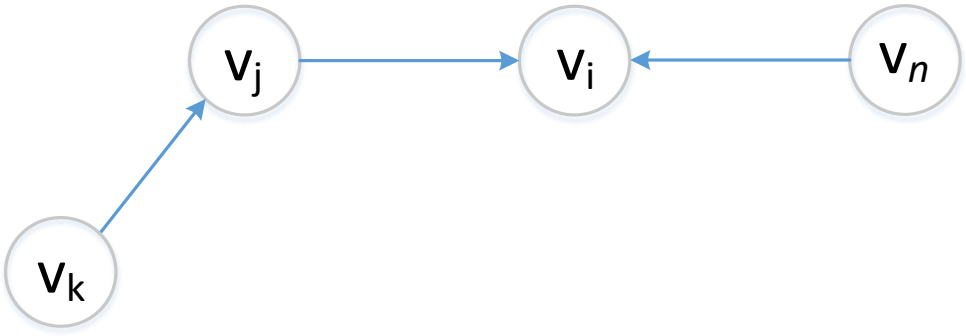


Figure 4-5: An example of Reputation Flow.

In graph theory, the power of the transition matrix determines the paths from non-directly connected objects towards a particular object, having an edge count equal to the power of the matrix. Based on this property, we can find a number of indirectly connected relationships, which contribute to the final reputation score. However, as the networks grow, there can be a large number of such paths, which are not closely associated with the targeted object to provide valuable opinions in order to calculate a

final reputation value. Hence, the number of hops that one object can pass the reputation score towards another object is restricted to a pre-defined value. Furthermore, we observe through simulations that after three levels deep in the graph, the effect of the reputation score is negligible. Based on these grounds, reputation value from v_k towards v_i can be modeled as in (4.10).

$$R_{\text{rep}}^{k \to i}(v_i) = \sum_{j=1}^{N} T_{kj} T_{ji} \,.\, R_{\text{rep}}(v_k) \tag{4.10}$$

where T_{kj} is the transition matrix from v_k towards v_j and T_{ji} is the transition matrix from v_j towards v_i. However, according to graph theory, the product of the above two transition matrixes gives the equivalent of 2-length transition matrix as in (4.11).

$$\sum_{j=1}^{N} T_{kj} T_{ji} = \boldsymbol{T}^2 \tag{4.11}$$

Substituting (4.11) in to (4.10), matrix representation of tier-2 reputation scores can be calculated as in (4.12).

$$\boldsymbol{R}_{rep}^{k \to i}(v_i) = \boldsymbol{T}^{\boldsymbol{2}} \boldsymbol{R}_{rep}(v_k) \tag{4.13}$$

Similarly for an n-length graph,

$$\boldsymbol{R}_{\text{rep}}^{t+1} = \boldsymbol{T}^n \, \boldsymbol{R}_{\text{rep}}^{t} \qquad n>1 \tag{4.14}$$

where n is the depth level and limited to three consecutive objects as in Figure 4-6, in order to reduce the computational overhead as well as due to their negligible effect.

Figure 4-6. Depth level that Reputation scores are collected.

4.3.3. **Aggregated Assessment**

Combining the two scenarios (Recommendation and Reputation), discussed in (4.9) and (4.14) the final algorithm, which aggregates recommendation and reputation scores is shown in (4.15).

$$\boldsymbol{R}_{\mathrm{RpR}} = \lambda \boldsymbol{R}_{\mathrm{rec}} + \mu \boldsymbol{R}_{\mathrm{rep}} \tag{4.15}$$

where $\boldsymbol{R}_{rec}$ represents the recommendation scores from friends and $\boldsymbol{R}_{rep}$ denotes the reputation scores which had interactions with the target object in the past. Again, λ and μ are normalization factor which satisfies the condition $\lambda+\mu=1$, in order to maintain the final RpR score of each object in between 0 and 1.

```
1.  function RpR*
2.  input
3.          N               number of objects
4.          T               transition matrix
5.          U               inverse transition matrix
6.          δ               threshold value for good recommendations
7.          α               decay factor of recommendations
8.          β               decay factor of trustworthy roots
9.          γ               decay factor of  reputations
10.         m               number of iterations
11. output
12.         t_r             trustworthy roots
13.         R_rec           recommendation scores of each object
14.         R_rep           reputation scores of each object
15.         R_RpR           RpR trust scores
16. begin
17. //discover trustworthy objects
18.     t_r(···)
19. //evaluate recommendation scores
20.     R_rec^t = t_r
21.     for i=1 to m do
22.         R_rec^(t+1) = α T R_rec^t + β t_r
23.     return R_rec
24. //evaluate reputation scores
25.     R_rep^t = 1/N
26.     for i =1 to m do
27.         R_rep^(t+1) = T^n R_rep^t          1 < n < 4
28.     return R_rep
29. //joint RpR scores
30.     return R_RpR = λR_rec + μ R_rep
31. end
```

Figure 4-7: The Algorithm that Calculates RpR Trust Scores.

The function RpR, shown in Figure 4-7, computes the RpR trust scores for the model obtained for a SIoT environment in (4.15). The first step of the algorithm is to find the most trustworthy objects by calling the function $\boldsymbol{t_r}$ ***(···)***. This will create a vector in which initial scores are positive only for most trustworthy objects where scores are greater than the threshold value requested by the context. In the second step, Recommendation scores are calculated in a recursive manner as in (4.9) where initial conditions are set as described by trust vector $\boldsymbol{t_r}$. Similarly, Reputation scores are calculated where opinions are collected from objects up to the third tier, i.e. $1<n<4$.

Finally, the two scores are combined after normalizing with decay factors α, β, λ, and μ.

4.4. Experiments and Results

In order to evaluate the model, we have conducted a simulation using MatLab and based on the illustration in Figure 4-2. The experiments are carried out on a PC which consists of 8 CPU cores (Intel Core i7-2600, 3.4GHz) and 8GB RAM. At the beginning of the experiment, a network of five objects created randomly using MatLab. Then the algorithms described in Figure 4-7 and Figure 4-10 are executed to find the RpR, PR and ID scores respectively. Once these scored are stored separately, the network is expended by adding five more objects randomly at a time until it reaches a hundred objects. At each step RpR, PR and ID are calculated and stored for later analysis.

However, we observed that the algorithm described in Figure 4-7, is capable of handling tens of thousands of nodes due to the simplicity of the calculation model just like in the PR method. The complexity of the model is constrained by the *n*-value, which determines how far the algorithm goes to collect the reputation values, and by limiting the number of recursive iterations to preserve an accuracy up to 10^{-5}.

With these adjustments, the algorithm converges quickly with only about six iterations as shown in Figure 4-8 compared to the PR algorithm. The horizontal axis of Figure 4-8 represents the number of iterations and the vertical axis denotes the error between the predicted score and the current score.

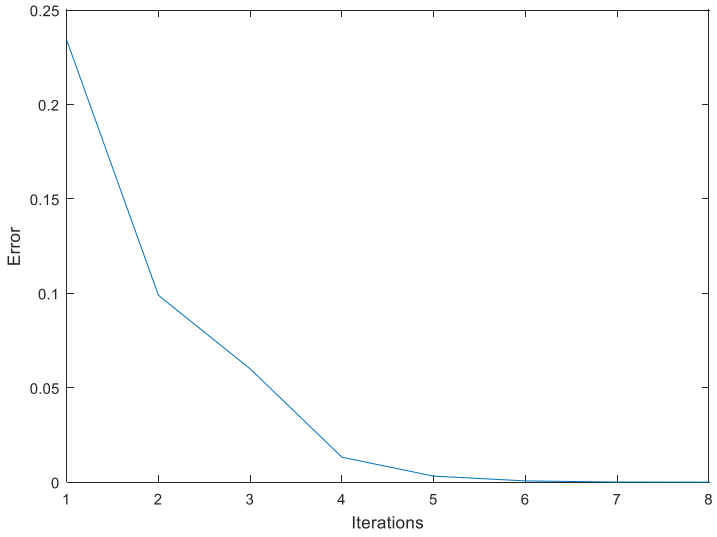


Figure 4-8: Convergence Rate of the Algorithm.

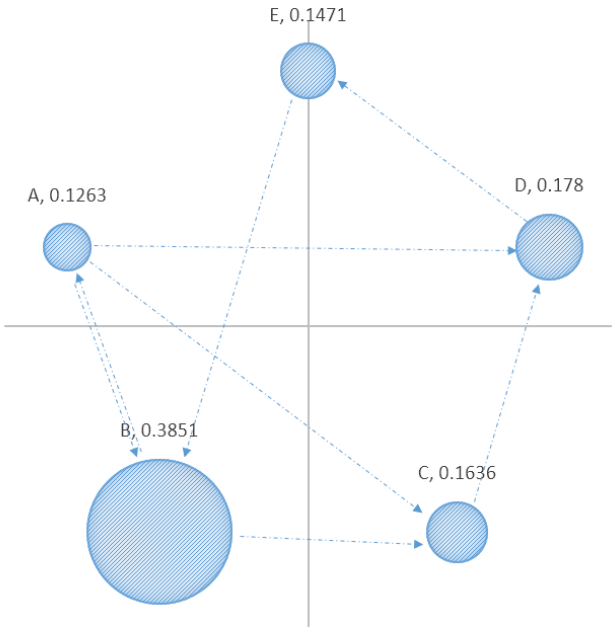


Figure 4-9: Distribution of RpR Scores.

The RpR ranking scores obtained for the five object model is shown in Figure 4-9 to clarify the correctness of the algorithm. The area of the circles is proportional to the trustworthiness score of a particular object. It is obvious that "B" has gained a higher score compared to others as it does have many incoming and outgoing relationships with friendly neighbor objects. On the other hand, object "A" received the minimum score, as it only got one recommendation from "B" while it tried to reach three others. "C" and "D" have obtained a nearly equal rank, as both have two incoming recommendations and one outgoing link.

```
1.  function PR*
2.  input
3.    N   number of objects
4.    d   probability of random surfer
5.  Output
6.    PR(N) PageRank matrix
7.  Begin
8.  //Let PR be an array of N elements
9.  for i=1 to N-1 do
10.   PR[i]=1/N
11. end
12. Repeat
13.   J=PR
14.   for i=1 to N-1 do
15. // Let Oj be the number of outgoing
       edges
16.     PR[i]=(1-d)+ d.ΣPR[j}/Oj
17.   end
18.   If PR-J<error
19.     Return PR
20. End
```

```
1.  function ID*
2.  input
3.    N   number of objects
4.  Output
5.    ID(N) Indegree matrix
6.  Begin
7.    ID[i]= Number of inward edges
8.    Return ID
9.  End
```

Figure 4-10. Pseudocode algorithms of PR and indegree algorithms.

Another view of the same example is shown in Figure 4-11, where RpR scores in each individual object are compared with the PR model and with In Degree (ID). Pseudocode algorithms of PR and indegree models are shown in Figure 4-10. It shows that the RpR model is more sensitive to detect trustworthy or untrustworthy objects compared to others. As an example, when the trend is increasing, RpR assigns a reasonably high score for the most trustworthy objects. Similarly, when the trend is decreasing, RpR assigns a lower value compared to others, which makes the model more sensitive to dishonest behaviors as shown in Figure 4-12. It can be observed that RpR is always a good candidate for the detection of suspicious behavior in comparison to the PR model. In this experiment, objects, which have the lowest ID, have been taken as the dishonest objects. That is, after estimating their ID values, they are sorted according to their values in a descending order. Then, the first 20% of the objects are taken as trustworthy objects and the remaining 80% are considered as untrustworthy.

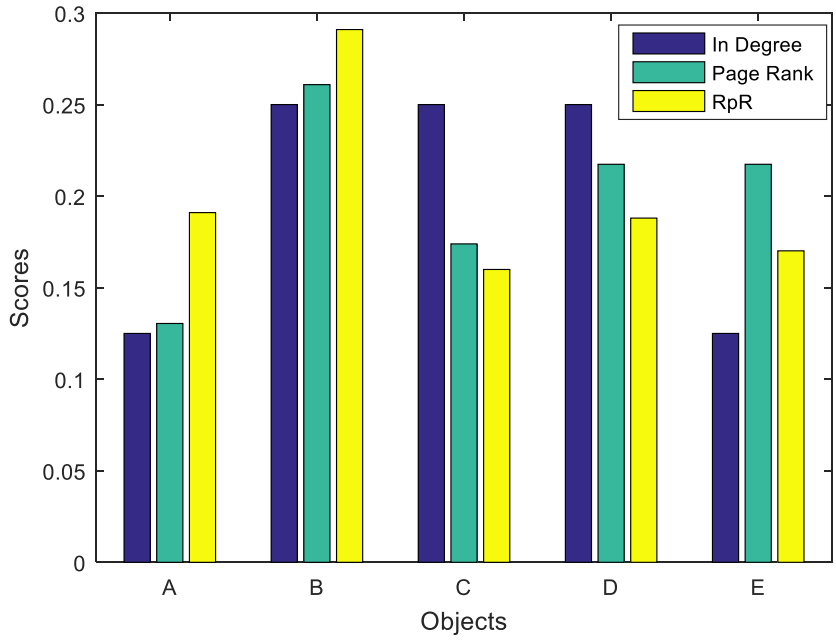


Figure 4-11: Comparison of Distribution of Scores with five Objects.

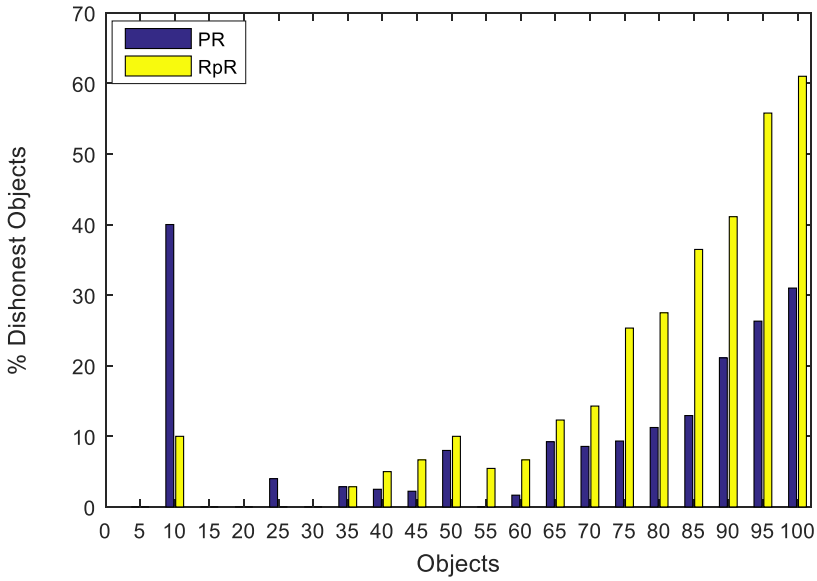


Figure 4-12: Dishonest Object Detection.

One important property of a good ranking system is that it must be unbiased in extreme conditions like when the network size is small or big. We implement this scenario in our simulation and it can be observed that neither PR nor ID algorithms were able to fulfill this with the growth in the number of objects. In this, we checked whether the algorithm is capable of detecting objects, which are in the best 20% of trust scores as shown in Figure 4-13. RpR always showed a consistent performance irrespective of the network size while PR performance was heavily degraded with the increase in the object count as PR/ID models are heavily dependent on opinions from old objects.

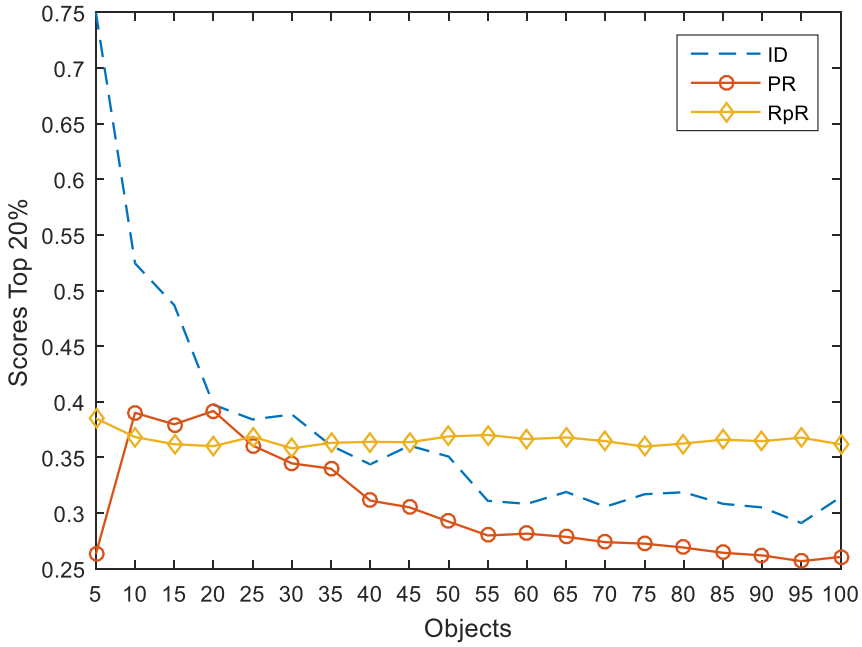


Figure 4-13: Detection of top 20% of trustworthy Objects.

To observe this effect more clearly, we have calculated the Kendall correlation coefficient of both PR and RpR methods compared to ID, as it is the basis for both methods. According to Figure 4-14, the PR method always shows a higher correlation compared to RpR as the effect of inward relationships plays a more dominant role in the PR method to rank objects. That means the score generated by both PR and ID algorithms are heavily dependent on the number of incoming links without considering the credibility of the recommender. On the other hand, RpR is not dependent only on inward links but also recommendations from trustworthy objects, reputations from unfriendly objects and the ability to detect the least trustworthy objects.

In the model described in Figure 4-2, we assumed that there are no hanging objects, which have only inward relationships, but no outgoing edges, in order to avoid accumulation of all the trust scores towards one object during the process of iteration. However, these types of objects should have good scores as several friends recommend them. Therefore, to be fair with hanging objects, we suggest replacing the column of the dangling object with equally distributed values. In this way, the importance of the object would be equally redistributed among the other objects at the beginning, instead of being lost and at the end of the iteration, the true value will be transported back to the object. Furthermore, the objects, which do not have any incoming links will be ignored from the index, as no object would prefer to get any service from these type of objects

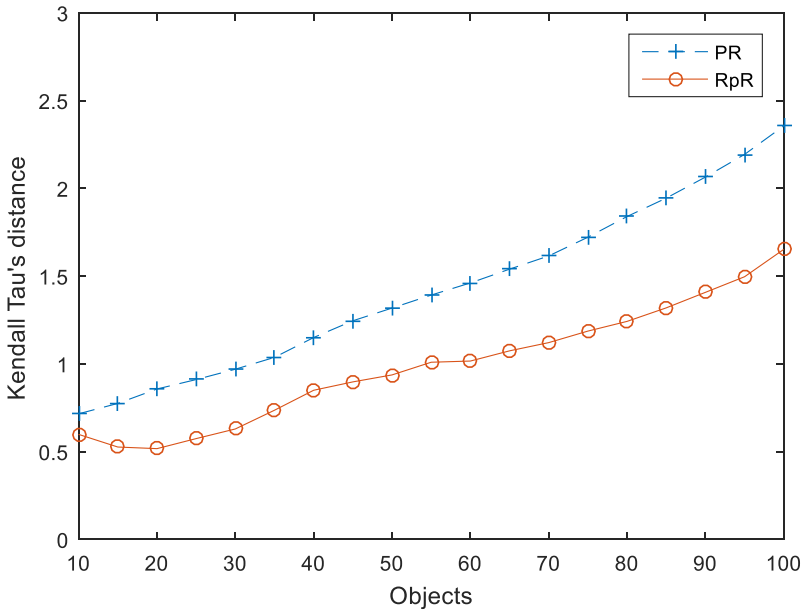


Figure 4-14: Kendall Correlation with ID.

4.5. Discussion

The typical reputation models like the ones stated in Section 2.2.2. , often suffer from biased or discriminative recommendations. To avoid such malicious intentions, proposed approach identifies two types of reputations depending on trustors social proximity with its surrounding objects. The first category represents reputation received from friends (named as recommendations) who represent a more trustworthy cluster and the second category represent more generic reputations from all other objects who are away from the close circle. According to social IoT concept, friends tend to be more honest than foreign objects and hence more reliable feedback can be expected in contrast to typical reputation systems. Nevertheless, proposed model filters objects who provide an extremely high number of reputations towards others without having a good score for them self. This is particularly useful in e-commerce systems like eBay and Amazon [129] where sellers with malicious intentions try to improve the reputation of their services through fake reputations. Further, the proposed model is based on the PR algorithm and which in fact has proven to be a reliable solution to scalability issues when ranking millions of web pages. Hence, inarguably the proposed model is a prospective candidate to assess reputation in large networks where millions of objects interact with each other.

4.6. Chapter summary

In this chapter, we have proposed an algorithm called RpR to assess recommendations and reputations provided by the objects in a distributed SIoT environment. First, we have separated and identified the meanings of reputation and recommendations in SIoT and their importance when it comes to trust evaluation. Then, we have applied these concepts to the possible use case scenario and numerically assessed the trustworthiness of each object in the environment. After that, we formally examined the key properties like convergence, accuracy, and resilience against deceitful activities through a simulation. We observe that the proposed model provides a robust method to compute trust within a few iterations for thousands of objects accurately, especially with the downgrading feature for the untrustworthy objects over time. Finally, we demonstrated the effectiveness and performance of our algorithm over other well-known ranking systems like [38], [39].

CHAPTER 5: KNOWLEDGE BASED TRUST EVALUATION MODEL

5.1. Introduction

Trust is a crucial fact that affects the appetite of an object to consume a particular service or product offered by another object. Typically, trust can be seen as a metric used to evaluate social actors in consideration of mutual benefits, coordination, and cooperation. Stakeholders continuously update their trust data on others in response to the variations of perceptions, generated by direct interactions and based on the opinions of others who are around (indirect observations). Often, this can be observed in our everyday life where trust decisions are made.

In this chapter, we address the issue of understanding and evaluating knowledge that is an important TM in the trustworthiness evaluation process in social networks. First, we identify and define several TAs, which directly affect the knowledge acquisition of a particular interaction. Then, a numerical model is derived, which is built on many aspects such as object relationships, spatial and temporal properties of objects, and their behavioral history. Based on the outputs of this model, a final trust level is predicted using regression analysis. Finally, the effectiveness of our model is investigated through simulations.

5.2. Knowledge based Trust Metrics

The knowledge TM covers all aspects of direct trust evaluations, which provide a perception about a trustee before and during an interaction. To make this possible, it must provide relevant data to the trustor for its assessment. If a data feature can be represented using a quantitative measurement, then the result is a numerical value in a certain range. As an example, social relationships like co-location and co-work, credibility factors like cooperativeness, time-dependent features like the frequency and duration of interactions, and spatial distribution of relevant trustees compared to the trustor can be used as direct trust measurements. The TAs, which we evaluate are shown in Figure 5-1.

The relationship TAs defines the mutual relationship between the trustor and a trustee. It is reasonable to assume that if two objects have a noble relationship between them,

a higher trustworthiness can be expected between them. As an example, if the trustor and the trustee are operated in close proximity such as looking for a parking space near a supermarket, then both benefit (e.g. getting a vacant, close, easily navigable parking space) from their relationship based on location similarity that we have identified as co-location TA. Likewise, if the two objects are in a working relationship like car sharing in which one needs to provide a service and other needs to get the service, both can support each other via their co-work association.

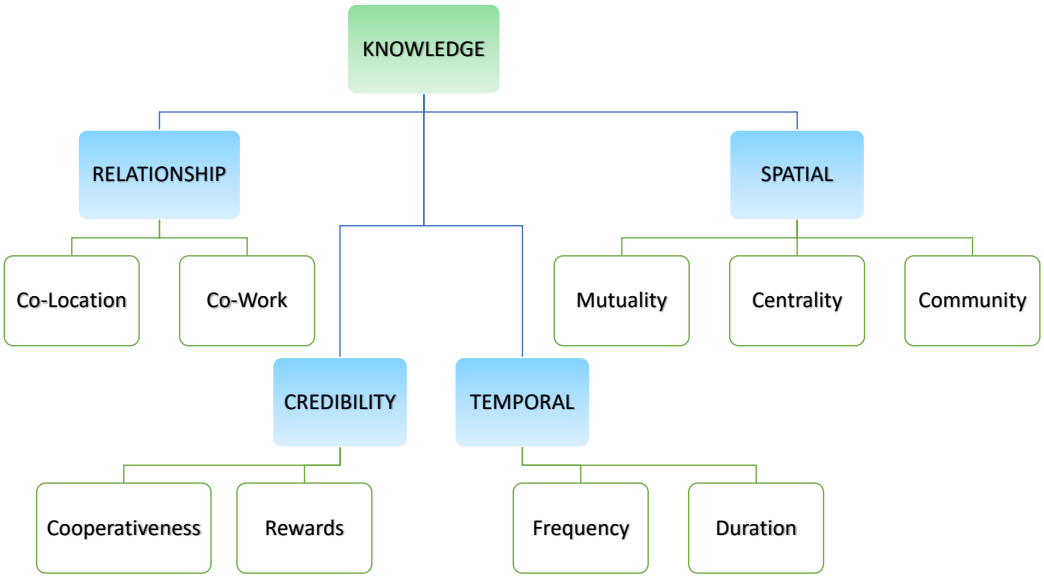


Figure 5-1: Composition of Knowledge TM based on TAs.

Furthermore, it is important to maintain knowledge about the consistency of trustworthy service provisioning. We discuss properties related to this issue under credibility. The cooperativeness under credibility represents the level of social cooperation from the trustee to the trustor. The higher cooperativeness means a higher trust level in an IoT ecosystem. A user can evaluate the cooperativeness of others based on social ties and select socially cooperative users. Additionally, we have introduced a rewarding system in order to track the history of misbehavior situations or unsuitable reactions originated by the trustee. Rewarding a TA can be used to either encourage or discourage further interactions with a particular trustee based on its past character.

To capture the significance of time-related information to trustworthiness evaluation, TAs like the frequency and duration of the interactions can be used. It is logical to assume that the higher the frequency and duration of interactions, the more trust is built up among the associating objects. On the contrary, the shorter time spent on each

other, the less knowledge is gathered on each other's actions and capabilities. As an example, in whitewashing attacks, a dishonest object can vanish for some time and rejoin the service in order to clear its reputation. However, if a trustor can keep a record of the consistency of the interested trustees then it can avoid such situations.

Moreover, in an IoT ecosystem, service provisioning (discover, manage and terminate) is based on its social relationships without solely depending on the underlying system level information. Therefore, it is vital to identify TAs, which determine the social proximity of the objects in collaboration. In this aspect, we identify three properties under spatial TAs in Figure 5-1 as mutuality, centrality, and CoI as governing features that define the social positioning of a trustee. Mutuality measures the degree of profile similarity between the trustee and trustor in resemblance to what is used in social networking. The community-interest represents whether the trustor and the trustee have a close relationship in terms of social communities, groups, and capabilities. Two objects with a high degree of community-interest have more opportunities for interacting with each other, and this can result in a higher trust level. Centrality measures the importance of a trustee among other objects with respect to a particular task and context.

5.3. Evaluation of Knowledge Based Trust

Even though an IoT environment produces a large amount of data, it is questionable how much of it can be directly used for the trustworthy evaluation process. Therefore, it is vital to extract trust features by scanning social and system level interaction logs and store them in a DR for further analysis. Hence, a numerical model that can extract basic features discussed in Section 5.2 is addressed here.

We define the assessment of knowledge (K) towards an object j by an object i at time t as $K^x_{ij}(t)$, where x represents one of the features: Co-location relationship (CLR), Co-work relationship (CWR), Mutuality and Centrality (MC), Cooperativeness-Frequency-Duration (CFD), and Reward. Note that a trust assessment is always between two or more objects.

5.3.1. **Co-Location Relationship (CLR)**

An IoT ecosystem enables users to share their resources, ideas, situations, and services with nearby devices. In such a situation, if both the trustor and the trustee are in close

proximity and they have subscribed to a DR in the platform, the trustor can conveniently get the required information from the selected trustee who is trustworthy in terms of the physical location compared to other objects far away from the scenario. However, in an IoT model, objects are always in relationship with their owner (Owner Object Relationship-OOR) and hence the static or dynamic nature of the OOR always affects the CLR [2]. In order to avoid objects leaving the physical location, a decision boundary based on the distance from the trustor (e.g. based on GPS data) and the time spent within this decision boundary are taken into consideration as shown in Figure 5-2. Then the objects, which are within this distance boundary and exceed the minimum time threshold inside the region, are selected as prospective candidates for a trustee. Once the candidates are filtered, their CL relationship with the trustor can be calculated as in (5.1).

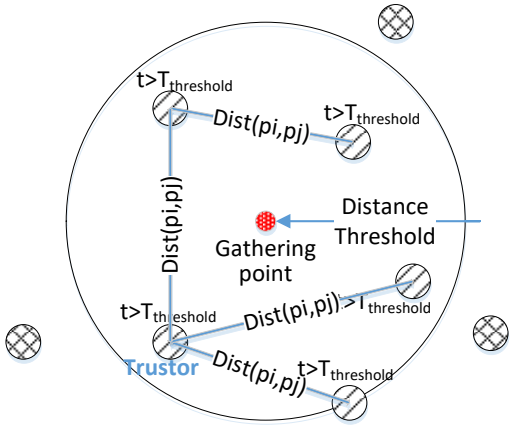


Figure 5-2: Decision Boundary for Objects in Close Proximity.

$$K_{ij}^{\mathrm{CLR}}(t) = \frac{1}{\mathbf{dist}(i,j)} \frac{\mathrm{G_i G_j}}{\|\mathrm{G_i}\| \|\mathrm{G_j}\|} \tag{5.1}$$

Here, g_i and g_j are the GPS coordinates of the trustor *i* and trustee *j*, respectively. The symbol "‖.‖" defines the norm of an element. The second term $\mathrm{G_i G_j}/\|\mathrm{G_i}\|\|\mathrm{G_j}\|$ in the equation (5.1) is the cosine similarity between the two objects and it is normalized by the geo distance factor $\mathbf{dist}(i,j)$ which can be calculated as in [130]. The application of the geo distance factor is important here as it provides a value with respect to an actual surface distance of the earth in contrast to a linear distance.

5.3.2. **Co-Work Relationship (CWR)**

The objects that are collaborating in common IoT applications can be characterized as CWR. In such a situation, more focus would be on a working relationship in a

particular service domain rather than physical proximity. To measure CWR as a numerical value, we compare the multicast interactions between a trustor and a trustee, as calculated below:

$$K_{ij}^{\mathrm{CWR}}(t) = \frac{\left|\mathbf{c}_{ij}^{\mathrm{MI}}\right|}{\left|\mathbf{c}_{j}^{\mathrm{MI}}\right|} \tag{5.2}$$

where $\mathbf{c}_{ij}^{\mathrm{MI}}$ is the vector of multicast interactions (MI) between trustor i and trustee j, and $\mathbf{c}_{j}^{\mathrm{MI}}$ is the vector of MI originated at j. The symbol "|. |" represents the determinant of a vector. $K^{\mathrm{CWR}}{}_{ij}(t)$ represents a relative measurement of shared multicast messages to total messages originated at the trustee.

5.3.3. **Cooperativeness, Frequency, and Duration (CFD)**

In a collaborative environment, it is important that every object fulfill its commitment to improving the level of the outcome of the whole service provision process. As an example, consider a malicious agent that provides fake ratings for a specific service. In this case, it is obvious that this agent deliberately tries to manipulate the genuineness of the information on the service and does not have any intention to use it. Therefore, the cooperativeness TA is vital to maintaining the above-mentioned content stability and thereby to provide a trustworthy service to the trustor upon its request. Furthermore, it can be anticipated that the more frequent and longer the interactions among objects, the more collaboration from each party can be expected. Based on this, a numerical model for cooperativeness, frequency, and duration is derived.

Let us consider a set of interactions, c_1, c_2... c_n over some period in which the trustor is interested. A trust level between trustor i and trustee j is calculated below:

$$K_{ij}^{\mathrm{CFD}}(t) = \sum_{m=1}^{n} \frac{\mathrm{c}_m}{\mathrm{t}_m} E(c_m) \tag{5.3}$$

Here, n is the number of interactions, indicating how frequently they interact with each other. For the m^{th} successful interaction, c_m is the length of an interaction between the trustor and the trustee, t_m is the total interaction length by the trustee. The factor c_m/t_m assesses the duration property, in which the trustee interacts with the trustor, relative to the total activity time of the trustee. $E(\mathrm{c}_m)$ is the binary entropy function which

measures the balance in the interaction or the cooperativeness which can be calculated as follows [51]:

$$E(\mathrm{c}_m) = -p \log p - (1-p)\log(1-p) \tag{5.4}$$

where p is the fraction of the interactions between the trustor and the trustee. $E(\mathrm{c}_m)$ follows a binary distribution as stated in [131]. It is obvious that the maximum entropy (i.e. $E(\mathrm{c}_m)$=1) is reachable only when p=0.5 that is 50% contribution from each party.

5.3.4. **Reward System (RS)**

An essential component of any service provisioning system needs to have a reward and punishing mechanism or a feedback model in order to assess the historical service experiences between a trustor and a trustee. It is always critical to maintaining the social relationships at the maximum trustworthy level and hence we use the exponential downgrading formula shown in equation (5.5) for this purpose.

$$K_{ij}^{\mathrm{RS}}(t) = \frac{\|n\| - \|n_{\mathrm{p}}\|}{\|n\|} e^{\left(-\frac{\|n_{\mathrm{p}}\|}{\|n\|}\right)} \tag{5.5}$$

Here, $\|n\|$ is the total number of interactions that have taken place during a period t, and $\|n_{\mathrm{p}}\|$ is the total number of unsuccessful or suspicious interactions. To punish misbehavior situations more severely, the slope of the distribution is increased by a factor of n_{p}, compared to the standard exponential distribution. Hence, a higher number of malicious interactions will result in a lower reward value.

5.3.5. **Mutuality and Centrality (MC)**

In an IoT ecosystem, service discovery and provisioning largely depend on the social relationship among the participating objects. In this regard, the mutuality and the centrality TAs define the location of a trustee with respect to a trustor in a social world. On the other hand, it is very intuitive to assume that a higher number of mutual objects imply a higher similarity between their social profiles. However, mutuality alone cannot be used as a TA due to the number of mutual friends being proportional to the number of friends of each individual object. That is, an object with a higher number of friends gets an additional advantage compared to an object that has recently joined the network but has higher trustworthiness. In order to avoid such circumstances, a

relative measurement of mutuality compared to the total number of friends is considered. If M_{ij} represents the set of common friends between i and j, and N_i is the set of trustee's friends, then the centrality property can be calculated as below:

$$K_{ij}^{\mathrm{MC}}(t) = \frac{|M_{ij}|}{|N_i|} \tag{5.6}$$

5.3.6. **Community of Interest (CoI)**

Objects in an IoT environment usually collaborate with at least one community. As an example, a person is registered as a frequent customer of a car sharing community while being a member of several other communities like online markets, social networking groups, etc. If another person is also a member of the car-sharing community, this shows the resemblance between both persons' interests. Similarly, if the trustor and the trustee share common interest groups, that is an indication of the degree of the common interest or similar capabilities of the trustee compared to the trustor. Mathematically, let us define M_{ij}^{coi} as the set of communities where both the trustor and the trustee are involved in, and N_{ij}^{coi} as the set of communities with each including the trustee as a member. Please note that both the trustor and the trustee can be a member of several communities and hence the trust level of the trustee based on CoI is calculated in (5.7).

$$K_{ij}^{\mathrm{CoI}}(t) = \frac{|M_{ij}^{\mathrm{CoI}}|}{|N_i^{\mathrm{CoI}}|} \tag{5.8}$$

5.3.7. **Assessing Final Trust Value**

After modeling the TAs using equations (5.1), (5.2), (5.3), (5.4), (5.5), (5.6), and (5.8), the next step is to calculate the final trust value of the trustee. A well-known approach is to combine each TA through a linear equation with weighting factors as shown in (5.8).

$$K_{ij}(t) = \alpha K_{ij}^{\mathrm{CLR}}(t) + \beta K_{ij}^{\mathrm{CWR}}(t) + \gamma K_{ij}^{\mathrm{CFD}}(t) + \varepsilon K_{ij}^{\mathrm{RS}}(t) + \delta K_{ij}^{\mathrm{MC}}(t) + \eta K_{ij}^{\mathrm{CoI}}(t \tag{5.9}$$

However, there are many drawbacks in this approach, including (i) lack of information and an infinite number of possibilities when it comes to estimating a weighting factor, (ii) unsuitability of a threshold based system to detect the trustworthiness of a

particular trustee, and (iii) inability to identify which TA has the most influence on trust in a particular context. Thus, we propose the Multiple Regression (MR) based methods to evaluate the future trust level based on knowledge as in equation (5.10)[132].

$$K_{ij}(t) = b_0 + \sum_{l=1}^{n} b_l K_{ij}(t) + \epsilon(t) \tag{5.10}$$

where $K_{ij}(t)$ is the series under investigation, and n is the order (length) of the model and b_0 is the estimated constant and b_i is the prediction coefficient of the i^{th} independent variable (attribute). $\varepsilon(t)$ is the error term and ignored for the simplicity in the simulation model which gives an estimated model.

Based on the above discussion, the overall principle of the assessment of the knowledge based trust metric is shown in Figure 5-3. It specifies three major tasks: (i) identify set of TAs that can be used to evaluate trustees' trustworthiness; (ii) model each of the TA based on the available data; and (iii) an aggregation mechanism to combine all the TAs to derive numerical value for knowledge based TM.

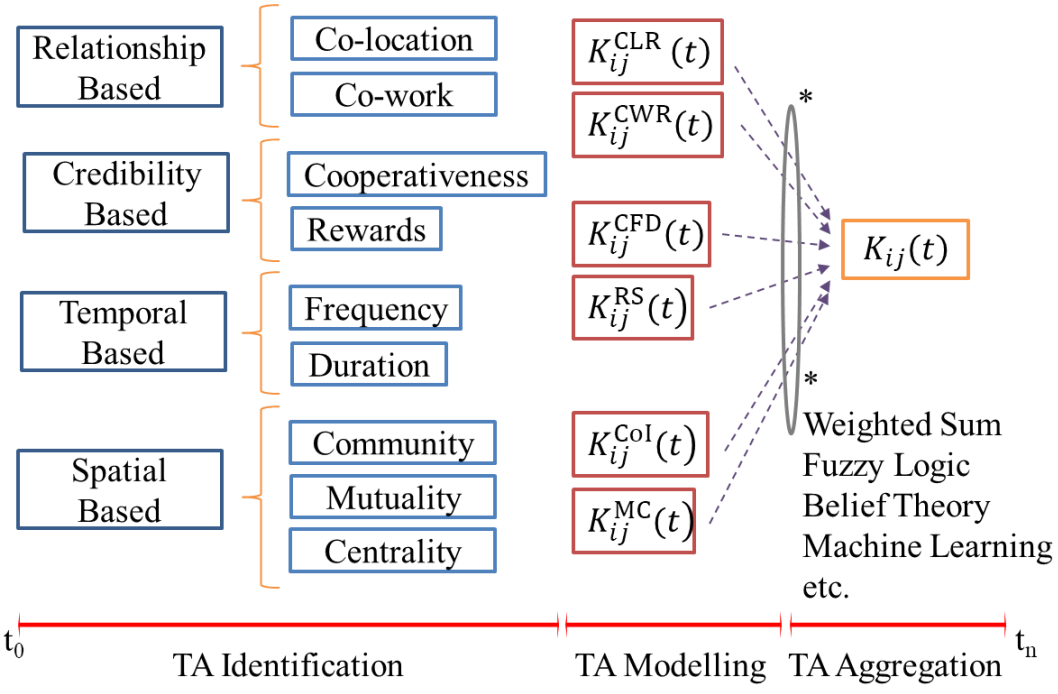


Figure 5-3. Overall principle of the assessment of the knowledge.

5.4. Experiments and Results

5.4.1. Experiment Setup

To evaluate our model, we would need traces of many objects. However, we find they are not publicly available for SIoT at least at the time of preparing this research. Hence, we have used traces taken at the SIGCOMM-2009 conference which is available in CRAWDAD [133], [134]. These traces contain the information on device proximity, activity logs, friendship information, interested groups, application level message logs, and data layer transmission logs which are acquired through the MobiClique mobile app as shown in Figure 5-4 [133]. We map the information to match with the IoT concepts described in [9]. In other words, we define a set of features, CWR, CFD, RS, MC, and CoI, related to IoT based on raw data found in the data set. Therefore, our experiment can be repeated with any real world IoT data set for further experiments without any ambiguity. This leads to the parameter settings and scenario of our simulation, as detailed in Table 5-1. Among 76 nodes, each pair of them (Trustor and Trustee) with at least a single interaction between them are considered as objects to match with the IoT concepts.

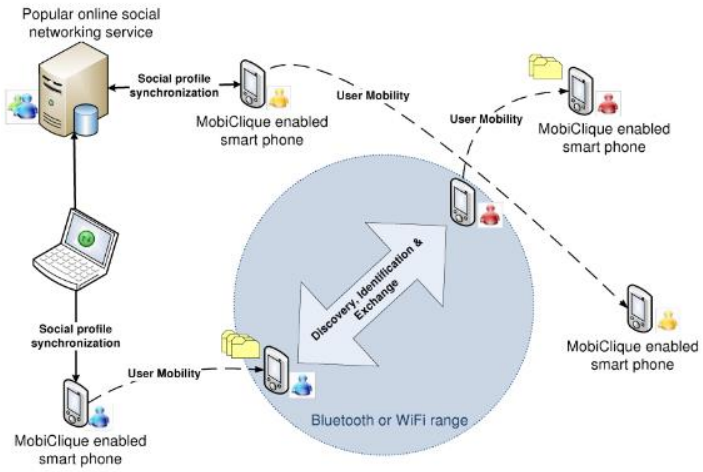


Figure 5-4. System architectures of the experiment.

Table 5-1: Parameters of the data set.

Parameter	Value	Parameter	Value
Nodes	76	Interactions	18226
Objects	5776	Communities	711
Messages	899	Message Type (UC/MC/BC)	266/57/576

5.4.2. **Simulation Results**

In this section, we present the simulation performance of our model discussed in Section 5.3. The simulation complexity is based on the number of interactions among objects and the number of nodes. For our model, around 18,000 interactions among 5,776 object pairs are used. The results based on the numerical models are shown in Figure 5-5 and Figure 5-6. Distribution of Trustworthiness Relative to a Specific Object.. The numerical result obtained for K_{ij}^{CLR} using Equation (5.1) is shown in Figure 5-5(a). The X-axis shows the Trustor (1st number) – Trustee (2nd number) pairs and Y-axis shows how the trust level changes based on the CLR. As the data set is based on the conference location, the CLR value is quite similar in each object pair as they are created in close proximity. Consequently, Figure 5-5(b) shows the effect of CWR which is based on the MC conversations analogy to data layer multicast messages. It can be observed that significantly fewer pairs are willing to create co-work relationships among them.

Figure 5-5(c) shows how the trust changes with cooperativeness among objects and also the frequency and duration of their conversations. It is visible that cooperativeness is distributed in the middle of the graph as often RF communication is limited to asymmetric as well as the short duration of message exchanges. Similarly, we have evaluated the trust level based on CoI and the centrality of the trustee object for the trustor as shown in Figure 5-5(d) and Figure 5-5(e). However, Figure 5-5(f) shows that most of the penalty coefficients are distributed at the low end of the graph i.e. low level of trustworthiness. This is mainly due to the unsuccessful interactions or misbehaviors occurring in the past conversations.

Similarly, Figure 5-6 shows the distribution of trustworthiness for each object (Trustees) with respect to one specific object (Trustor). For example, let us consider Figure 5-6 (d) in which trust variation with respect to object “45” is analyzed. This figure clearly shows the interpretation of the trustor’s view on other adjacent objects with respect to the features we have discussed in Section 5.3. As an example, trustee object “34” shows high CLR with the trustor (“45”) compared to other features while MC, CFD, and Reward is around 0.4, 0.15 and 0.16 respectively. Therefore, it is possible that the trustor will engage in location-based services with the trustee in future

interactions but limit its interactions related to collaborative services, as the MC and CFD values are low.

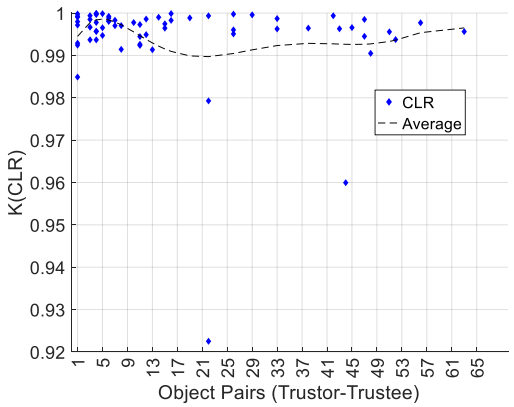
As a final part of our numerical model, we have performed a multiple regression analysis in order to predict future trust levels based on the values of current attributes as an alternative to simple weighted summation. To show the result clearly, the impact of penalty (or the reward) and centrality on trust is shown in Figure 5-7. Based on this, the trustor can predict what would be the next possible success rate for specific values of attributes, or the values, which must be satisfied to achieve a certain level of trust via direct observations.

5.5. Discussion

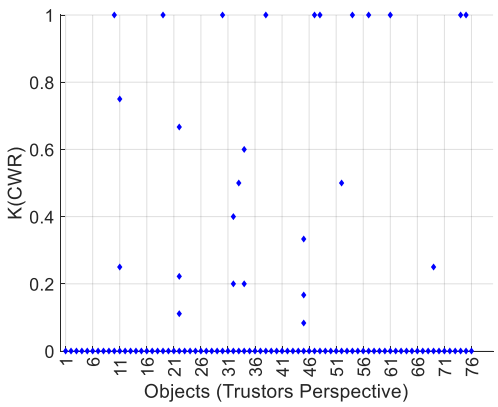
In contrast to the literature work discussed in Section 2.2.3. , this chapter proposes three main contributions, (i) a rational way to identify TAs which represent both social and non-social domain of an environment; (ii) a numerical way to model the TAs for trust evaluation; and (iii) feasibility of using a non-linear trust aggregation technique compared to linear summation. The first contribution basically is a leap forward for the existing systems in which both dependable and social properties are considered together to model knowledge TM. For example existing work like [77], [106], [107], [108], [109], [110], [81], [111] are mostly concentrated to represent knowledge based trust on dependable properties in which subjective nature of trust is completely ignored. On the other hand, research like [112], [113], [38], [39], [40] propose solutions which are specific to particular environments like p2p, MANET, WSN, etc ignoring the social aspect of trust as in SIoT. However, the proposed research in this section, consider subjective, social and dependable aspects of trust in the process of knowledge based trust modeling which in fact provide a more holistic view about the situation in the decision-making process.

On the other hand, there is some work in the literature which propose a numerical approach to model certain TAs as in [66], [77], and [135]. However, the experiments in these work quite biased towards assessing non-social properties that are even with synthesized data sets. On the contrary, the proposed numerical models are capable of assessing social trust aspects of knowledge and it shows promising performance against detecting malicious objects in a real-world environment in contrast to synthesized data. Further, the thesis proposes multiple regression based trust

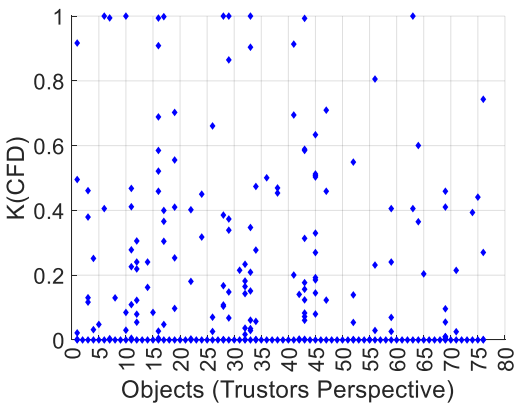
aggregation algorithm in place of existing methods like [65], [88], [89], [90] in which nonlinear nature of trust aggregation is ignored. Additionality to the process of aggregation, such method enables the decision-making system to predict future trust values based on the current knowledge.



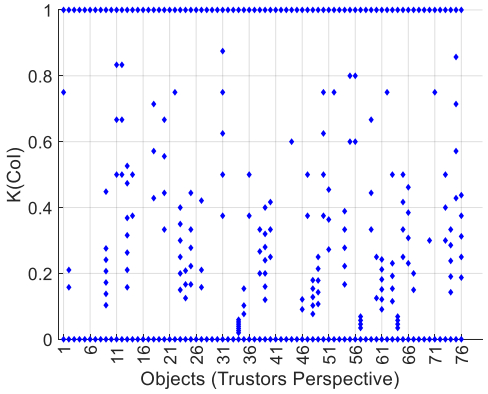
(a) CLR



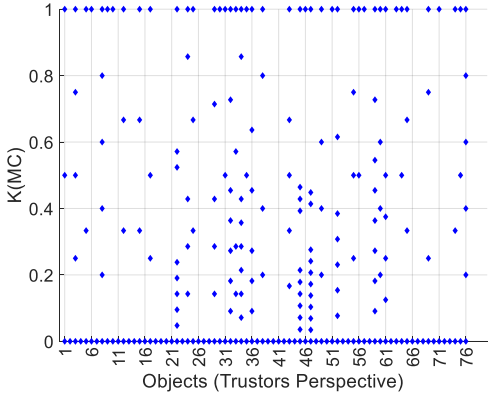
(b) CWR



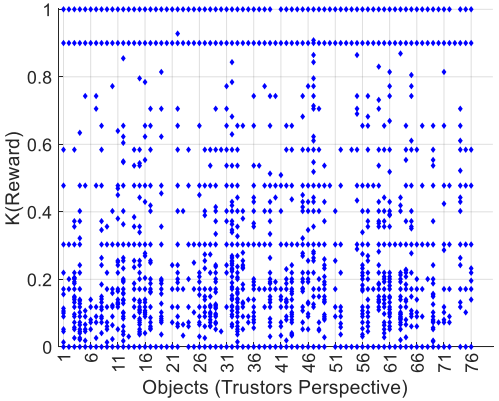
(c) CFD



(d) CoI

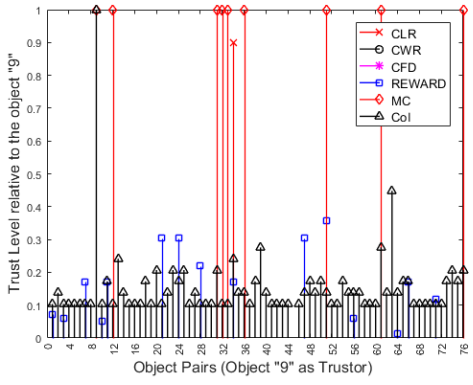


(e) MC

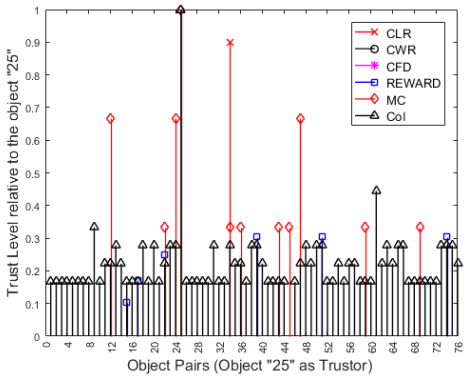


(f) RS

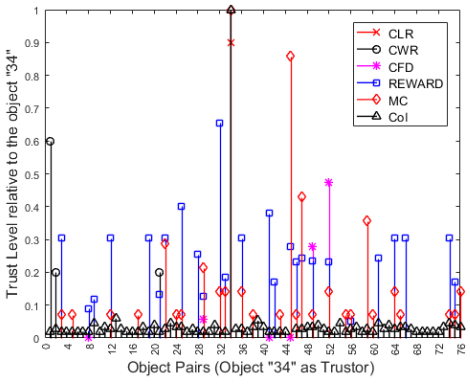
Figure 5-5: Impact of TAs on Knowledge TM.



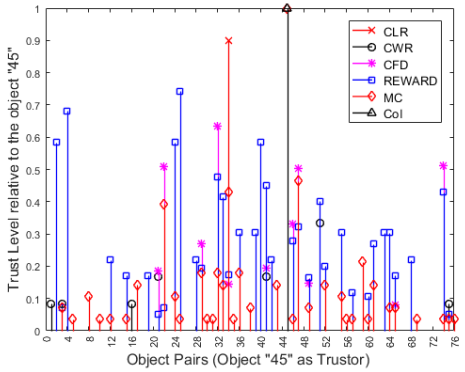
(a) Relative to Object “9”



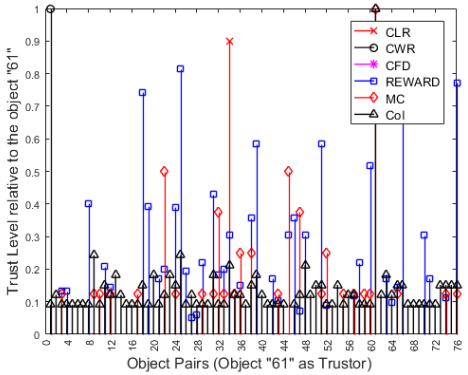
(b) Relative to Object “25”



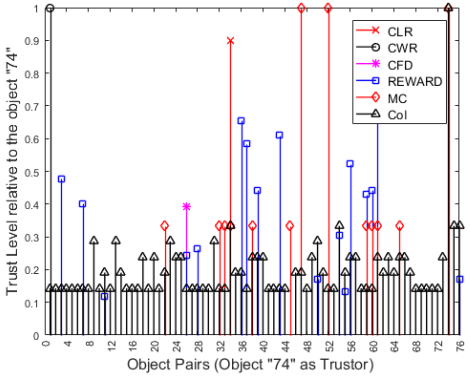
(c) Relative to Object “34”



(d) Relative to Object “45”



(e) Relative to Object “61”



(f) Relative to Object “74”

Figure 5-6. Distribution of Trustworthiness Relative to a Specific Object.

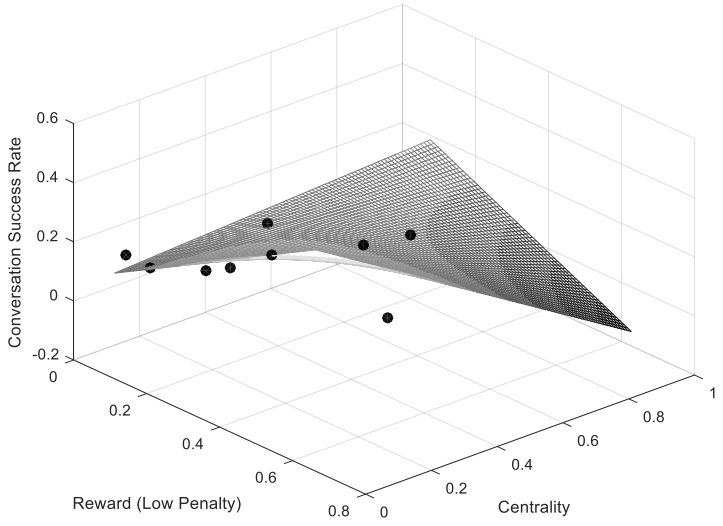


Figure 5-7: Prediction of Trust using MR.

5.6. Chapter summary

This chapter focused on evaluating "knowledge" which is a vital TM in trust assessment in SIoT. First, this work identifies several TAs after careful consideration, which directly affects the knowledge TM. Then, based on the SIoT concepts we present a numerical approach to estimate individual TAs. To demonstrate the usefulness of our model, we have considered a real-world scenario and analyzed the impact of each parameter on knowledge in a simulation environment. Finally, we propose a prediction technique in order to find future values of knowledge based on the multiple regression method that is an effective alternative to a weighted summation of attributes.

CHAPTER 6: MACHINE LEARNING BASED TRUST EVALUATION MODEL

6.1. Introduction

As we described in Chapter 3, there is a number of trust modeling techniques that can be observed in the literature. However, the influence of a particular TA on trust is often determined by a weighting factor, but the assessment of a proper weight is a complex task due to the fact that trust is a varying quantity which depends on many factors, e.g. expectations of a trustor, time and context. Thus, schemes that are more intelligent are required to find these weighting factors and a threshold that defines a trustworthy boundary.

Even though complex characteristics of trust make it challenging for traditional analysis, but ideal for the application of artificial intelligence, ML techniques, and big data analytics. The objective of AI for trust is to investigate the very large volumes of data produced by various components in the IoT ecosystem and transform this data into meaningful outputs such as trust-based decision making, fault detection, service composition and generate ultimate wisdom.

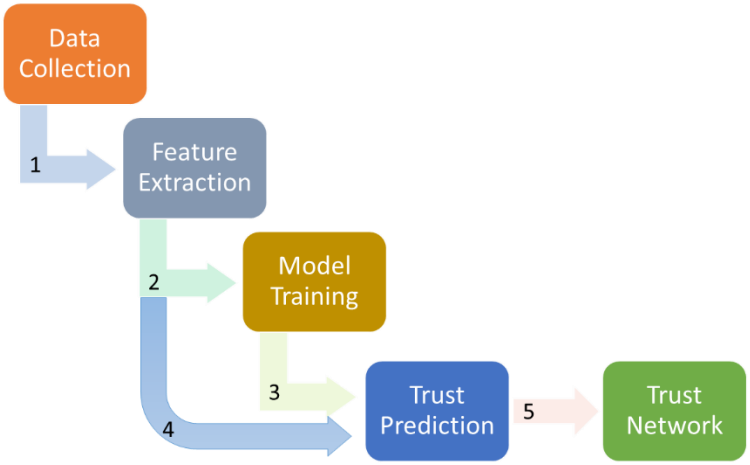


Figure 6-1: Steps for Trust evaluation based on ML techniques.

A generic process of trust assessment using ML techniques is shown in Figure 6-1. The overall process consists of five-sub process, namely, data collection, feature extraction, model training, trust prediction, and trust network buildup. The Data Collection module retrieves necessary data records from the IoT ecosystem and the Feature Extraction module generates a trust feature vector for each record. They serve

as the data preparation and preprocessing components of the model. Then, the Training module trains the classification model and the Trust Prediction module infers the trust level among users based on the training model. The overall process until this point is illustrated in Figure 6-2. Finally, the Trust Network buildup module filters distrustful relationships and creates a trustworthy network based on the estimated trust values of each object which is beyond the scope of this research.

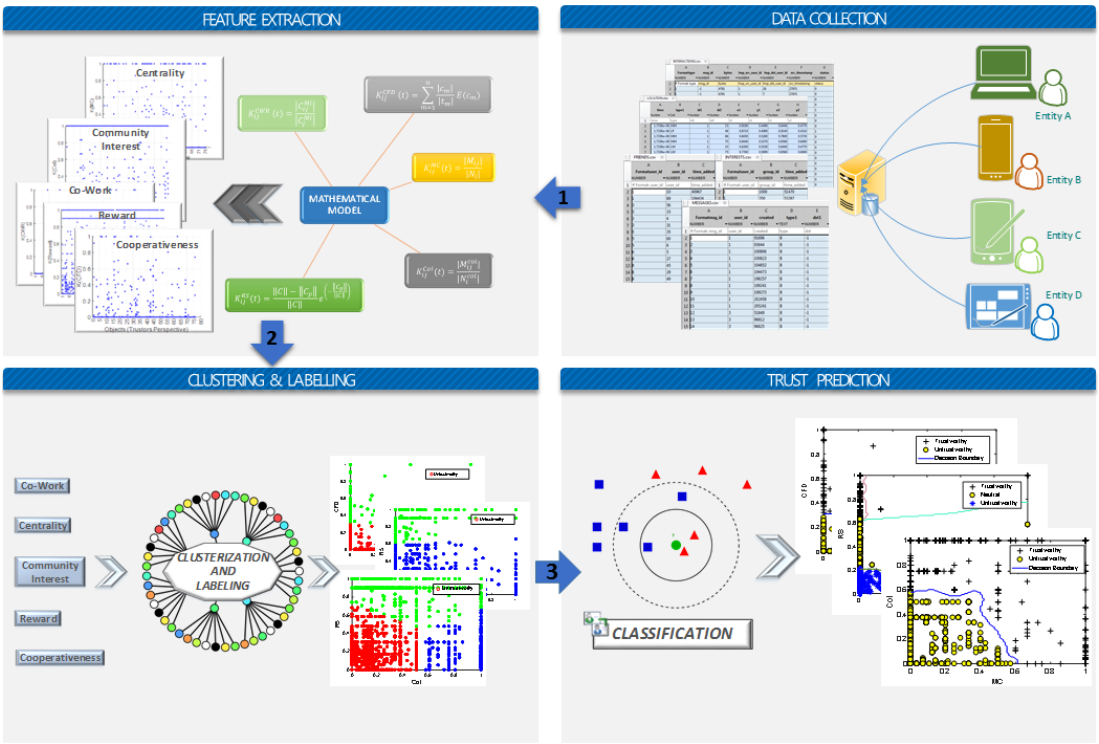


Figure 6-2. Principle of the machine learning based trust model

6.2. Machine Learning Based Trust Model

To overcome the weakness of the TA aggregation and make the trust assessment mechanism autonomous, we propose an ML based model to analyze the TAs extracted in Chapter 5 and predict the trustworthiness of prospective transactions based on a trained model. In order to achieve this, we first use an unsupervised learning algorithm to identify two different clusters or labels, namely trustworthy and untrustworthy. The main reason to use an unsupervised learning method over a supervised method is due to the unavailability of a labeled training set based on trustworthiness relationships.

Then a multi-class classification technique such as SVM is used to train the ML model in order to identify the best threshold level that separates trustworthy interactions from others. In this chapter, our main objective is to differentiate malicious interactions from

trustworthy interactions with maximum boundary separation and minimum outliers rather than classification itself. Therefore, it is not necessary to go for other algorithms like Random Forest, especially with a low dimensional dataset compared to its sample size used in this work. However, depending on the data set, dimensionality, a number of classifications required and noise levels of the samples, a model comparison can be performed to find out the best possible algorithm for each individual case. A well-trained model like this can differentiate an incoming interaction between two or more objects much more efficiently than linear weighting methods [77], [136] and is much more beneficial in the decision-making process.

Let us define the number of features considered in the model as n and the length of the training set as m. We use the features defined in Chapter 5, i.e. CWR, CFD, RS, MC, and CoI to train our model. They are expressed as a feature matrix $X^{(i)}_{(j)}$ where i denotes the i^{th} training sample and j signifies the j^{th} feature among the n features. Moreover, the label of each training sample i is denoted by $\mathbf{y}^{(i)}$. However, training labels are not readily available and a method for labeling will be discussed in Section 6.2.1. This allows us to identify each training set as $(X^{(i)}_{(j)}, \mathbf{y}^{(i)})$ for i=1,2,…,m and $\boldsymbol{j}$=1,2,…,n. In the following subsections, we break down our main algorithm into two parts and explain it separately in Section 6.2.1, and 6.2.2. respectively.

6.2.1. **Clustering and Labelling**

In this section, we develop an algorithm based on the K-means clustering technique, which is specified in detail in Algorithm 6-1, in order to group interactions based on the aforementioned features and thereby label each interaction as trustworthy or untrustworthy [137]. As our main objective is to differentiate malicious interactions from trustworthy interactions with maximum boundary separation and minimum outliers, the model comparison is omitted here. However, depending on the data set, dimensionality, number of classifications required and noise levels of the samples, a model comparison can be performed to find out the best possible algorithm for each individual case.

The K-means algorithm needs to define two initial conditions: number of clusters (k) and initial centroid positions (μ) that each interaction is assigned to. As there is no way to find out these values at the beginning of the algorithm, we randomly assign initial centroid locations for a range of cluster sizes, e.g. from k=1 to k=5. After that, steps 4-9 in Algorithm 6-1Algorithm 6-1. Data Clustering and Labelling.

are executed until the cluster points "μ" are not changing any further (i.e. until the convergence). Then, the Elbow method is used to find out the optimum cluster size which gives the lowest value for the K-mean cost function J(c,μ) where c is the index of a cluster centroid and μ is the coordinates of cluster centroids with the dimension of k [137].

Note that initial inputs to the algorithm were normalized between [0, 1] in which "0" represents untrustworthy and "1" represents trustworthy. Hence, it is logical to label points close to "0" as untrustworthy and vice versa after the cauterization step. Therefore, after the step 13 of the algorithm, the clusters close to the origin (i.e. all zero point) of the N-dimensional space are marked as "0" or untrustworthy and the cluster away from the origin is identified as a trustworthy region. To check the influence of all n features at once, the Principal Component Analysis (PCA) algorithm based on Singular Value Decomposition (SVD) is applied to reduce the N dimensions to two dimensions for visualization purposes as in Algorithm 6-2, before applying Algorithm 6-1[138]. Even though it is possible to extend Algorithm 6-1 for n features with regularization, we observe that the PCA method is more efficient with respect to the computational complexity of unsupervised learning with regularization.

```
1.  Input: X       Output: y
2.  Initialize cluster centroids μ1, μ2,… ,μk ∈ℜn
3.  for k=1 to 5 do
4.      Repeat until convergence: {
5.          for i=1 to m do
6.              c(i):=arg minj||X(i)- μk||2
7.              μk := Average of points assigned to cluster k
8.          end for
9.      }
10.     J(k)(c,μ):=arg mink J(c,μ)
11. end for
12. Optimum k ← Elbow method ← plot J(k) vs k
13. for i=1 to m do
14.     if c(i) close to (0,0)
15.         y(i) = 0
16.     elseif
17.         y(i) = 1
18.     end if
```

Algorithm 6-1. Data Clustering and Labelling.

```
1.  Compute dot product matrix          : Σ=XᵀX
2.  Compute eigenvectors                : [U,S,V]=SVD(XᵀX)
3.  Specify the required dimension ,d   : Ud=[u1,…ud]
4.  Compute d(=2) features              : Z= UdᵀX
```

Algorithm 6-2. Principal Component Analysis.

The first step of the PCA algorithm is to calculate the covariance matrix Σ that has the dimension of n x n. In step two, principal components U and V are calculated using the SVD function, each having the same dimension as Σ [138]. As our intention here is to reduce the dimensions from five to two, dimensions (d) of the principal matrix U is set to two. Finally, step four calculates the two-dimensional feature vector Z corresponding to the five-dimension vector X.

6.2.2. **Classification Model**

Having obtained the completed data set ($X^{(i)}_{(j)}$, $\boldsymbol{y}^{(i)}$) via Algorithm 6-1, the next step is to train an algorithm based on an SVM technique which can identify the nonlinear boundaries of trustworthy and untrustworthy interactions. In order to obtain the maximum accuracy of the learning algorithm, the training set is divided into two parts in such a way that the training set occupies 80% of the data and 20% for the cross-validation data set which is denoted as ($X^{(i)}_{\text{val}}$, $\boldsymbol{y}^{(i)}_{\text{val}}$) for i=1,2,…, ⌊0.2*m⌋ and j=1,2,…,n. This is important to avoid overfitting data through the regularization parameter and variance.

In Algorithm 6-3, we use a Radial Basis Function Kernel (RBFK) due to the smaller number of features (n) compared to the training set samples (m) as the authors in [139] have claimed. Furthermore, in order to optimize the computational resources, the LIBSVM library is used to run the RBFK kernel [140]. First, we run the RBFK kernel over multiple instances of regularization parameters and variances in order to find optimum parameters for the learning algorithm as shown in steps 4-7 in Algorithm 6-3. As an example, both c (regularization parameter) and γ (variance) are varied as a geometric series (e.g. 0.01, 0.03, 0.09… 30) to save the time and computational resources. Then the parameters that give the minimum error in the prediction step are chosen as the optimum factors for the SVM model. It is essential to improve the accuracy of the final ML model and suppress any noise generated by the previous clustering algorithm. Hence, we use regularization techniques to avoid such issues during the training process in Algorithm 6-3.

After estimating best parameters for c and γ, the algorithm is trained for all the training data samples using Algorithm 6-3 and model parameters are recorded to estimate future trust values based on the incoming feature statistics. The function *svmtrain* is

defined in the LIBSVM library (MatLab environment) and calculates the decision boundaries based on the RBFK kernel as per the SVM technique. Similar to Algorithm 6-1, first we consider two trust features at a time and investigate the trust boundaries. After that, features derived through the PCA algorithm are considered to investigate the effect of all five features on the trust boundaries.

```
1.  Input: X, y, X_val, y_val
2.  Output: Weights and Decision boundary
3.  //Find best parameters for c and γ
4.  for c, γ=0.01 (multiple of 3) 30 do
5.      model=svmtrain(y,X,RBFK,c, γ)
6.      prediction=svmpredict(y_val, X_val ,model)
7.      error [c, γ]= predictions ≠ y_val
8.  end for
9.  Choose c, γ ← minimum [error]
10. [weights, accuracy, decision values]= svmtrain(y,X,RBFK,c, γ)
```

Algorithm 6-3. Classification Model.

6.3. Experiments and Results

6.3.1. Simulation Setup

For simulation, we use the same feature set which is already modeled in Section 5.3. After obtaining the trust feature vector X_j for each node pair, they are organized as shown in (6.1) to generate the *m* training samples. We have deliberately omitted the results from CLR as the data set itself was obtained when all objects were within very close proximity and it is not meaningful to test location-based trust in this scenario. The dimension of the training sample matrix is in the order of $m \times n$, where m=5776 (node pairs) and n=5. The notation $[.]^{\mathrm{T}}$ is used to denote the transposition of a vector and has the dimension of $m \times 1$. Note that feature normalization is not required here as each feature value is in the range of 0 and 1.

$$[\boldsymbol{X}]_{m\times n} = \begin{bmatrix} \vdots & \vdots & \vdots & \vdots & \vdots \\ [CWR]^{\mathrm{T}} & [CFD]^{\mathrm{T}} & [RS]^{\mathrm{T}} & [MC]^{\mathrm{T}} & [CoI]^{\mathrm{T}} \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \tag{6.1}$$

For the multiclass calcification problem, 4620 samples (i.e. 80% of the total samples) are chosen as the training set, and 1156 samples (i.e. around 20% of the total data set) are used as cross-validation samples to avoid the data-overfitting problem. For both ML experiments, two features out of five are selected at a time for the sake of demonstration purposes, as it is not feasible to show a five-dimension vector. However, it is critical to analyze five features at the same time and evaluate their

influence on the final trust value. Therefore, we then consider all the five features together and generate numerical results. However, to demonstrate the results, the PCA method is used to reduce the dimensions from five to two and generate the graphical results [141]. Note that PCA not only simplifies the visualization but also the algorithm complexity that makes our model more practical in the case of a large number of features even though we use around five dimensions in this research to prove the effectiveness of our model in trust evaluation. Here, feature normalization is used to bring the new data samples, obtained through PCA, into the range of zero and one.

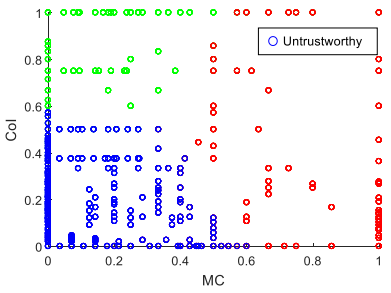
6.3.2. **Algorithm I: Clustering and Labelling**

With the successful abstraction of the trust features, the next step is to investigate how to combine each of them to generate a final trust value. To filter out most trustworthy interactions from untrustworthy interactions, the algorithm explained in Algorithm 6-1. Data Clustering and Labelling.

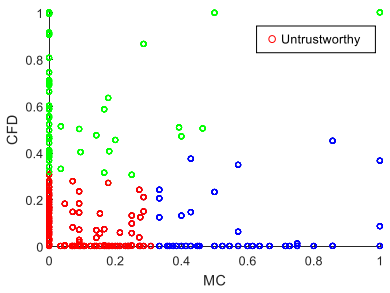
is applied and the results obtained are shown in Figure 6-3. In order to determine the optimum number of clusters, the Elbow method is used as shown in Figure 6-4. In certain feature combinations, the algorithm is capable of categorizing interactions into three groups: trustworthy, neutral, and untrustworthy. Instances, where the Elbow method gives K=3 represent such situations. The results clearly show the boundaries of separation from the untrustworthy interactions as marked in Figure 6-3.

As an example, let us consider Figure 6-3(a) which shows the distribution of trust values compared to the centrality and community interest. It can be observed that the region above MC=0.6 and CoI=0.6 is the trustworthy region with respect to these two features. Similarly, Figure 6-3(b) to Figure 6-3(g) show a clear boundary between the trustworthy and untrustworthy regions. However, Figure 6-3(h) and Figure 6-3(i) show slightly different results compared to others. In both figures, the trustworthiness boundaries are learned with one common feature: the reputation. Hence, it is noticeable that the algorithm finds a lower trust value when the reputation value is low, even with higher trustworthiness values for CFD or CWR. This is one of the interesting results as reputation is one of the critical factors when it comes to the trustworthiness evaluation process.

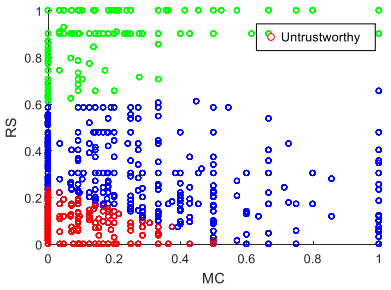
Note that we first run the algorithm pairwise to generate visual results and then combine all five features to find out the trustworthy region as shown in Figure 6-3(j) where PCA is used to reduce the feature dimensions from 5D to 2D for visualization purposes. To bring the new dimensions into the range of 0 and 1, feature normalization is implemented. It can be clearly observed that values around 0.5 on the 1st dimension and values around 0.7 on the 2nd dimension show the boundary between trustworthy and untrustworthy interactions.



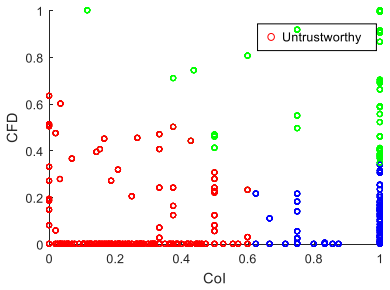
(a) MC and CoI



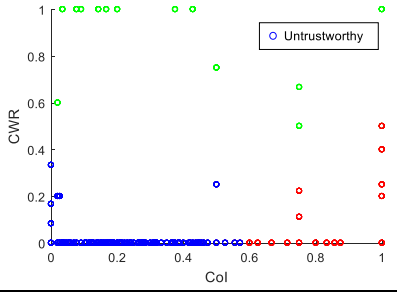
(b) MC and CFD



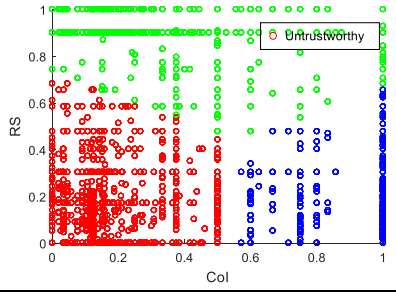
(c) MC and RS



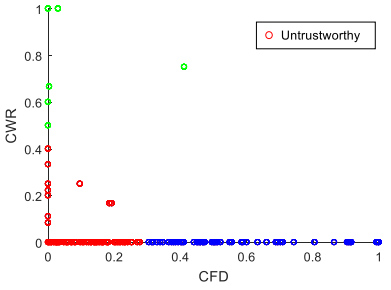
(d) CoI and CFD



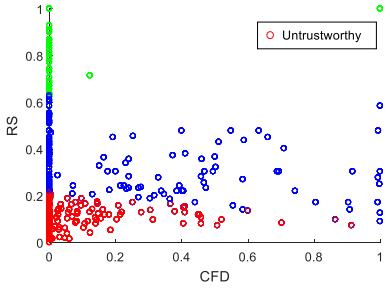
(e) CoI and CWR



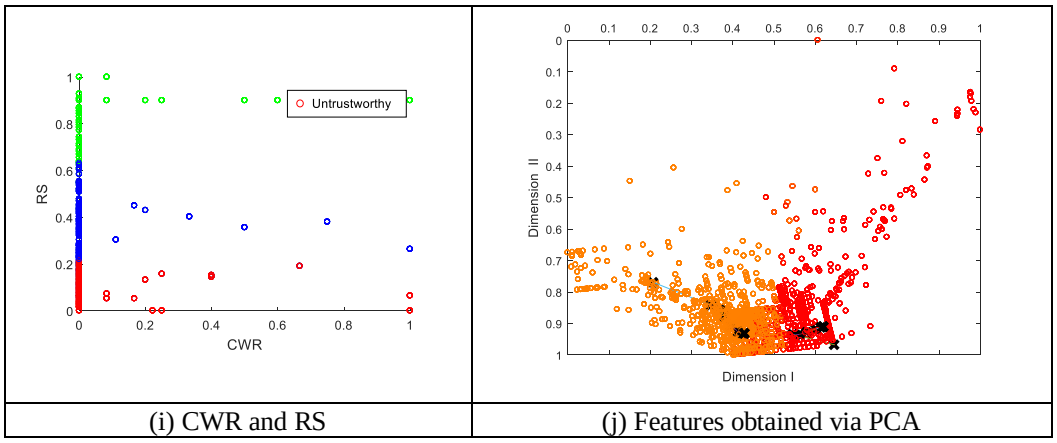
(f) CoI and RS



(g) CFD and CWR

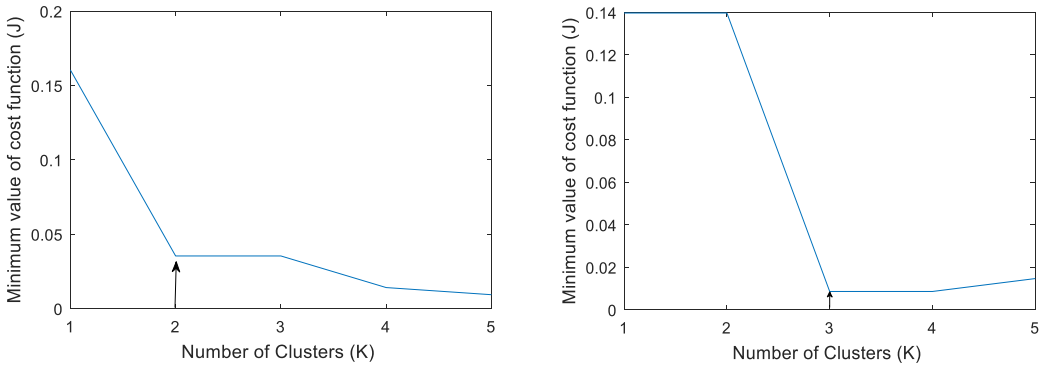


(h) CFD and RS



(i) CWR and RS | (j) Features obtained via PCA

Figure 6-3: K-means clustering on different pairs of features



(a) *k*=2 (b) k=3

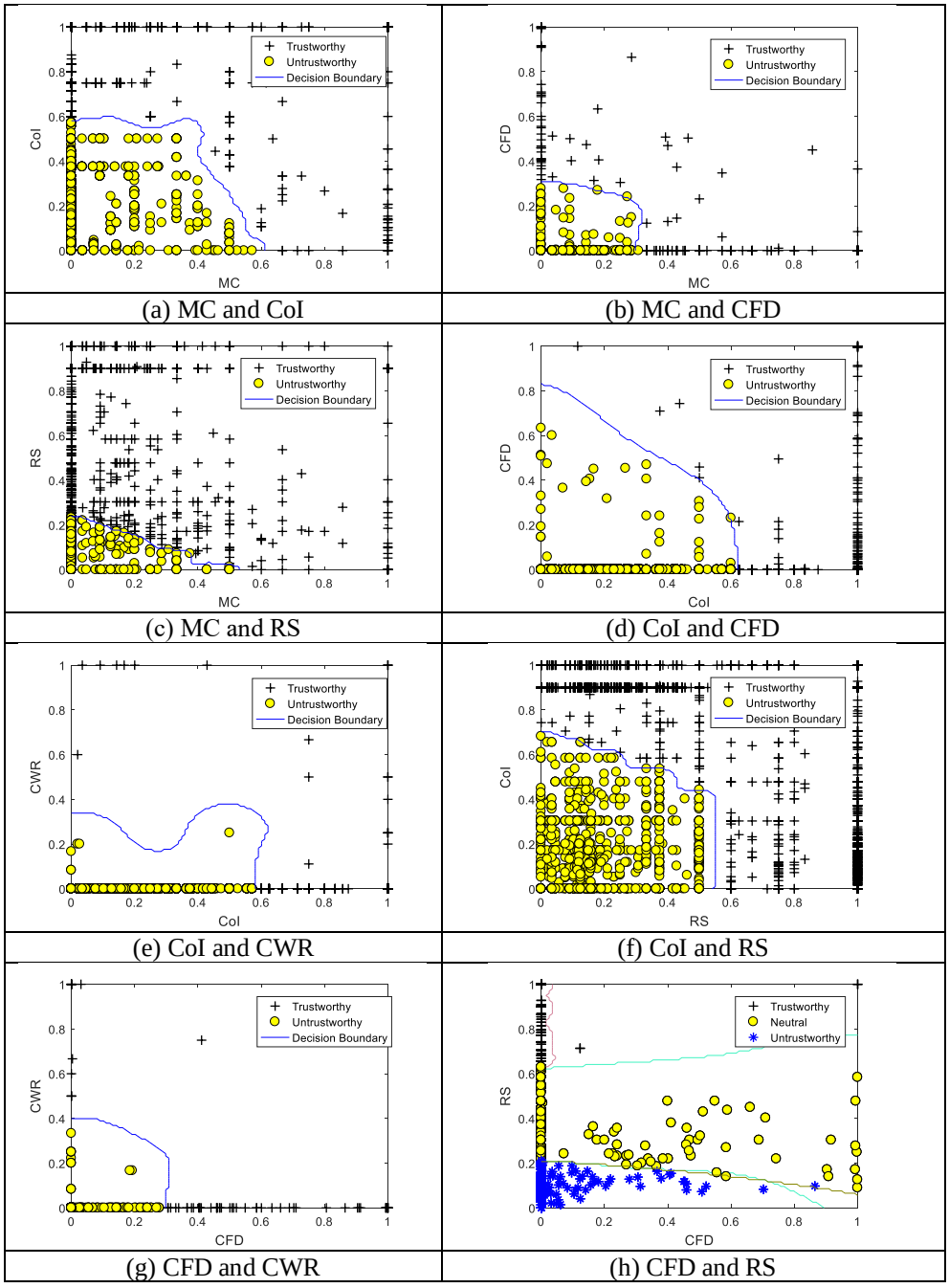
Figure 6-4: Elbow method: To decide the optimum number of clusters-*k*.

6.3.3. **Algorithm II: Classification Model**

Having investigated which interactions belong to the trustworthy region, we have used this information to label the data set. As an example, let us consider the same case in Figure 6-3(a). The points around the cluster centroid of the untrustworthy region are labeled as untrustworthy or "0" in the label vector "y", whereas the points outside the untrustworthy centroid are labeled as trustworthy or "1".

Then, with the labeled data, we train a model that can clearly identify whether incoming interactions are trustworthy. To estimate the optimum boundary, it is important to calculate the best regularization parameters "*C*" and "*gamma*" for each scenario mentioned above to avoid the data overfitting. For that, we have used part of the training samples as a cross-validation set and the results obtained via the trained model are shown in Figure 6-5, which clearly illustrates the decision boundary between the trustworthy and untrustworthy regions.

Furthermore, Figure 6-5(j) shows the result after applying the dimensionality reduction for all five features. For instance, let us consider Figure 6-5(a) in which the CoI and MC are in consideration. Now it is a matter of applying this model to the new data stream to distinguish which interactions fall into the trustworthy region and vice versa without any weight or threshold calculation. This not only reduces the calculation complexity and redundant work but also saves processing time.



(a) MC and CoI

(b) MC and CFD

(c) MC and RS

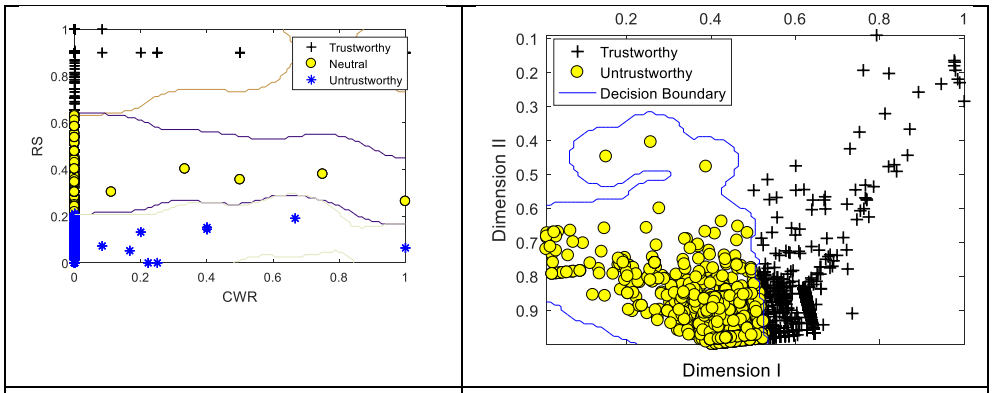
(d) CoI and CFD

(e) CoI and CWR

(f) CoI and RS

(g) CFD and CWR

(h) CFD and RS



(i) CWR and RS | (j) Features obtained via PCA

Figure 6-5: Application of Algorithm II on different pairs of features.

6.3.4. **Performance Analysis**

With these proven results, it is evident now that the system does not need to rely on conventional weighting factors and thresholds to decide the region of trustworthiness. Furthermore, the reduced complexity of the algorithms makes the system more lightweight while producing efficient results. However, the main assumption of this research is the centralized nature of the trust management platform. Particularly, we assume that every object under consideration is subscribed to a centralized database for publishing its data so that the trust management platform can access the data, train a model, and publish the trust values back into the data repository, which can be used by trustors. However, with the availability of powerful data centers, this issue of storing and computing can be solved without much difficulty to allow end devices to work efficiently without heavy burdens.

To demonstrate the effectiveness of our proposed method over the most common methods like the linear aggregation of TAs, a confusion matrix method is considered. Classification accuracy often gives misleading results and hides the details needed to diagnose the performance of the model especially when the number of observations in each class varies as in the dataset. On the other hand, the confusion matrix shows at which point the algorithm makes errors and importantly the types of errors made, which is critical for the investigation of algorithm applicability over expected results.

As a comparison, we consider the linear algorithms described in [77] and [136] and a nonlinear algorithm based on multiple regression described in Chapter 5. The obtained results are shown in Table 6-1. In the matrix, trustworthiness is considered as true positive conditions and untrustworthiness as true negative conditions. Furthermore,

predicted trustworthiness and untrustworthiness metrics are considered as predicted positive conditions and predicted negative conditions respectively. Based on the results from Table 6-1, parameters that define the performance of the two algorithms are shown in Table 6-2.

In classification, *Recall* gives an important insight about classification performance relative to the number of wrong predictions. According to results, the proposed algorithm shows 100% *Recall* or True Positive Rate (TPR) compared to 98.13% by the linear methods. As the data set is relatively small, a 1.87% performance improvement in the proposed algorithm will be very critical in real-world application deployment where billions of transactions happen in each second. This is again confirmed by the false negative rate (FNR) where the proposed algorithm shows 0% false negative predictions in comparison with 1.8% false predictions by liner methods. Note that TPR is similar in both proposed and nonlinear methods, as the nonlinear method only replaces the second part of the proposed algorithm. But, the proposed method outperforms the nonlinear approach as it gives a lower false positive rate (FPR) and a higher true negative rate (TNR) in contrast to the multiple regression, indicating that the proposed method shows compelling performance against untrustworthy objects.

Table 6-1: Algorithm Comparison with Confusion Matrix.

		Trustworthy	**Untrustworthy**
Trustworthiness Prediction	Proposed	105	12
	Linear	105	2
	Nonlinear	105	19
Untrustworthiness Prediction	Proposed	0	2862
	Linear	2	2874
	Nonlinear	0	2855

Table 6-2: Parameters Derived from Confusion Matrix.

	TPR/Recall	**FPR**	**Precision**
Proposed	1	0.004175	0.897436
Linear	0.981308	0.000695	0.981308
Nonlinear	1	0.006611	0.846774
Proposed	0	0.995825	
Linear	0.018692	0.999305	
Nonlinear	0	0.993389	
	FNR	**TNR**	

Further, there are infinite possibilities when aggregating multiple TAs using a linear weighted summation method. However, in this comparison, the same weighting factors given by the cauterization algorithm are used in the liner algorithm to calculate the final score. Due to this reason, both proposed and multiple regression methods give a comparatively low score in contrast to the linear method. However, in a realistic case, it is difficult to estimate these weighting factors without a proper cauterization algorithm as discussed in this work and hence precision will severely degrade compared to our proposed method. On the other hand, the regularization factor used to manage the overfitted data and the optimization algorithm used to find the optimum parameters for the features could have a significant effect on this cause. Thus, the precision of both models can be increased by observing the learning curve while tweaking this regularization factor depending on the data set and using advanced methods of optimization as described in [142], [143].

Moreover, the algorithms described in this paper can be clustered so that the end devices can perform a fraction of the analysis and obtain the same results as before. To establish a distributed platform and address scalability issues, methods like map-reduction and data parallelism are strong candidate technologies. This is quite beneficial in an environment like IoT where scalability and collaboration are prominent factors.

6.4. Discussion

The ML algorithm discussed here basically propose following contributions; (i) Taking both social and non-social features in to consideration to train the ML model; (ii) TA aggregation technique in contrast to linear summation; (iii) Labeling interactions based on their trustworthiness; and (iv) Usage of real-world dataset to investigate the performance of the algorithm in contrast to synthesized data.

Authors in [144], [98] and [145] investigate more innovative models and solutions for preserving privacy, security and data integrity based on statistical and deep learning concepts. Moreover, authors in [146] and [101] propose a regression-based model which compares the variation of trustworthiness with respect to trust features in mobile ad-hoc networks (MANET) and WSN. However, they have investigated a limited number of trust features, which only represent the system level information like packet forwarding ratio, Quality of Service, energy-sensitivity, capability-limitation, and

profit-awareness. This motivates us to present a generic algorithm that represents features from both social level as well as system level data as in 1st contribution.

Recently, authors in [147], [103] and [104] present several algorithms based on reinforcement learning and multiclass classification techniques which lay the foundation for the algorithms considered in this work especially in relation to the 3rd contribution. Even though these research achievements show some prominent results by applying ML techniques, they still lack the potential of being a generic algorithm in relation to 1st, 2nd, and 4th contributions.

6.5. Chapter summary

A novel algorithm is proposed in this chapter as opposed to traditional weighted summations to determine whether an incoming interaction is trustworthy, based on several trust features corresponding to an IoT environment. First, a method for labeling the data depending on their trustworthiness is realized based on unsupervised ML techniques, which is the vital first step for any system to identify which interactions are trustworthy. Following this labeling process, a trust prediction model, which can correctly identify the trust boundaries of any interactions and learn the best parameters to combine each TA to obtain a final trust value, is proposed based on the SVM model. Our simulation results have shown promising outcomes including the ability and accuracy of the algorithm with respect to identifying trustworthiness interactions.

CHAPTER 7: DATA TRUST EVALUATION

7.1. Introduction

The most common method of assessing trust in IoT applications is to estimate the trust level of the end objects (object-centric) relative to the trustor. In these systems, the trust level of the data is assumed to be the same as the trust level of the data source. However, most of the IoT based systems are bassed on multiple data streams and operate in dynamic environments, which need immediate actions without waiting for a trust report from end objects. Furthermore, having trustworthy source objects is not always prominent but the trustworthy data. As an example, reliable, up to date and location, sensitive information about weather, traffic, safety warnings and transport information from a smart city application are more important than the facts about objects who are actually generating them. The other common misinterpretation is that the assumption of having OT would guarantee DT, which is in fact significantly different in various aspects such as the validity of data, timeliness, and other properties unique to data, which are often ignored in calculating trust for end objects.

Nevertheless, information is the governing factor for any IoT system and is generated from the data by combining it (data) with the context. Hence, if there is a Data Quality (DQ) problem, it would eventually lead to an Information Quality (IQ) problem [22]. In other words, once the right data item is delivered to the desired object at the precise time in a clear, usable, and meaningful manner, IQ is guaranteed. Therefore, we address this challenge in this chapter by extending our previous approach on trust establishment for objects based on their reputation, experience, and knowledge, as described in Chapter 2.

7.2. Data Trust Evaluation

To the present day, evaluation of trust in data is assumed to be identical to trust estimation of end objects. However, this is not entirely true and in fact, most IoT systems rely highly on several data streams and these systems often do not care about the integrity and quality of who is generating them but the integrity of the data itself. For an example, let’s consider a situation where it is impossible to obtain the required factors to access the trustworthiness of objects, for example, a cloud service. In this case, cloud service user (Trustor) has no way to access the trust between him and the

cloud server (Trustee) as often information of the servers is not visible to its' customers. However, it is unquestionable that user needs to find good cloud service that matches with his requirements in terms of trustworthiness in addition to performance metrics. In such a case, the proposed model provides an alternative way to evaluate trust of the data, coming from the server without worrying about the factors that determine the trustworthiness of the object itself (i.e. cloud server). Another example is where the interactions happened for a short duration without any prior relationship with the trustee. In such situations, it can be disadvantageous to calculate trust between objects due to time-critical nature of the application. As an example, obtaining accurate information about certain accident situations from less trustworthy objects like taxi drivers and passengers are more important than waiting for a report from a police officer, who is considered more trustworthy than a taxi driver for requesting quick attention from medical authorities and other relevant parties.

To address these challenges, we propose a data trust evaluation model and the extended version of the trust management platform, which can analyze both data trust as well as object trust separately or in a collective manner.

7.2.1. **Data Trust Attributes**

Alongside the REK model, we first consider a separate set of TAs, which essentially define the properties of data. Many research papers on DQ show that the six parameters (e.g., completeness, uniqueness, timeliness, validity, accuracy and consistency) provide prominent insight for assessing the DQ matters as in [148-150]. With respect to the notion of trust, we can consider these properties as trustworthiness attributes. We also consider two additional attributes, "success" and "cost", which characterize experience and reputation DTM calculation, in addition to the aforementioned attributes stated in Chapter 4. We consider these eight DT Attributes (DTA) as the core dimensions in finding the trust between a data item and the trustor. Thus, we model these properties as below:

- Success (T_B^{su}): the probability that B will successfully execute the task
- Cost (T_B^{ct}) : the probability that the cost of executing the task by B is not more than expected

- Completeness (T_B^{cm}): the probability of complete data records over total data records
- Uniqueness (T_B^{uq}): the probability of expected records over total records noted
- Timeliness (T_B^{tm}): the difference between last update and the current one
- Validity (T_B^{vl}): the validity of data type, syntax and range
- Accuracy (T_B^{ac}): the probability of accurate data records over total data records
- Consistency (T_B^{cn}): the probability of valid, accurate and unique records over total data records

7.2.2. **Data Trust Evaluation Model**

In this section, we extend our object based trust model discussed in Chapter 3 to comply with the data based trust as shown in Figure 7-1 and explain how each DTA is combined to generate data trust as discussed in Section 3.3.3. . For that, we identify completeness, uniqueness, timeliness, validity, accuracy, and consistency as DTAs, which represent knowledge TM as it conveys trustworthiness information about the trustee. On the other hand, "success" DTA and "cost" DTA represent the experience DTM of the trustor after each task. Finally, reputation DTM can be considered by aggregating opinions of other trustees if there are any. Based on this, basic DT assessment towards B by A (T_{AB}^{x}) over *x* DTM can be numerically modelled as below:

- Knowledge DTM (T_{AB}^{K})

$$T_{AB}^{K} = \alpha T_B^{cm} + \beta T_B^{uq} + \gamma T_B^{tm} + \delta T_B^{vl} + \varepsilon T_B^{ac} + \epsilon T_B^{cn} \tag{7.1}$$

where α,β,γ,δ,∈, and ε are weighting factors such that α+β+γ+δ+∈+ε=1. However, calculating these weighting factors is computationally costly and not practical due to the infinite number of possibilities. Hence, the thesis suggest applying an ML technique to combine all the TAs, which we have discussed in our previous work [151].

- Experience DTM (T_{AB}^{E})

$$T_{AB}^{E} = \sigma T_B^{su} + \varphi \frac{1}{T_B^{ct}} \tag{7.2}$$

where σ and ϕ are weighting factors such that σ+ϕ =1 and $T_B^{ct} > 0$. The ML method discussed in [151] is preferable for the TA combination in this case as well.

– Reputation DTM (T_{AB}^{R})

$$T_{AB}^{R} = T_{1B}^{R} + T_{2B}^{R} + \cdots + T_{nB}^{R} \tag{7.3}$$

where T_{nB}^{R} represents the reputation towards data source B by its previous users *n*. A mechanism that computes reputation based on the PageRank algorithm is presented in our previous research [152].

After releasing the main DTMs, the next objective is to combine them in order to produce a final DT value (T_{AB}^{d}) for each data source based on DTAs as below:

$$T_{AB}^{d} = \rho T_{AB}^{K} + \tau T_{AB}^{E} + \omega T_{AB}^{R} \tag{7.4}$$

where ρ,τ, and ω are weighting factors based on the trustor's preference on each TM. In here, we suggest two mechanisms to combine each TM either based on the ML approach we followed in [151] or applying the rule based reasoning mechanism explained in [153].

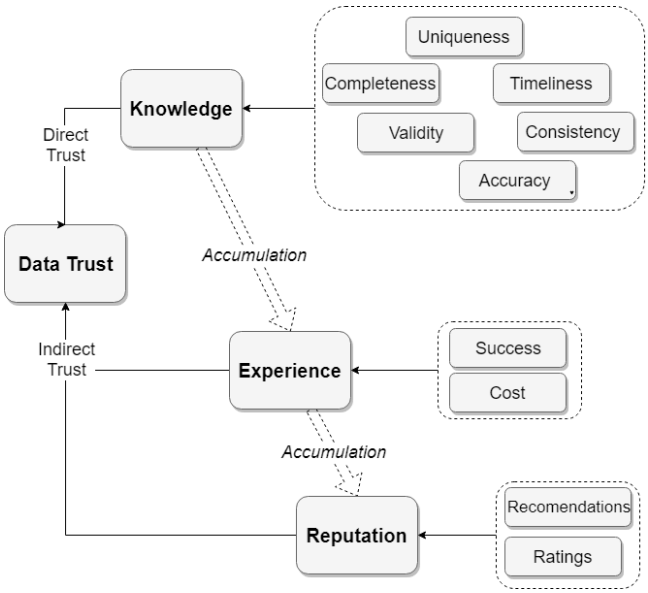


Figure 7-1: Data Trust Evaluation Model.

7.2.3. **Data Trust Management Platform**

The platform consists of several important modules such as Trust Computation, Prediction and Decision Making (TCPD), TAG, Trust Data Access Object (TrustDAO), Data Repository, Trust Computation, and decision-making module, Trust Service Enabler and API as shown in Figure 7-2. Once the TCPD identifies the requirement of the data, it asks the TAG via TrustDAO to collect necessary

information and pre-process it for trust evaluation. Then, this pre-processed data is stored in the DR to be used by other modules including external platforms through TCPD API.

Afterward, a TM extraction module estimates the necessary TAs based on the requirements. These attributes can be categorized either as attributes that define data trust as explained in Section 7.2.1. or attributes that define trust of object as described in Chapter 2. Next, all the attributes are combined based on the REK model with the assistance of the trust computation module, which is capable of performing the calculation based on either numerical methods or an artificial intelligence approach. Finally, the decision-making and delegation module uses the predicted trust values in order to complete the decision process, perhaps with the support of a service enabler who actually performs the judgment made by the decision module. In the following sections, we explain the data based TA estimation, DT computation, and DT prediction in detail.

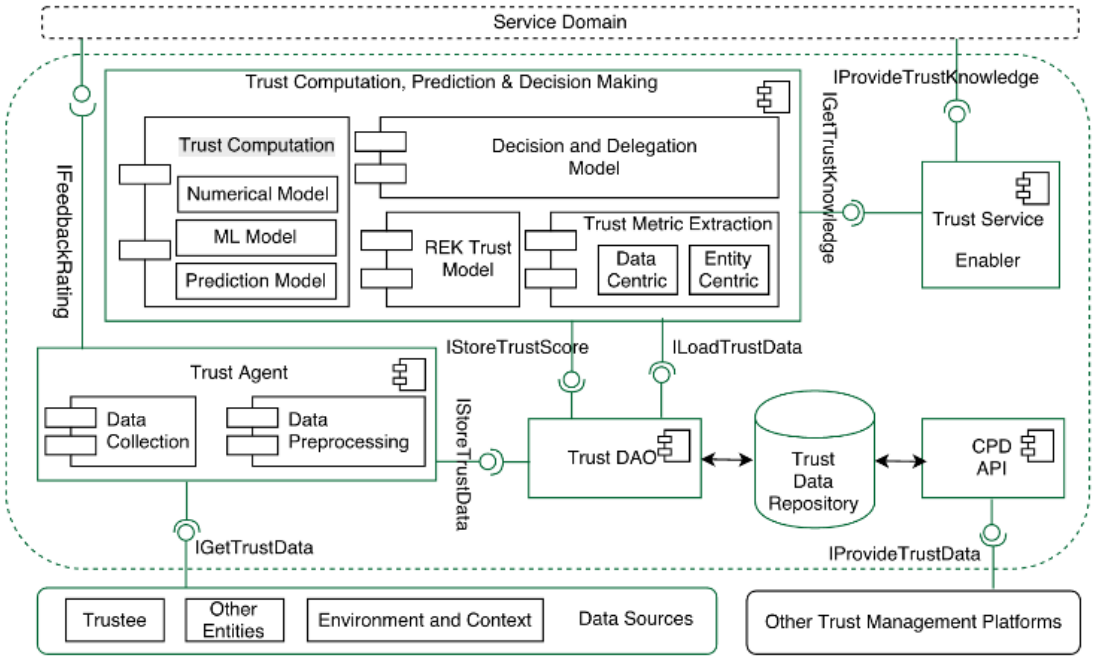


Figure 7-2: Data trust management platform.

7.2.4. **Data Trust Prediction Algorithm**

Once the trust values based on DTA are collected, the next step is to find the trust relationship among data sources and the trustors who have not had prior encounters. For that, we use the concepts of the CF technique to predict the unknown trust values between the user and specific data source, with respect to six different data-centric

features (e.g., completeness, uniqueness, timeliness, validity, accuracy, and consistency). Now the prediction is solely based on the properties of data, it is unnecessary to rely on the trustworthiness of the data source as in traditional methods anymore. Among various methods of recommendation techniques, we chose a variant of the multifaceted CF model for our application, due to its unique properties that match with our DT model. Such properties include an emphasis on the concept of social contribution (where everyone's contribution matters), the capability to capture weak signals in the overall data, ability to detect strong relationships between close items and competence in avoiding overfitting [154].

Table 7-1: The Input Matrix of Users × Items × Features for the CF Algorithm.

	Trustors (Users)				Features					
Trustees (DS)	u_1	u_2	…	u_{n_u}	T^{cm}	T^{uq}	T^{tm}	T^{vl}	T^{ac}	T^{cn}
i_1	◬		◬		◈	◈	◈	◈	◈	◈
⋮		◬		◬	◈	◈	◈	◈	◈	◈
j_{nm}		◬	◬		◈	◈	◈	◈	◈	◈

First, we define the inputs to our algorithm as the number of trustors or users (n_u), number of Trustees or DSs (n_m) and the six features, as shown in Table 7-1. Users who already have a trust relationship with DSs are noted with "◬" symbol which actually represents some trust value between [0,1], calculated using equation (7.4) and the blank spaces denote the missing information, which is to be predicted. Formally, if user j and item i already have a trust relationship, then r(i,j)=1 otherwise r(i,j)=0, otherwise. Moreover, the DT value given by user j to DS i is denoted by $y^{(i,j)}$. The symbol "◈" represents the values of all six features in between 0 and 1.

The next step of our algorithm is to find a parameter that describes the profile of users involved in a certain situation. For now, let's assume this parameter is denoted by $\boldsymbol{\theta}^{(j)}$ for a particular user j and the feature vector for DS i is denoted by $\mathbf{T}^{(i)}$. Then the predicted DT value T^{dp}_{ij} between the trustor and the data can be calculated as shown in equation (7.5). The symbol $(.)^T$ represents the transposition of the vector.

$$T_{ij}^{dp} = \left(\boldsymbol{\theta}^{(j)}\right)^T (\boldsymbol{T}^{(i)}) \tag{7.5}$$

The basic but essential requirement of the predicted trust value is that it must provide the closest possible prediction for each trust value that is already calculated by each user. Based on this assumption, we can use the Mean Square Error (MSE) method to find the distance between the actual trust values and predicted ones. The parameter $\theta^{(j)}$ which gives minimum error would be our best-predicted trust value. This idea is formulated as below for trustor *j*:

$$\min_{\theta^{(j)}} \frac{1}{2}\sum_{i:r(i,j)=1}\left(\left(\boldsymbol{\theta}^{(j)}\right)^T\left(\boldsymbol{T}^{(i)}\right) - y^{(i,j)}\right)^2 + \frac{\lambda}{2}\sum_{k=1}^{6}(\theta_k^{(j)})^2 \tag{7.6}$$

In the first part of the equation, the MSE is calculated over all the records, where the trust value is already available through preliminary calculation. The second part of the equation is used to regularize the minimization process thereby avoiding overfitting issues. The *k* denotes the number of features. Similarly, we can find the best parameter for each trustor as shown below:

$$\min_{\boldsymbol{\theta}^{(1)},\boldsymbol{\theta}^{(2)},\ldots,\boldsymbol{\theta}^{(n_u)}} J(\boldsymbol{\theta}^{(1)},\boldsymbol{\theta}^{(2)},\ldots,\boldsymbol{\theta}^{(n_u)}) \tag{7.7}$$

where J(.) denotes the cost function as described in equation (7.6). In order to minimize the cost function, this work simply adopt the gradient descent method and solve for the best parameter $\theta^{(j)}_k$ as shown below [155]:

$$\theta_k^{(j)} = \begin{cases} \theta_k^{(j)} - \alpha \sum\limits_{i:r(i,j)=1}\left(\left(\boldsymbol{\theta}^{(j)}\right)^T\left(\boldsymbol{T}^{(i)}\right) - y^{(i,j)}\right)^2 T_k^{(i)}, k = 0 \\ \theta_k^{(j)} - \alpha\left(\sum\limits_{i:r(i,j)=1}\left(\left(\boldsymbol{\theta}^{(j)}\right)^T\left(\boldsymbol{T}^{(i)}\right) - y^{(i,j)}\right)^2 T_k^{(i)} + \lambda\theta_k^{(j)}\right), k \neq 0 \end{cases} \tag{7.8}$$

Once the parameter $\theta^{(j)}$ is estimated through equation (7.7), and (7.8) predicted trust value between user *j* and item *i* will be given by the equation (7.5). Please note that this process is an iterative process and that more users who have experience with similar DSs would make the system more accurate and trustworthy.

7.3. Implementation of Trust Model

In this section, we propose a possible implementation scenario of our findings based on an air pollution crowd sensing use case, aimed at collecting and monitoring pollution data. The air pollution sensing requires active citizen participation by carrying wearable sensors as they traverse the city based on an opportunistic crowd

sensing application [156]. However, monitoring such air pollution via crowd sensing requires the data provided to be trustworthy and relied upon by the city authority or government to make an immediate decision. The air pollution crowd sensing application will take advantage of citizens' smartphones and smart cities' air pollution/environment sensors. The data collected from the air pollution sensors are delivered to the IoT Cloud, hosting the TCPD proposed in this paper. Thus, a mobile app for trusted air quality data monitoring can be developed on top of this platform integrating data collected from low-cost environment sensors for temperature, humidity, CO, CO2 NO2, SO2, as well as compounds including benzene and lead (VOCs), etc. The sensors' readings will be transmitted via either an Android or iOS app to the proposed system for assessing and predicting the trust of the data before it is sent to the IoT Cloud. Such data can then be visualized along with its trust level by interested individuals, government, city administrators etc. via a web application.

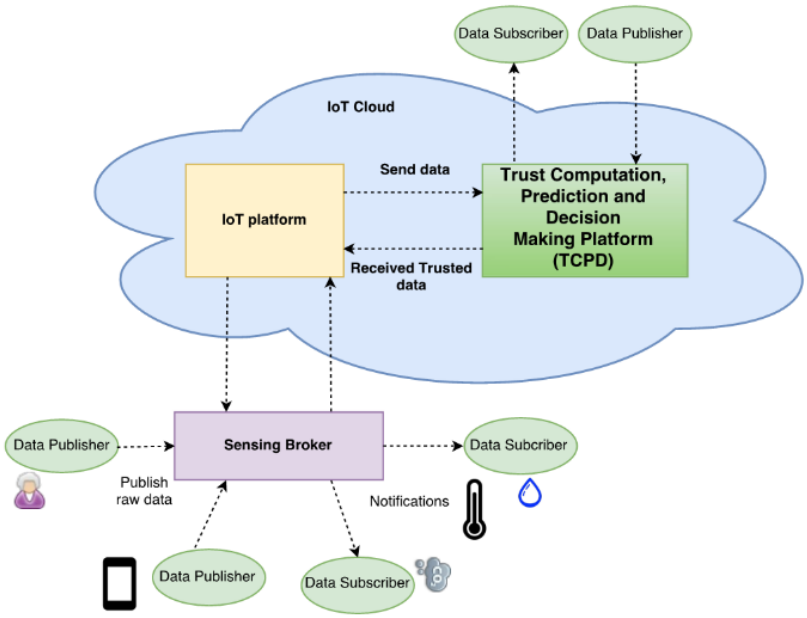


Figure 7-3: High-level Implementation Architecture of the proposed System.

For the above use case to profit from the proposed solution, we have proposed a distributed publish-subscribe architecture such as CoreDX distributed publish-subscribe middleware [157] whereby interested parties can subscribe via a broker to environmental data of interest in a specific location of their choice as illustrated in Figure 7-3. The TCPD section of the figure implements appropriate components of the platform as shown in Figure 7-2, for providing trusted data to the interested parties. This is a typical publish-subscribe system whereby publishers publish the sensor data

to the broker and subscribers receive notifications matching their subscriptions from the broker. As illustrated in Figure 7-2, the TCPD can communicate with the IoT platform via an edge server that implements the IGetTrustData and IProvideTrustData interfaces. Also, the TCPD can receive data from the IoT platform for predicting the trust of such received data.

Finally, Figure 7-4 illustrates an example of a scaled down sequence of interactions between some important stakeholders of an implementation instance of the system. Anytime a new environment sensor is available, it registers its presence with the sensing broker, which in turn informs the platform of the new available sensor. The new sensor can then publish its data to the broker. The broker notifies the TCPD to predict the trust of the received data. Similarly, whenever a new subscriber joins the system, its subscription is submitted to the broker via the TCPD system. If a subscription matching at least one of the subscriptions of the new subscriber is available, the broker notifies the TCPD system to deliver the data to the subscriber along with the trust level of the data.

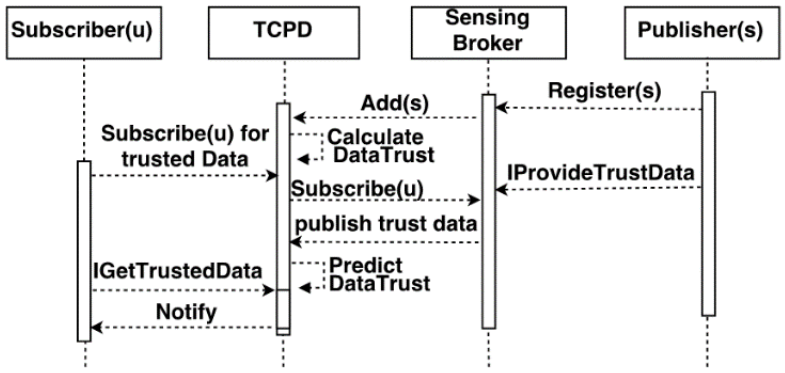


Figure 7-4: Interactions between various stakeholders and the proposed platform.

7.4. Discussion

Presently, data is the key governing factor with respect to service provisioning and decision-making process in IoT. Hence, the assurance of DQ and IQ are utmost important for trustworthy interactions. In this regard, authors in [149], [158] and [159] discuss various techniques and metrics that can be considered for DQ and IQ measurement. The framework proposed by Askham et al. [148] is one of the most prominent and widely accepted model for DQ assessment due to its generic nature. Hence, this thesis adopts most of the concepts from this work even though it doesn't explain how these attributes can be used to assess data trust as proposed by this thesis.

Moreover, authors in [160] and [161] argue a data-centric trust model for vehicular networks based on several techniques like Bayesian inference and Dempster-Shafer theory. However, these mechanisms do not model trust explicitly but use cross-checking mechanisms to check the trustworthiness of the data coming from various sources. In contrast the proposed model use metrics from three dimensions according to REK model in order to evaluate data trust and hence it does not need to rely on the number of sources which send the same information.

In contrast to traditional means, there are quite a few works on trust prediction based on collective methods where numerically assessed trust metrics are analyzed through an intelligent algorithm like supervised and unsupervised learning. In this regard, a model to improve trust prediction accuracy by combining user similarity rating and the traditional trust is proposed in [162] and [163]. Furthermore, Xiang et al. propose a model based on unsupervised learning algorithm to estimate relationship strength from interaction activities like tagging, communication, and interference [164]. However, none of these have addressed the issue of evaluating trust when there are no previous encounters between trustor and trustee. However, the proposed method in this thesis uses the ability of collaborative systems to estimate unknown relationships through the known relationships.

7.5. Chapter summary

In this work, we argue that the traditional means of trust evaluation for objects does not necessarily guarantee the trustworthiness of data that they generate. Hence, we propose a hybrid trust computational platform, which is capable of assessing both data trust as well as traditional object-based trust. Furthermore, we provide a model to compute individual DTA and the main DTM by combining numerical models with ML algorithms. Afterward, a DT prediction scheme based on CF is proposed to find the DT between trustors and data sources who do not have prior encounters, which avoids using data from potentially malicious actors. Finally, a possible implementation scenario is discussed based on a crowd sensing use case. Similarly, our algorithm would be beneficial to filter out malicious data and data sources to maintain the integrity and quality of the outcomes that any crowd sensing application produces.

CHAPTER 8: CONCLUSION AND FUTURE WORK

8.1. Conclusion

To the present day, existing approaches for trust evaluation are quite specific to the cyber-physical domain and show a weak performance against SIoT models. To the best of our knowledge, there is no model available, which is implemented in the platform layer and utilizes TMs from all other layers in the IoT stack. Therefore, it is important to investigate trust realization methods when dealing with billions of service requests and to effectively mitigate overlay threats including self-promoting, good mouthing/bad-mouthing and other possible vulnerabilities in CPSS. While defending from these threats, it is also important to investigate resilient and autonomous approaches to enhance the trustworthiness among objects and improve the service provisioning capabilities. As a solution, this research proposes a trust management platform that cooperates with applications and services to automatically evaluate all aspects of trust among any objects in the IoT environment and to realize trustworthy services and experiences.

First, a generalized explanation for trust in the IoT ecosystem is formularized, which can be used as the baseline definition for any system, avoiding any ambiguities caused by existing definitions on trust. However, to use trust as an input in the decision-making process, it should be modeled in a quantifiable manner. This is another area where existing solutions struggle and often the solutions are biased towards specific application areas. To overcome this issue, a generic trust evaluation model is proposed that can be used to obtain the trust profile of any system based on three TMs: knowledge, experience, and reputation. These represent direct information which is obtained after having at least one interaction with the trustee, to calculate trust; and opinions collected from global objects, who had previous encounters with the trustees respectively.

It is also important to figure out how this model can be used to evaluate trust using these three TMs. Therefore, a novel trust management platform is proposed, which identifies the must-have components and modules in such a platform. The functions, usability, and the place of each of these modules is then defined so that this platform can be used as a standard for future developments. Furthermore, it can be argued that

the traditional means of trust evaluation for objects does not necessarily guarantee the trustworthiness of data that they generate. Hence, a hybrid trust management platform, which is capable of assessing both data trust as well as traditional object-based trust, is proposed by extending our trust platform for objects towards data.

Based on the above grounds, specific solutions are proposed according to the defined trust evaluation model to acquire trust using the three TMs. First, a novel algorithm, which analyzes the opinions and experiences based on reputations and recommendations, is proposed. Then an algorithm that models direct trust information is designed, which combines multiple numerical models together. Each of these numerical models represents a certain feature of trust; for example, reputation, experience, relationship, spatial distance, credibility, and consistency. Depending on the requirement, each of these features is further broken down into sub-TAs in the process of numerical modeling.

However, there is very limited existing work on combining such models together to represent the final value of trust. Often, the existing proposals suggest the use of linear methods with predefined weighting factors, which essentially do not represent the true nature of the requirements of prospective services, who expect to invest in trust values. Thus, there is an urgent need for more effective trust aggregation methods, which might be based on intelligent ML techniques as discussed before. In this regard, several intelligent approaches are presented based on unsupervised, supervised, and CF algorithms to label incoming interactions and predict future relationships based on their trustworthiness.

From a standardization point of view, until now, a number of standards focusing on network-security and cyber-security technologies have been developed in various standardization bodies including IETF and ITU. The scope of these standards needs to be expanded to take into consideration trust issues in future ICT infrastructures. There are few preliminary activities taking place, for instance in Online Trust Alliance (OTA) and Trusted Computing Group (TCG). However, existing standardization activities on trust are still limited.

Hence, this thesis has provided several inputs including a computational definition for trust, a trust evaluation model targeting both object and data trust, and platform to manage trust evaluation process under ITU-T SG 13 [165]. Further, the thesis has

contributed to developing a technical report containing definitions, use cases, functional classifications as well as challenges and technical issues related to trust including overall strategies of standardization for trust provisioning. As the lead group of trusted networking infrastructure, ITU-T SG13 successfully published the recommendation Y.3052 on trust in March 2017 [119]. Recently Question 16/13 "*Knowledge-centric trustworthy networking and services*" has focused on basic issues and key features on trust. Q16/13 is now mainly focusing on the development of core technical solutions for trust provisioning for ICT infrastructures and services. Q16/13 also plans to consider technology deployment as well as new services and business aspects on trust-based networks and eco-platforms.

Our proposals provided a strong suggestion to improve the current standardization activities on trust in ITU-T SG13 towards a hybrid model based on the concepts of OT and DT. Among them, a trust relationship model described in this work elaborates on some important factors when it comes to trust based decision making, which is a vital part of the standardization process. On the other hand, trust evaluation via ensemble methods, which is by combining numerical, ML and recommendation algorithms, provides a robust perception about trust compared to traditional one-dimensional trust calculation techniques.

In summary, the novel contributions presented in this thesis are outlined below.

- Development of a trust evaluation model and a trust management platform that describes a complete trust provisioning lifecycle.
- A computational model to estimate indirect trust attributes.
- A computational model to estimate direct trust attributes.
- A machine learning based trust assessment, aggregation, and prediction model.
- An extended version of the trust evaluation model and the trust management platform towards evaluating the trust of data.
- A trust prediction model for data trust based on collaborative filtering.
- An implementation scenario of the trust platform based on the crowd sensing use case in a distributed environment.

8.2. Future work

The work contained in this thesis provides a novel and promising approach on trust evalaution in IoT systems. However, there are a number of ways that this work could be further improved and address new challenges accordingly to enhance the performance of the discussed platform. Hence, an overview of such possible extensions and considerations is presented here.

Our research plan includes designing an autonomous trust decision-making algorithm for dynamically changing IoT environments. In this regard, different decision mechanisms can be observed in the literature with different techniques. A promising approach to handle such dynamics is self-adaptation that can be realized by a MAPE-K feedback loop as shown in Figure 8-1 [166]. There are many such techniques that can be observed in the literature as in [167]. Based on these concepts, it is vital to investigate such systems' capabilities to generate decisions and manage the platform adaptively according to dynamic requirements of the ecosystem.

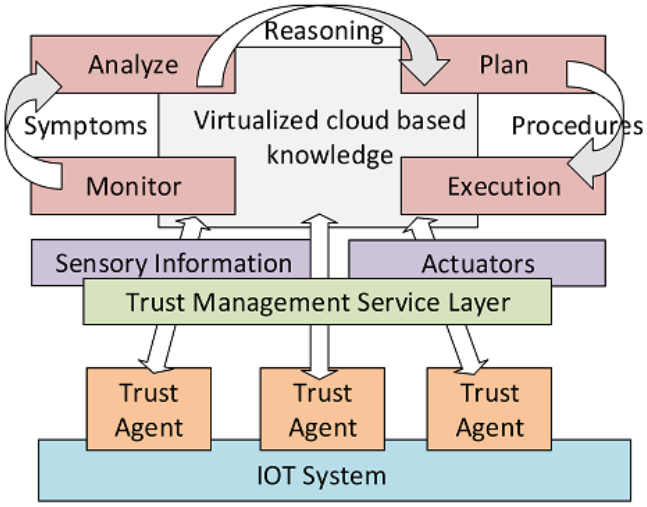


Figure 8-1: MAPE-K feedback loops for adaptive TAGs.

To support these characteristics, autonomics through feedback loop control under dynamic conditions is required and recent advances like fog computing or edge computing can be a possible solution for distributed and localized trust-based decision-making. Thus, the IoT platforms should be equipped with tools that allow intelligent services to be composed as data-driven microservices. The rationale is to address the weaknesses of the current monolithic Cloud-based AI services, which cannot meet the requirements of real-time and ultra-low delay sensitive IoT applications. Rather than shipping the trust data to the cloud data centers where AI algorithms are applied to

incorporate intelligent decision-making capabilities into IoT applications. These AI algorithms can be implemented and deployed closer to the sources of the IoT Data and users, by factoring the AI functionality into smaller functions that can be implemented as distributed microservices as shown in Figure 8-2.

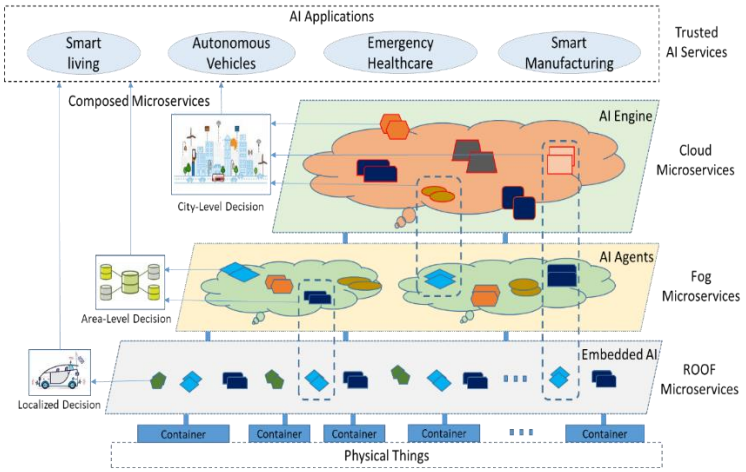


Figure 8-2 : Distributed AI Architecture for Trusted Services.

One of the ultimate objectives of trust provision is that IoT systems should organize themselves, under the constraints and guidance of external inputs, to meet the goals of their participants. In this regard, the idea of a Knowledge Plane (KP) must be considered in parallel to management and data planes, to supervise network dynamics, participants, and relationships between the objects, and thereby manage and connect information within the system in each event or activity. With the above properties, our target is to design a distributed KP architecture providing necessary functionalities to gather information from reasoning mechanisms, TAGs, and brokers while providing its service to trust management processes. Alternatively, a KP may support the acquisition of high-level goals, understand current system conditions, storage and propagation of knowledge information among trusted parties, identify constraints and finally assist the decision-making process perceptively.

As compared to network security, it is essential to investigate whether trust validation methods can effectively defend against different attacks including self-promoting and good mouthing/bad mouthing attacks. While defending from attacks, it is also important to investigate resilient self-healing approaches to enhance trust recovery after a successful attack. Furthermore, the effectiveness of trust management solutions when it comes to billions of devices and applications should be studied carefully. One

possible direction is to investigate trust management with concepts like big data and RL. Essentially employing trust capabilities should minimally compromise the performance and process of the IoT as many devices have limited resources. A possible research direction is the investigation of intelligent trust-based routing protocols, which are more reliable while consuming minimum energy and reducing traffic overhead.

REFERENCES

[1] F. Y. Wang, "The Emergence of Intelligent Enterprises: From CPS to CPSS," *IEEE Intelligent Systems,* vol. 25, no. 4, pp. 85-88, 2010.

[2] L. Atzori, A. Iera, G. Morabito, and M. Nitti, "The Social Internet of Things (SIoT) – When social networks meet the Internet of Things: Concept, architecture and network characterization," *Computer Networks,* vol. 56, no. 16, pp. 3594-3608, 2012.

[3] A. Sheth, P. Anantharam, and C. Henson, "Physical-Cyber-Social Computing: An Early 21st Century Approach," *IEEE Intelligent Systems,* vol. 28, no. 1, pp. 78-82, 2013.

[4] P. Bonatti, and P. Samarati., "Regulating service access and information release on the web," in *Proceedings of the 7th ACM conference on Computer and communications security,* Athens, Greece, 2000, pp. 134-143.

[5] N. Li, and J. Mitchell, "RT: A Role-based Trust-management Framework," in *DARPA Information Survivability Conference and Exposition (DISCEX)*, Washington D.C, 2003, pp. 201-212.

[6] R. Gavriloaie, W. Nejdl, D. Olmedilla, K. E. Seamons, and M.Winslett, "No Registration Needed: How to use declarative policies and negotiation to access sensitive resources on the semantic web," in *1st European Semantic Web Symposium (ESWS)*, Berlin, Heidelberg., 2004, pp. 342-356.

[7] P. A. Bonatti, and D. Olmedilla, "Driving and monitoring provisional trust negotiation with metapolicies," in *IEEE 6th International Workshop on Policies for Distributed Systems and Networks (POLICY)*, Stockholm, Sweden, 2005, pp. 14-23.

[8] M. Winslett, K. E. S. T. Yu, A. Hess, J. Jacobson, B. S. R. Jarvis, and L. Yu, "Negotiating trust on the web," *IEEE Internet Computing,* vol. 6, no. 6, pp. 30-37, 2002.

[9] J. Kohl, and B. C. Neuman, "The Kerberos network authentication service," *IETF RFC 1510,* 1993.

[10] W.H.Winsborough, K. E. Seamons, and V. E. Jones, "Automated trust negotiation," in *Proceedings of the DARPA Information Survivability Conference,* Hilton Head, SC, USA,, 2000, pp. 88-102.

[11] M. W. T. Yu, "Policy migration for sensitive credentials in trust negotiation," in *Proceedings of the 2003 ACM Workshop on Privacy in the Electronic Society (WPES 03)*, New York, USA, 2003, pp. 9-20.

[12] M. W. T. Yu, K.E. Seamons, "Interoperable Strategies in Automated Trust Negotation," in *Proceedings of the 8th ACM Conference on Computer and Communications Security*, New York, USA, 2001, pp. 146–155.

[13] W. Nejdl, D. Olmedilla, and M. Winslett, "Peertrust: automated trust negotiation for peers on the semantic web,," in *Workshop on Secure Data Management*, Berlin, Heidelberg, 2004, pp. 118–132.

[14] N. Li, W. H. Winsborough, and J. C. Mitchell, "Distributed credential chain discovery in trust management," *Journal of Computer Security,* vol. 11, no. 1, pp. 35-86, 2003.

[15] C. D. J. J.-M. Seigneur, "Trust enhanced ubiquitous payment without too much privacy loss," in *Proceedings of the 2004 ACM Symposium on Applied Computing*, New York, USA, 2004, pp. 1593–1599.

[16] N. M. S. F.L. Gandon, “Semantic web technologies to reconcile privacy and context awareness,” in *UbiMob ’04: Proceedings of the 1st French-speaking Conference on Mobility and Ubiquity Computing*, New York, USA, 2004, pp. 123–130.

[17] A. Uszok, J. Bradshaw, R. Jeffers, N. Suri, P. Hayes, M. Breedy, L. Bunch, M. Johnson, S.Kulkarni, and J. Lott, “KAoS policy and domain services: toward a description-logic approach to policy representation, deconfliction, and enforcement policy,” in *IEEE 4th International Workshop on Policies for Distributed Systems and Networks*, Lake Como, Italy, 2003, pp. 93-96.

[18] L. Kagal, T. W. Finin, and A. Joshi, “A policy-based approach to security for Semantic Web,” in *Proceedings of the 2nd International Semantic Web Conference*, Berlin, Heidelberg, 2003, pp. 402–418.

[19] M. Nielsen, and K. Krukow, “Towards a formal notion of trust,” in *Proceedings of the 5th International Conference on Principles and Practice of Declaritive Programming (ACM SIG PLAN)*, Uppsala, Sweden, 2003, pp. 4-7.

[20] OASIS, "WS-Trust 1.4," *Available : http://docs.oasis-open.org/ws-sx/ws-trust /v1.4/ws-trust.html*, 2012.

[21] M. Carbone, M. Nielsen, and V. Sassone, “A formal model for trust in dynamic networks,” in *Proceedings of 1st International Conference on Software Engineering and Formal Methods*, Queensland, Australia, 2003, pp. 54-61.

[22] M. Y. Becker, and P. Sewell, “Distributed access control policies with tunable expressiveness,” in *Proceedings of 5th IEEE International Workshop Policies for Distributed Systems and Networks*, orktown Heights, NY, USA, 2004, pp. 159-168.

[23] T. Leithead, W. Nejdl, D. Olmedilla, K. E. Seamons, M. Winslett, T. Yu, and C. C. Zhang, “How to exploit ontologies for trust negotiation,” in *In ISWC Workshop on Trust, Security, and Reputation on the Semantic Web*, Hiroshima, Japan, 2004.

[24] M. Blaze, J. Feigenbaum, and J. Lacy, “Decentralized trust management,” in *Proceedings of IEEE Symposium on Security and Privacy*, Oakland, CA, USA, 1996, pp. 164-173.

[25] M. Blaze, J. Feigenbaum, J. Ioannidis, and A. Keromytis, “The KeyNote Trust Management System,” in *International Workshop on Security Protocols*, Berlin, Heidelberg, 1999, pp. 59-63.

[26] L. Xiong, and L. Liu, “A Reputation-based Trust Model for Peer-to-Peer E-Commerce Communities,” in *IEEE International Conference on E-Commerce Technology (CEC)*, Newport Beach, 2003, pp. 275-284.

[27] K. Aberer, and Z. Despotovic, “Managing trust in a peer-2-peer information system,” in *Proceedings of the tenth international conference on Information and knowledge management*, Atlanta, Georgia, USA, 2001, pp. 310-317.

[28] P. Resnick, K. Kuwabara, R. Zeckhauser, and E. Friedman, “Reputation Systems,” *Communications of the ACM*, vol. 43, no. 12, pp. 45-48, 2000.

[29] Y. Liu, W. Xue, K. Li, Z. Chi, G. Min, and W. Qu, “DHTrust: A Robust and Distributed Reputation System for Trusted Peer-to-Peer Networks,” in *2010 IEEE Global Telecommunications Conference (GLOBECOM)*, Miami, FL, USA, 2010, pp. 1-6.

[30] T. Roosta, M. Meingast, and S. Sastry, “Distributed Reputation System for Tracking Applications in Sensor Networks,” in *3rd Annual International Conference on Mobile and Ubiquitous Systems: Networking & Services*, San Jose, CA, USA, 2006, pp. 1-8.

[31] A. Abdul-Rahman, and S. Hailes, "A distributed trust model," in *The New Security Paradigms Workshop*, Langdale, Cumbria, United Kingdom, 1997, pp. 48–60.
[32] A. Abdul-Rahman, and S. Hailes, "Using recommendations for managing trust in distributed systems," in *Proceedings of IEEE International Conference on Communication*, Kuala Lumpur, Malaysia, 1997.
[33] B. Yu, and M. P. Singh, "A social mechanism of reputation management in electronic communities," in *International Workshop on Cooperative Information Agents*, London, UK, 2000, pp. 154–165.
[34] B. Yu, and M. P. Singh, "An evidential model of distributed reputation management," in *Proceedings of the 1st International Joint Conference onAutonomousAgents and Multiagent Systems*, New York, USA, 2002, pp. 294–301.
[35] B. Yu, and M. P. Singh, "Detecting deception in reputation management," in *AAMAS '03: Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems*, New York, USA, 2003, pp. 73–80.
[36] J. Sabater, and C. Sierra, "Reputation and social network analysis in multiagent systems," in *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems*, New York, USA, 2002, pp. 475-482.
[37] T. Beth, M. Borcherding, and B. Klein, "Valuation of trust in open networks," in *Proceedings of the 3rd European Symposium on Research in Computer Security*, Berlin, Heidelberg, 1994, pp. 1-18.
[38] S. Brin, and L. Page, "The anatomy of a large-scale hypertextual Web search engine," *Computer Networks*, vol. 56, pp. 3825-3833, 1998.
[39] S.D.Kamvar, M. T. Schlosser, and H. Garcia-Molina, "The eigentrust algorithm for reputation management in P2P networks," in *12th International Conference on World Wide Web*, New York, NY, USA, 2003, pp. 640-651.
[40] E. Damiani, D. C. d. Vimercati, S. Paraboschi, P. Samarati, and F. Violante, "A reputation-based approach for choosing reliable resources in peer-to-peer networks," in *9th ACM Conference on Computer and Communications Security*, New York, NY, USA, 2002, pp. 207-216.
[41] J. Golbeck, and J. Hendler, "Accuracy of metrics for inferring trust and reputation," in *Proceedings of the 14th International Conference on Knowledge Engineering and Knowledge Management*, Berlin, Heidelberg, 2004, pp. 116-131.
[42] J. Golbeck, and J. Hendler, "Inferring reputation on the semantic web," in *Proceedings of the 13th International World Wide Web Conference*, NY, USA, 2004.
[43] P. Massa, and P. Avesani, "Controversial users demand local trust metrics: an experimental study on epinions.com community," in *25th American Association for Artificial Intelligence Conference*, Pittsburgh, Pennsylvania, 2005, pp. 121-126.
[44] P.-A. Chirita, W. Nejdl, M. Schlosser, and O. Scurtu, "Personalized reputation management in P2P networks," in *Proceedings of the Trust, Security and Reputation Workshop Held at the 3rd International Semantic Web Conference*, Hiroshima, Japan, 2004.
[45] B. W. Husted, "The Ethical Limits of Trust in Business Relations," *Business Ethics Quarterly*, vol. 8, no. 2, pp. 233-248, 2015.

[46] D. Hongwei, L. Albert, and W. Jiming, "Trust in Electronic Commerce: Definitions, Sources, and Effects," *Encyclopedia of E-Business Development and Management in the Global Economy*, L. In, ed., pp. 65-74, Hershey, PA, USA: IGI Global, 2010.

[47] ITU-T, "Trust Provisioning for future ICT infrastructures and services," *Technical Report*, 2016.

[48] L. E. Holmquist, F. Mattern, B. Schiele, P. Alahuhta, M. Beigl, and H.-W. Gellersen, "Smart-Its Friends: A Technique for Users to Easily Establish Connections between Smart Artefacts," in *Proceedings of the 3rd international conference on Ubiquitous Computing*, Atlanta, Georgia, USA, 2001, pp. 116-122.

[49] L. Atzori, A. Iera, and G. Morabito, "From "smart objects" to "social objects": The next evolutionary step of the IoT," *IEEE Communications Magazine*, vol. 52, no. 1, pp. 97-105, 2014.

[50] S. Nepal, W. Sherchan, and C. Paris, "Strust: A trust model for social networks," in *IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, 2011, pp. 841-846.

[51] S. Adali, R. Escriva, M. K. Goldberg, M. Hayvanovych, M. Magdon-Ismail, B. K. Szymanski, W. A. Wallace, and G. Williams, "Measuring behavioral trust in social networks," in *IEEE International Conference on Intelligence and Security Informatics (ISI)*, Vancouver, BC, 2010, pp. 150-152.

[52] Y. Hu, D. Wang, H. Zhong, and F. Wu, "SocialTrust: Enabling long-term social cooperation in p2p services," *Springer P2P Networking and Applications*, vol. 7, no. 4, pp. 525-538, 2014.

[53] M. Nitti, R. Girau, L. Atzori, A. Lera, and G. Morabito, "A Subjective Model for Trustworthiness Evaluation in the Social IoT," in *IEEE Intl. Symposium on Personal Indoor and Mobile Radio Communications, PIMRC*, Australia, 2013, pp. 18-23.

[54] J. Zhan, and X. Fang, "A novel trust computing system for social networks," in *IEEE Third International Conference onPrivacy, Security, Risk and Trust (PASSAT) and Social Computing (SocialCom)*, Boston, MA, USA, 2011, pp. 1284-1289.

[55] G. Yin, F. Jiang, S. Cheng, X. Li, and X. He, "Autrust: A practical trust measurement for adjacent users in social networks," in *Second International Conference on Cloud and Green Computing (CGC)*, 2012, pp. 360-367.

[56] X. L. Dong, E. Gabrilovich, K. Murphy, V. Dang, W. Horn, C. Lugaresi, S. Sun, and W. Zhang, "Knowledge-based trust: Estimating the trustworthiness of web sources," *Proceedings of the VLDB Endowment*, vol. 8, no. 9, pp. 938-949, 2015.

[57] X. L. Dong, E. Gabrilovich, G. Heitz, W. Horn, K. Murphy, S. Sun, and W. Zhang, "From data fusion to knowledge fusion," *Proceedings of the VLDB Endowment*, vol. 7, no. 10, pp. 881-892, 2014.

[58] X. Li, X. L. Dong, K. Lyons, W. Meng, and D. Srivastava, "Truth finding on the deep web: Is the problem solved?," in *Proceedings of the VLDB Endowment*, 2012, pp. 97-108.

[59] Y. Ping, J. Xinghao, W. Yue, and L. Ning, "Distributed intrusion detection for mobile ad hoc networks," *Journal of Systems Engineering and Electronics*, vol. 19, no. 4, pp. 851-859, 2008.

[60] R. Sekar, A. Gupta, J. Frullo, T. Shanbhag, A. Tiwari, H. Yang, and S. Zhou, "Specification-based anomaly detection: a new approach for detecting network

intrusions," in *Proceedings of the 9th ACM conference on Computer and communications security*, Washington, DC, USA, 2002, pp. 265-274.
[61] F. Yu, C. Xu, Y. Shen, J.-y. An, and L.-f. Zhang, "Intrusion detection based on system call finite-state automation machine," in *IEEE International Conference on Industrial Technology (ICIT)*, Hong Kong, China, 2005, pp. 63-68.
[62] M. Blaze, J. Feigenbaum, and J. Lacy, "Decentralized trust management," in *Security and Privacy, Proceedings., IEEE Symposium on*, 1996, pp. 164-173.
[63] T. Grandison, and M. Sloman, "A survey of trust in internet applications," *IEEE Communications Surveys & Tutorials*, vol. 3, no. 4, pp. 2-16, 2000.
[64] Z. Yan, P. Zhang, and A. V. Vasilakos, "A survey on trust management for Internet of Things," *Journal of Network and Computer Applications*, vol. 42, pp. 120-134, 2014.
[65] F. Bao, and I. R. Chen, "Trust Management for the Internet of Things and Its Application to Service Composition," in *IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks*, San Francisco, USA, 2012, pp. 1-6.
[66] I. R. Chen, F. Bao, and J. Guo, "Trust-based Service Management for Social Internet of Things Systems," *IEEE Transactions on Dependable and Secure Computing*, vol. PP, no. 99, pp. 1-1, 2015.
[67] I. R. Chen, and J. Guo, "Dynamic Hierarchical Trust Management of Mobile Groups and Its Application to Misbehaving Node Detection."
[68] W. Conner, A. Iyengar, T. Mikalsen, I. Rouvellou, and K. Nahrstedt, "A trust management framework for service-oriented environments." pp. 891-900.
[69] Z. Movahedi, M. Nogueira, and G. Pujolle, "An autonomic knowledge monitoring scheme for trust management on mobile ad hoc networks." pp. 1898-1903.
[70] Z. Yan, and R. MacLaverty, "Autonomic Trust Management in a Component Based Software System," *Autonomic and Trusted Computing: Third International Conference, ATC 2006, Wuhan, China, September 3-6, 2006. Proceedings*, L. T. Yang, H. Jin, J. Ma and T. Ungerer, eds., pp. 279-292, Berlin, Heidelberg: Springer Berlin Heidelberg, 2006.
[71] M. Firdhous, O. Ghazali, and S. Hassan, "Trust Management in Cloud Computing: A Critical Review," *Advances in ICT for Emerging Regions (ICTer)*, vol. 04, no. 2, pp. 24-36, 2012.
[72] G. Barbian, "Trust Centrality in Online Social Networks," in *in European Intelligence and Security Informatics Conference (EISIC)*, Athens, Greece, 2011, pp. 372-377.
[73] W. Yan, L. Lei, and L. Ee-Peng, "Price Trust Evaluation in E-service Oriented Applications," in *E-Commerce Technology and the Fifth IEEE Conference on Enterprise Computing, E-Commerce and E-Services, ,* Washington DC, USA, 2008, pp. 165-172.
[74] L. Mui, "Computational models of trust and reputation : agents, evolutionary games, and social networks," Ph.D. Thesis, Massachusetts Institute of Technology, 2003.
[75] G. Zacharia, and P. Maes, "Collaborative Reputation Mechanisms for Online Communities," Ph.D. Thesis, Dept. of Architecture, Massachusetts Institute of Technology, 2005.
[76] S. S. Park, J. H. Lee, and T. M. Chung, "Cluster-based trust model against attacks in ad-hoc networks," in *IEEE Third International Conference on*

Convergence and Hybrid Information Technology (ICCIT), Busan, South Korea, 2008, pp. 526-532.

[77] F. Bao, and I.-R. Chen, "Dynamic trust management for internet of things applications," in *Proceedings of the 2012 international workshop on Self-aware internet of things*, 2012, pp. 1-6.

[78] M. J. Probst, and S. K. Kasera, "Statistical trust establishment in wireless sensor networks," in *Parallel and Distributed Systems, 2007 International Conference on*, Hsinchu, Taiwan, 2007, pp. 1-8.

[79] A. Jøsang, R. Ismail, and C. Boyd, "A survey of trust and reputation systems for online service provision," *Decision Support Systems*, vol. 43, no. 2, pp. 618-644, 2007.

[80] C. Dellarocas, "Immunizing online reputation reporting systems against unfair ratings and discriminatory behavior," in *Proceedings of the 2nd ACM Conference: Electronic Commerce*, Minnesota, USA, 2000, pp. 150-157

[81] N. Iltaf, A. Ghafoor, U. Zia, and M. Hussain, "An Effective Model for Indirect Trust Computation in Pervasive Computing Environment," *Wireless Personal Communications*, vol. 75, no. 3, pp. 1689-1713, 2014/04/01, 2014.

[82] L. Zhaoyu, A. W. Joy, and R. A. Thompson, "A dynamic trust model for mobile ad hoc networks," in *10th IEEE International Workshop on Future Trends of Distributed Computing Systems (FTDCS)*, Suzhou, China, 2004, pp. 80-85.

[83] A. Boukerche, and Y. Ren, "A security management scheme using a novel computational reputation model for wireless and mobile Ad hoc networks," in *Proceedings of the 5th ACM symposium on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks*, Vancouver, British Columbia, Canada, 2008, pp. 88-95.

[84] B. Lagesse, M. Kumar, M. Wright, and J. M. Paluska, "DTT: A distributed trust toolkit for pervasive systems," in *IEEE International Conference on Pervasive Computing and Communications*, Galveston, TX, USA, 2009, pp. 1-8.

[85] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.

[86] N. Fotiou, and G. C. Polyzos, "Decentralized name-based security for content distribution using blockchains," in *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, San Francisco, CA, USA, 2016, pp. 415-420.

[87] N. Alexopoulos, J. Daubert, M. Mühlhäuser, and S. M. Habib, "Beyond the Hype: On Using Blockchains in Trust Management for Authentication," in *IEEE Trustcom/BigDataSE/ICESS*, Sydney, NSW, Australia, 2017, pp. 546-553.

[88] F. Bao, and I.-R. Chen, "Dynamic Trust Management for the Internet of Things Applications," in *International Workshop on Self-Aware Internet of Things*, San Jose, USA, 2012, pp. 1-6.

[89] D. Chen, G. Chang, D. Sun, J. Li, J. Jia, and X. Wang, "TRM-IoT: a trust management model based on fuzzy reputation for Internet of Things," *Computer Science and Information Systems*, vol. 8, pp. 1207-1228, 2011.

[90] N. B.Truong, T.-W. Um, and G. M. Lee, "A Reputation and Knowledge Based Trust Service Platform for Trustworthy Social Internet of Things," in *Innovations in Clouds, Internet and Networks (ICIN)*, Paris, France, March 2016.

[91] A. Jøsang, "A logic for uncertain probabilities," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems,* vol. 9, pp. 279– 311, June 2001.

[92] F. Bao, I. R. Chen, and J. Guo, "Scalable, Adaptive and Survivable Trust Management for Community of Interest Based Internet of Things Systems," in *11th International Symposium on Autonomous Decentralized System*, Mexico City, Mexico, 2013, pp. 1-7.

[93] I. R. Chen, and J. Guo, "Dynamic Hierarchical Trust Management of Mobile Groups and Its Application to Misbehaving Node Detection," in *International Conference on. Advanced Information Networking and Applications*, Victoria, Canada, 2014, pp. 49-56.

[94] S. Hauke, S. Biedermann, M. Mühlhäuser, and D. Heider, "On the Application of Supervised Machine Learning to Trustworthiness Assessment," in *12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications*, Melbourne, VIC, Australia, 2013, pp. 525-534.

[95] K. Zhao, and L. Pan, "A Machine Learning Based Trust Evaluation Framework for Online Social Networks," in *13th IEEE International Conference on Trust, Security and Privacy in Computing and Communications*, Beijing, China, 2014, pp. 69-74.

[96] S. Weihua, and V. V. Phoha, "Neural network-based reputation model in a distributed system," in *IEEE International Conference on e-Commerce Technology (CEC)*, Beijing, China, 2004, pp. 321-324.

[97] F. Fei, S. Li, H. Dai, C. Hu, W. Dou, and Q. Ni, "A K-Anonymity Based Schema for Location Privacy Preservation," *IEEE Transactions on Sustainable Computing,* vol. PP, no. 99, pp. 1-1, 2017.

[98] F. Jiang, Y. Fu, B. B. Gupta, F. Lou, S. Rho, F. Meng, and Z. Tian, "Deep Learning based Multi-channel intelligent attack detection for Data Security," *IEEE Transactions on Sustainable Computing,* vol. PP, no. 99, pp. 1-1, 2018.

[99] J. Shen, D. Liu, D. He, X. Huang, and Y. Xiang, "Algebraic Signatures-based Data Integrity Auditing for Efficient Data Dynamics in Cloud Computing," *IEEE Transactions on Sustainable Computing*, pp. 1-1, 2017.

[100] Y. Wang, Y.-C. Lu, I.-R. Chen, J.-H. Cho, A. Swami, and C.-T. Lu, "LogitTrust: A Logit Regression-based Trust Model for Mobile Ad Hoc Networks," in *Proceedings of the 6th ASE International Conference on Privacy, Security, Risk and Trust* Cambridge, MA, 2014, pp. 1-10.

[101] Z. Li, X. Li, V. Narasimhan, A. Nayak, and I. Stojmenovic, "Autoregression Models for Trust Management in Wireless Ad Hoc Networks," in *IEEE Global Telecommunications Conference (GLOBECOM)*, Kathmandu, Nepal, 2011, pp. 1-5.

[102] F. Boustanifar, and Z. Movahedi, "A Trust-Based Offloading for Mobile M2M Communications," in *Intl IEEE Conferences on Ubiquitous Intelligence & Computing*, Toulouse, France, 2016, pp. 1139-1143.

[103] W. Li, W. Meng, L.-F. Kwok, and H. Horace, "Enhancing collaborative intrusion detection networks against insider attacks using supervised intrusion sensitivity-based trust management model," *Journal of Network and Computer Applications,* vol. 77, pp. 135-145, 2017.

[104] A. Bolster, and A. Marshall, "Analytical metric weight generation for multi-domain trust in autonomous underwater MANETs," in *IEEE Third Underwater Communications and Networking Conference (UComms)*, Lerici, Italy, 2016, pp. 1-5.

[105] T. Yu, and M. Winslett, “Policy migration for sensitive credentials in trust negotiation,” in *Proceedings of the 2003 ACM Workshop on Privacy in the Electronic Society*, New York, USA, 2003, pp. 9-20.
[106] M. J. Probst, and S. K. Kasera, "Statistical trust establishment in wireless sensor networks." pp. 1-8.
[107] A. Boukerche, and Y. Ren, "A security management scheme using a novel computational reputation model for wireless and mobile Ad hoc networks." pp. 88-95.
[108] B. Lagesse, M. Kumar, M. Wright, and J. M. Paluska, "DTT: A distributed trust toolkit for pervasive systems."
[109] J. Audun, R. I. smail, and C. Boyd, “A survey of trust and reputation systems for online service provision,” *Decis. Support Syst.*, vol. 43, no. 2, pp. 618-644, 2007.
[110] C. Dellarocas, “Immunizing online reputation reporting systems against unfair ratings and discriminatory behavior,” *Proceedings of the 2nd ACM Conference: Electronic Commerce*, pp. 150, 10/17/, 2000.
[111] L. Zhaoyu, A. W. Joy, and R. A. Thompson, “A dynamic trust model for mobile ad hoc networks,” in *Distributed Computing Systems, 2004. FTDCS 2004. Proceedings. 10th IEEE International Workshop on Future Trends of*, Suzhou, China, 2004, pp. 80-85.
[112] A. Rajaram, and D. S. Palaniswami, “A Trust Based Cross Layer Security Protocol for Mobile Ad hoc Networks,” *International Journal of Computer Science and Information Security, IJCSIS*, vol. 6, no. 1, pp. 165-172, 2009.
[113] L.-H. Vu, M. Hauswirth, and K. Aberer, “QoS-Based service selection and ranking with trust and reputation management,” in *Proceedings of the Confederated international conference on "On the Move to Meaningful Internet Systems"*, Berlin, Heidelberg, 2005, pp. 466-483.
[114] T. Jim, “SD3: a trust management system with certified evaluation,” in *Proceedings 2001 IEEE Symposium on Security and Privacy S&P*, 2001, pp. 106-115.
[115] L. Xiong, and L. Liu, “A reputation-based trust model for peer-to-peer e-commerce communities,” in *IEEE Intl. Conference on E-Commerce,CEC.*, Newport Beach, 2003, pp. 275-284.
[116] M. Blaze, J. Feigenbaum, J. Ioannidis, and A. Keromytis, "The KeyNote Trust-Management System Version 2," *RFC Editor*, 1999.
[117] M. A. Azer, S. M. El-Kassas, A. W. F. Hassan, and M. S. El-Soudani, “A survey on trust and reputation schemes in ad hoc networks,” in *Third International Conference on Availability, Reliability and Security, ARES* Barcelona, Spain, 2008, pp. 881-886.
[118] G. Suryanarayana, and R. N. Taylor, “A survey of trust management and resource discovery technologies in peer-to-peer applications,” ISR Technical Report UCI-ISR-04-6, University of California, 2004.
[119] ITU-T, "Overview of trust provisioning for information and communication technology infrastructures and services," *Recommendation Y.3052*, 2017.
[120] I. Pranata, G. Skinner, and R. Athauda, “A Holistic Review on Trust and Reputation Management Systems for Digital Environments,” *International Journal of Computer and Information Technology*, vol. 1, no. 1, pp. 44-53, 2012.

[121] E. Chang, F. K. Hussain, and T. S. Dillon, “Fuzzy nature of trust and dynamic trust modelling in service oriented environments,” in *Workshop on secure web services*, Fairfax, USA, 2005, pp. 75-83.
[122] E. Chang, T. Dillon, and F. K. Hussain, "Technologies for building business intelligence and consumer confidence," *Trust Reputation for Service-Oriented Environments*, West Sussex, England: John Wiley & Sons Ltd, 2006.
[123] C. Castelfranchi, and R. Falcone, “Principles of trust for MAS: Cognitive anatomy, social importance, and quantification,” in *International Conference on Multi Agent Systems*, Paris, France, 1998, pp. 72-79.
[124] U. Jayasinghe, N. B. Truong, G. M. Lee, and T.-W. Um, “RpR: A Trust Computation Model for Social Internet of Things,” in *Smart World Congress , Intl IEEE Conferences on Ubiquitous Intelligence & Computing*, Toulouse, France, 2016, pp. 930-937.
[125] G. M. Lee, U. Jayasinghe, N. B. Truong, and C.-h. Cho, “Features, Challenges and Technical Issues,” in *The Second Bright ICT Annual Workshop on Bright ICT 2016*, Dublin, Ireland, 2016.
[126] X. Yang, P. Moore, and J. S. Pu, “Service Enabler: A Software Agent using Lifecycle Data to Enable Knowledge-Intensive Services,” in *Third International Conference on Semantics, Knowledge and Grid*, Shan Xi, China, 2007, pp. 519-522.
[127] S. Brin, and L. Page, “Reprint of: The Anatomy of a Large-Scale Hypertextual Web Search Engine,,” *Computer Networks*, vol. 56, pp. 3825-3833, 2012.
[128] J. Fei, Y. Yang, J. Shuyuan, and X. Jin, “Fast Search to Detect Communities by Truncated Inverse Page Rank in Social Networks,” in *IEEE International Conference on Mobile Services (MS)*, New York, USA, 2015, pp. 239-246.
[129] F. L. Dragomir, “Models of Trust and Reputation in eCommerce,” *Acta Universitatis Danubius. Œconomica,* vol. 12, no. 6, 2017.
[130] M. T. Ltd. "Calculate distance, bearing and more between Latitude/Longitude points," 2016; [Online] Available: http://www.movable-type.co.uk/scripts/latlong.html.
[131] D. J. MacKay,“Information theory, inference and learning algorithms,” Cambridge university press, 2003.
[132] MathWorks. "Interpret Linear Regression Results," 2016; [Online] Available: https://uk.mathworks.com/help/stats/understanding-linear-regression-outputs.html.
[133] A.-K. Pietil, E. Oliver, J. LeBrun, G. Varghese, and C. Diot, “MobiClique: middleware for mobile social networking,” in *Proceedings of the 2nd ACM workshop on Online social networks*, Barcelona, Spain, 2009, pp. 49-54.
[134] P. Anna-Kaisa, and D. Christophe. "CRAWDAD dataset thlab/sigcomm2009 (v. 2012-07-15)," [Online] Available: http://crawdad.org/thlab/sigcomm2009/20120715.
[135] K. Govindan, and P. Mohapatra, “Trust computations and trust dynamics in mobile adhoc networks: A survey,” *IEEE Communications Surveys & Tutorials*, vol. 14, no. 2, pp. 279-298, 2012.
[136] M. Nitti, R. Girau, and L. Atzori, “Trustworthiness management in the social internet of things,” *IEEE Transactions on knowledge and data engineering*, vol. 26, no. 5, pp. 1253-1266, 2014.
[137] X. Amatriain, and J. M. Pujol, "Data Mining Methods for Recommender Systems," *Recommender Systems Handbook*, pp. 227-262, Boston, MA: Springer, 2015.

[138] S. Zafeiriou. "Notes on Implementation of Component AnalysisTechniques," 2015; [Online] Available: https://ibug.doc.ic.ac.uk/media/uploads/documents/notes_implementation_component_analysis.pdf.
[139] C. W. Hsu, C. C. Chang, and C. J. Lin, "A practical guide to support vector classification," Tech. rep., Dept. Computer Science, National Taiwan University, 2003.
[140] C. Chih-Chung, and L. Chih-Jen, "LIBSVM : A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, pp. 27:1-27:27, 2011.
[141] I. Jolliffe,"Principal component analysis," New York: Wiley Online Library,2002.
[142] M. D. Zeiler, "ADADELTA: an adaptive learning rate method," *arXiv preprint arXiv:1212.5701*, 2012.
[143] D. C. Liu, and J. Nocedal, "On the limited memory BFGS method for large scale optimization," *Mathematical Programming*, vol. 45, no. 1, pp. 503-528, August 01, 1989.
[144] F. Fei, S. Li, H. Dai, C. Hu, W. Dou, and Q. Ni, "A K-Anonymity Based Schema for Location Privacy Preservation," *IEEE Transactions on Sustainable Computing*, 2017.
[145] J. Shen, D. Liu, D. He, X. Huang, and Y. Xiang, "Algebraic Signatures-based Data Integrity Auditing for Efficient Data Dynamics in Cloud Computing," *IEEE Transactions on Sustainable Computing*, 2017.
[146] Y. Wang, Y.-C. Lu, I.-R. Chen, J.-H. Cho, A. Swami, and C.-T. Lu, "LogitTrust: A Logit Regression-based Trust Model for Mobile Ad Hoc Networks," in *Proceedings of the 6th ASE International Conference on Privacy, Security, Risk and Trust*, Cambridge, MA, 2014, pp. 1-10.
[147] F. Boustanifar, and Z. Movahedi, "A Trust-Based Offloading for Mobile M2M Communications," in *Intl IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld)*, Toulouse, France, 2016, pp. 1139-1143.
[148] N. Askham, D. Cook, M. Doyle, H. Fereday, M. Gibson, U. Landbeck, R. Lee, C. Maynard, G. Palmer, and J. Schwarzenbach, "The six primary dimensions for data quality assessment," Technical report, DAMA UK Working Group, 2013.
[149] Y. W. Lee, D. M. Strong, B. K. Kahn, and R. Y. Wang, "AIMQ: a methodology for information quality assessment," *Information & management*, vol. 40, no. 2, pp. 133-146, 2002.
[150] P. R. Benson, "ISO 8000 Data Quality," *The Fundamentals Part*, 2009.
[151] U. Jayasinghe, G. M. Lee, T.-W. Um, and Q. Shi, "Machine-Learning-based Trust Computational Model for IoT Services," *IEEE Transactions on Sustainable Computing*, vol. Accepted for Publication, 2018.
[152] U. Jayasinghe, N. B. Truong, G. M. Lee, and T.-W. Um, "RpR: A Trust Computation Model for Social Internet of Things," in *2016 Intl IEEE Conference on Smart World Congress*, Toulouse, France, 2016.
[153] N. B. Truong, H. Lee, B. Askwith, and G. M. Lee, "Toward a Trust Evaluation Mechanism in the Social Internet of Things," *Sensors*, vol. 17, no. 6, pp. 1346, 2017.

[154] Y. Koren, “Factorization meets the neighborhood: a multifaceted collaborative filtering model,” in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, Las Vegas, Nevada, USA, 2008, pp. 426-434.
[155] Y.-x. Yuan, “Step-sizes for the gradient method,” *AMS IP Studies in Advanced Mathematics*, vol. 42, no. 2, pp. 785, 2008.
[156] A. Jian, G. Xiaolin, Y. Jianwei, S. Yu, and H. Xin, “Mobile crowd sensing for internet of things: A credible crowdsourcing model in mobile-sense service,” in *IEEE International Conference on Multimedia Big Data (BigMM)*, Beijing, China, 2015, pp. 92-99.
[157] Twin Oaks Computing Inc. "CoreDX Distributed Publish-Subscribe System," 03/05/2017; [Online] Available: http://www.twinoakscomputing.com/coredx/develop.
[158] L. L. Pipino, Y. W. Lee, and R. Y. Wang, “Data quality assessment,” *Commun. ACM*, vol. 45, no. 4, pp. 211-218, 2002.
[159] B. Heinrich, M. Kaiser, and M. Klier, “How to measure data quality? A metric-based approach,” 2007.
[160] S. Mazilu, M. Teler, and C. Dobre, "Securing vehicular networks based on data-trust computation." pp. 51-58.
[161] M. Raya, P. Papadimitratos, V. D. Gligor, and J.-P. Hubaux, "On data-centric trust establishment in ephemeral ad hoc networks." pp. 1238-1246.
[162] P. Borzymek, M. Sydow, and A. Wierzbicki, "Enriching trust prediction model in social network with user rating similarity." pp. 40-47.
[163] N. Korovaiko, and A. Thomo, “Trust prediction from user-item ratings,” *Social Netw. Analys. Mining*, vol. 3, no. 3, pp. 749-759, 2013.
[164] R. Xiang, J. Neville, and M. Rogati, "Modeling relationship strength in online social networks." pp. 981-990.
[165] ITU-T SG13. "Future networks, with focus on IMT-2020, cloud computing and trusted network infrastructures,," [Online] Available: https://www.itu.int/en/ITU-T/studygroups/2017-2020/13/Pages/default.aspx.
[166] IBM Corporation, “An architectural blueprint for autonomic computing,” Autonomic Computing White Paper, 2005.
[167] M. Maggio, H. Hoffmann, A. V. Papadopoulos, J. Panerati, M. D. Santambrogio, A. Agarwal, and A. Leva, “Comparison of Decision-Making Strategies for Self-Optimization in Autonomic Computing Systems,” *ACM Trans. Auton. Adapt. Syst.*, vol. 7, no. 4, pp. 1-32, 2012.

APPENDIX A: CRAWDAD Data Set

The dataset contains data collected by an opportunistic mobile social application, MobiClique. The application was used by conference attendees during SIGCOMM 2009 conference in Barcelona, Spain. The data sets include traces of Bluetooth device proximity, opportunistic message creation and dissemination, and the social profiles (friends and interests) of the participants [133], [134].

A.1. Participants

Format: *csv: user_id;key;value*

List of participants and basic social profile including home city, country, and affiliation. The user_ids run from 1 to 76 (inclusive). Each user carries a single device that is identified by the same user_id. The 'key' is one of ['institute','city','country'] and the values are anonymized to simple integer ids.

A.2. Interests1

Format: *csv: user_id;group_id*

List of initial interest groups of the participants based on their Facebook groups and networks. The list contains also three pre-configured common groups for each participants (ids 1,2,3).

A.3. Friends1

Format: *csv: user_id;friend_user_id*

List of friends of the participants based on their Facebook friends.

A.4. Interests2

Format: *csv: user_id;group_id;timestamp*

The MobiClique application lets users to discover and join existing interest groups, and create new interest groups at any time. Hence, the interest lists are changing over time.

A.5. Friends2

Format: *csv: user_id;friend_user_id;timestamp*

Similarly, to the interest groups, the MobiClique application lets users to discover and friend other MobiClique users upon opportunistic encounters with them. Hence, the friendship graph is changing over time.

A.6. Activity

Format: *csv: user_id;start;end*

A device is active when it is collecting data. The inactivity periods occur due to batteries running out, at night time when the device is turned off, and due to some software problems.

A.7. Proximity

Format: *csv: timestamp;user_id;seen_user_id;seen_device_major_cod;seen_device_minor_cod*

The trace records all the nearby Bluetooth devices reported by the periodic Bluetooth device discoveries.

A.8. Messages

Format: *csv: msg_id;src_user_id;created;type;dst*

This file lists the application level messages created by the users during the experiment. MobiClique allowed messaging between friends or among members of an interest group. In addition, MobiClique contained an epidemic voting application that allowed users to give rankings (1 to 5 stars) to the talks of the conference and see the real time results on their device.

A.9. Transmission

Format: *csv: type;msg_id;bytes;hop_src_user_id;hop_dst_user_id;src_timestamp;status*

The data transmission protocol logs from the sender side. Data is transmitted between two devices using Bluetooth RFCOMM protocol on a fixed channel (no service discovery required).

A.10. Reception

Format: *csv: type;msg_id;bytes;hop_src_user_id;hop_dst_user_id;src_timestamp;status*

The data transmission protocol logs from the receiver side. Data is transmitted between two devices using Bluetooth RFCOMM protocol on a fixed channel (no service discovery required).

APPENDIX B: Simulation of the Trust Model based on Indirect Trust

B.1. Main Function: ReputeRank.m

```
clc
clear
for n = 5:5:100                          %Number of nodes
    if n<51
        load(['mat' num2str(n) '.mat'])
        load(['Umat' num2str(n) '.mat'])
        A=smat;                          %Transition Matrix
        U=U;                             %Inverse Transition Matrix
    else
        A=stomat(n);
        U=invstomat(A);
    end
    % stopping residual
    err= 1;
    tol=1e-2;
    N=size(A,1);
    PR=recrank1(err,tol,N,A);    % Recommendation rank
    REC=recrank2(err,tol,N,A,U); % Recommendation rank & clustering
    REP=reprank(err,tol,N,A,U);  % Reputation Rank
    g=0.8;
    HYB=g*REC+(1-g)*REP;         % RPR Rank
    G=digraph(A');
    indeg = indegree(G);
    ID=indeg./sum(indeg);
    C{n/5}=[ID PR REC REP HYB];
End
results(A,U,PR,REC,REP,HYB)
```

B.2. Recommendation Rank: recrank1.m

```
function R=recrank1(err,tol,N,A)  %initial guess for r
R = 1/N*ones(N,1);
while(err>tol) %iterating until the stopping criterion is not met
    S = A*R;       % pre-multiply the current rec rank vector
    err=norm(S-R); % use an absolute stopping residual
    R=S;           % update the rec rank vector
end
```

B.3. Recommendation Rank with clustering: recrank2.m

```
function R=recrank2(err,tol,N,A,U)
t=invrank(U,err,tol);                    %Calculate inverse rec rank
R = t;                                   %initial guess for r
a=0.8;
while(err>tol)
    S=a*A*R+(1-a)*t;
    err=norm(S-R);
    R=S;
end
```

B.4. Reputation Rank: reprank.m

```
function Rf=reprank(err,tol,N,A,U)
tr=invrank(U,err,tol);
```

```
3.  Rf=zeros(N,1);
4.  beta=0.8;
5.  for n=2:4                   %Graph Length Level
6.      err=1;
7.      T=A^n;
8.      R = tr;                 %Trust Vector
9.      while(err>tol)
10.         S= beta*T^n*R+(1-beta)*tr;
11.         err=norm(S-R);
12.         R=S;
13.     end
14.     Rf(:,n-1)=R;
```

B.5. Inverse Rank: invrank.m

```
1.  function t=invrank(U,err,tol)
2.  N=size(U,1);                    % initial guess for IR
3.  IR = 1/N * ones(N,1);
4.  while(err>tol)
5.      IS = U*IR;
6.      err=norm(IS-IR);
7.      IR=IS;
8.  end
9.  x = IR/sum(IR);      %normalize the page rank to have unit sum
10. m=sort(x(:),'descend'); %Selecting trustworthy nodes
11. tn=N/5;                 %tn number of highest nodes
12. t=(x>=m(tn)).*x;
13. t=(t>0);
14. t=(t>0)./tn;
```

B.6. Generate Transition Matrix: stomat.m

```
1.  function smat=stomat(n)
2.      x=randsrc(n,n,[0 1;0.8 0.2]);
3.      x(all(x==0,2),:)=[];
4.      x(:,all(x==0,1))=[];
5.      [a,b]=size(x);
6.      m=min(a,b);
7.      x=x(1:m,1:m);
8.      x=x-diag(diag(x));
9.      nrm=sum(x,1);
10. div=repmat(nrm,size(x,1),1);
11. smat=x./div;
12. size(smat);
```

B.7. Generate Inverse Transition Matrix: invstomat.m

```
1.  function ismat=invstomat(AA)
2.  % load('mat10.mat')
3.  % A=smat
4.  A=AA;
5.  G=digraph(A');
6.   [sOut,tOut] = findedge(G);
7.  s=tOut;
8.  t=sOut;
9.  I = digraph(s,t);
10. Ua=adjacency(I);
11. Ua=Ua';
12. U=full(Ua);
13. div=repmat(sum(U,1),size(U,1),1);
```

```
14. U=U./div;
15. ismat=U;
16. sum(sum(A))
17. sum(sum(U))
```

APPENDIX C: Simulation of the Trust Model based on Direct Trust

Note: filename.mat files contains the transmission logs obtained [133], [134].

C.1. Co-work Relationship (CWR)

```
load('msgmc.mat');
for n=1:76
    nn=n;
    row=msgmc.userid==n;
    totmc=size(msgmc.userid(row,:),1);
    for m=1:76
        a=msgmc(row,:);
        if(isempty(a))
            continue;
        else
            mc=sum(a.dst==m);
            mul(n,m)=mc;
            cw(n,m)=mc/totmc;
        end
    end
end
```

C.2. Cooperativeness, Frequency and Duration (CFD)

```
clc
load('tranuc.mat')
load('receuc.mat')
for n=1:76
    nn=n;
    for m=1:76
        t=tranuc(tranuc.src==n & tranuc.dst==m, :);  % n (Frequency)
        r=receuc(receuc.src==m & receuc.dst==n, :);
        if (isempty(t)||isempty(r))
            continue;
        else
            totmsg=size(t,1)+size(r,1);
            suc=t(t.status==1,:);
            sucmsg=size(suc,1);
            sucbyt=sum(suc.bytes);                % cm
            totbyt=sum(t.bytes);                  % tm
            p=sucmsg/totmsg;                      % p
            cm=sucbyt/totbyt;                     % cm/tm (Duration)
            if(p~=0)
                en=-p*log2(p)-(1-p)*log2(1-p);    % entropy
                cop(n,m)=cm*en;
            else
                cop(n,m)=0;
            end
        end
    end
end

```

C.3. Reward System (RS)

```
load('tx.mat');
```

```
for n=1:76
    for m=1:76
        row=tx.src==n & tx.dst==m;
        if(sum(row))
            tot=size(tx.status(row,:),1);
            suc=sum(tx.status(row,:));
            pnc=(suc/tot)*exp(-(tot-suc)/tot);
            pn(n,m)=pnc;
        else
            pn(n,m)=0.9;
        end
    end
end

```

C.4. Mutuality and Centrality (MC)

```
load('frnd.mat');
for n=1:76                          %network size 76 nodes
    nn=n;
    row=frnd.srcid==n;
    totf=size(frnd.srcid(row,:),1);%number of friends of B (Trustee)
    for m=1:76
        rsrc1=frnd.srcid==n;
        rsrc2=frnd.srcid==m;
        a=frnd(rsrc1,:);
        b=frnd(rsrc2,:);
        if(isempty(a)||isempty(b))
            continue;
        else   %compute total number of mutual friend between A & B
            cm=sum(ismember(a.frnid,b.frnid));
            cn=cm/totf;                  % Centrality
            comm(n,m)=cm;
            cent(n,m)=cn;                % Centrality of each pair
        end
     end
end
```

C.5. Community of Interest (CoI)

```
load('int1.mat');
for n=1:76
    nn=n;
    row=int1.userid==n;
    totc=size(int1.userid(row,:),1);
    for m=1:76
        rsrc1=int1.userid==n;
        rsrc2=int1.userid==m;
        a=int1(rsrc1,:);
        b=int1(rsrc2,:);
        if(isempty(a)||isempty(b))
            continue;
        else
            cmi=sum(ismember(a.grpid,b.grpid));
            cni=cmi/totc;
            cint1(n,m)=cmi;
            comint(n,m)=cni;
        end
     end
end
```

APPENDIX D: Simulation of the ML Model

Note: File fullfeat.mat contains the five feature vectors; CLR, CWR, MC, CFD, and RS.

Algorithm I: Clustering and Labeling

D.1. Main File

```
load('fullfeat.mat');
X=[X(:,1) X(:,4)];
[m n] = size(X);
max_iters = 10;
kk=5;                  % Settings for running K-Means
J=zeros(kk,1);
for K=1:kk
   initial_centroids = kMeansInitCentroids(X, K);
    [centroids, idx] = runkMeans(X, initial_centroids, max_iters,
false);
    fprintf('\nK-Means K=%d.\n\n',K);
    %Optimization : Run Elbow method
    dist=0;
    for i=1:m
        dist=dist+sum((X(i,:)- centroids(idx(i),:)).^2);
    end
    J(K)=dist/m;
end
figure (2);
plot(1:kk,J)
```

D.2. Initial Centroids

```
function centroids = kMeansInitCentroids(X, K)
centroids = zeros(K, size(X, 2));
randidx = randperm(size(X, 1));
centroids = X(randidx(1:K), :);
end
```

D.3. K-means Clustering

```
function [centroids, idx] = runkMeans(X, initial_centroids,
max_iters, plot_progress)
if ~exist('plot_progress', 'var') || isempty(plot_progress)
    plot_progress = false;   % Set default value for plot progress
end
if plot_progress
    figure;
    hold on;
end
[m n] = size(X);             % Initialize values
K = size(initial_centroids, 1);
centroids = initial_centroids;
previous_centroids = centroids;
idx = zeros(m, 1);
for i=1:max_iters          % Run K-Means
% For each example in X, assign it to the closest centroid
    idx = findClosestCentroids(X, centroids);
    if plot_progress            % Optionally, plot progress here
```

```
        plotProgresskMeans(X,centroids,previous_centroids,idx,K,i);
        previous_centroids = centroids;
    end
    % Given the memberships, compute new centroids
    centroids = computeCentroids(X, idx, K);
end
if plot_progress  % Hold off if we are plotting progress
    hold off;
end
end
```

D.4. Find Optimum centroid and closet Centroid for Data points

```
function centroids = computeCentroids(X, idx, K)
[m n] = size(X);
centroids = zeros(K, n);
for i=1:K
    row=(idx==i);
    x=row.*X;
    centroids(i,:)=(1/sum(row))*sum(x,1);
end
end

function idx = findClosestCentroids(X, centroids)
K = size(centroids, 1);
idx = zeros(size(X,1), 1);
dist=zeros(size(centroids, 1),1);
for i=1:size(X,1)
    for j=1:K
       dist(j)=sum((X(i,:)- centroids(j,:)).^2);
    end
    [m,id]=min(dist);
    idx(i)=id;
end
end
```

D.5. Plot Data Points

```
function plotDataPoints(X, idx, K)  % same idx have the same colour
palette = hsv(3);                   % Create palette
colors = palette(idx, :);
h=scatter(X(:,1), X(:,2),15,colors);
end
```

Algorithm II: Weight learning and Decision Boundary

D.6. Main File

```
load('svmlabeled.mat')
XX=svm_cwlpn;
[m n]=size(XX);
k = randperm(m);
Xt=XX(k(1:4620),:) ;  %Cross validation 20%
Xv=XX(k(4620:end),:);
X=Xt(:,1:2); y=Xt(:,3);
Xval=Xv(:,1:2); yval=Xv(:,3);

numLabels=[1 0 -1];
for k=1:size(numLabels,2)
```

```
12.     [C, sigma] = BestParamsLIBSVM(X, double(y==numLabels(k)), Xval,
    double(yval==numLabels(k)));
13.     gamma=1/(2*sigma^2);
14.     model = svmtrain(double(y==numLabels(k)), X,sprintf('-q -s 0 -t
    2 -g %g -c %g', gamma,C));
15.     fprintf('Class No %d \n',k);
16.     visualizeBoundaryIBSVMNL2_3d(X, y, model);
17.     C(:,k)=C;
18.     gamma(:,k)=gamma;
19.     w(:,k) = model.SVs' * model.sv_coef;
20.     % ========== Accuracy==========
21.     %predictions=svmpredict(yy,Xval ,model, '-q');
22.     [T,predicted_labels, accuracy, decision_values] =
    evalc('svmpredict(double(yval==numLabels(k)),Xval,model)');
23.      clear T;
24.      acc(k,:)=accuracy;
25. end
```

D.7. Find Optimum parameters for C and gamma

```
1.  function [C, sigma] = BestParamsLIBSVM(X, y, Xval, yval)
2.  s=[0.01 0.03 0.1 0.3 1 3 10 30];
3.  error=zeros(8,8);
4.  for i=1:8
5.      for j=1:8
6.          %fprintf('s(i)=%f \t s(j)=%f \t \n',s(i),s(j));
7.          C=s(i);
8.          sigma=s(j);
9.          gamma=1/(2*sigma^2);
10. %model= svmTrain(X, y, C, @(x1, x2) gaussianKernel(x1, x2, sigma));
11.         model = svmtrain(y, X,sprintf('-q -s 0 -t 2 -g %g -c %g',
    gamma,C));
12. %predictions = svmPredict(model, Xval);
13. %yy=zeros(size(yval,1),1);
14.         predictions=svmpredict(yval,Xval ,model, '-q');
15.         error(i,j)= mean(double(predictions ~= yval));
16.     end
17. end
18. [M,I] = min(error(:));
19. [I_row, I_col] = ind2sub(size(error),I);
20. C=s(I_row);
21. sigma=s(I_col);
22. end
```

D.8. Plots a non-linear decision boundary

```
1.  function visualizeBoundaryIBSVMNL2_3d(X, y, model, varargin)
2.  plotData (X, y)
3.  x1plot = linspace(min(X(:,1)), max(X(:,1)), 100)';
4.  x2plot = linspace(min(X(:,2)), max(X(:,2)), 100)';
5.  [X1, X2] = meshgrid(x1plot, x2plot);
6.  vals = zeros(size(X1));
7.  for i = 1:size(X1, 2)
8.     this_X = [X1(:, i), X2(:, i)];
9.     yy=zeros(size(this_X,1),1);
10.    vals(:, i) =svmpredict(yy,this_X ,model, '-q');
11.    %vals(:, i)=predicted_labels;
12. end
13. hold on  % Plot the SVM boundary
14. contour(X1, X2, vals, [0.5 0.5] ,'color',rand(1,3));
15. end
16.  function plotData(X, y)
```

```
17. pos = find(y == 1); neg = find(y == 0); % Find Indices of Positive
    and Negative Examples
18.  plot(X(pos, 1), X(pos, 2), 'k+','LineWidth', 1, 'MarkerSize', 7)
19. hold on;
20. plot(X(neg, 1), X(neg, 2), 'ko', 'MarkerFaceColor', 'y',
    'MarkerSize', 7)
21. hold off;
22. end
```