5-2018

A Trust Management Framework for Decision Support Systems

Yefeng Ruan
Purdue University

A TRUST MANAGEMENT FRAMEWORK FOR DECISION SUPPORT SYSTEMS

A Dissertation

Submitted to the Faculty

of

Purdue University

by

Yefeng Ruan

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

May 2018

Purdue University

West Lafayette, Indiana

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF DISSERTATION APPROVAL

Dr. Arjan Durresi, co-Chair

Department of Computer and Information Science

Dr. Ninghui Li, co-Chair

Department of Computer science

Dr. Elisa Bertino

Department of Computer science

Dr. Xukai Zou

Department of Computer and Information science

Dr. Xia Ning

Department of Computer and Information science

Approved by:

Dr. Voicu Popescu by Dr. William J. Gorman

Head of Department Graduate Program

Dedicated to my parents.

ACKNOWLEDGMENTS

I would like to take this opportunity to express my gratitude to my advisor Prof. Arjan Durresi for his invaluable guidance and unconditional support. Prof. Durresi is a professional, thoughtful and disciplined researcher. His encouragement, patience, and dedication helped me a lot when I have difficulties. Without his guidance and support, I cannot develop my research work smoothly. Besides in research work, he also gave me many valuable suggestions in my life. I enjoyed the time working with him.

Also, I want to express my appreciation to all my committee members, Prof. Ninghui Li, Prof. Elisa Bertino, Prof. Xukai Zou and Prof. Xia Ning. They were willing to serve in my committee and gave me valuable guidance in my research work. I also want to thank Prof. Raj Jain at Washington University in St.Louis for his help in our discussions. I want to thank my lab mates, Lina Alfantoukh, Murat Karakus, Pawat Chomphoosang and Suleyman Uslu, for their kind help and inspiring discussions.

Finally, I want to express my special acknowledgements to my parents for their unconditional love and support for me.

TABLE OF CONTENTS

Page

Page

LIST OF TABLES

LIST OF FIGURES

SYMBOLS

T	trust
m	trustworthiness
c	confidence
r	error
σ	forgetting factor

ABSTRACT

Ruan, Yefeng Ph.D., Purdue University, May 2018. A Trust Management Framework for Decision Support Systems. Major Professor: Arjan Durresi.

In the era of information explosion, it is critical to develop a framework which can extract useful information and help people to make "educated" decisions. In our lives, whether we are aware of it, trust has turned out to be very helpful for us to make decisions. At the same time, cognitive trust, especially in large systems, such as Facebook, Twitter, and so on, needs support from computer systems. Therefore, we need a framework that can effectively, but also intuitively, let people express their trust, and enable the system to automatically and securely summarize the massive amounts of trust information, so that a user of the system can make "educated" decisions, or at least not blind decisions.

Inspired by the similarities between human trust and physical measurements, this dissertation proposes a measurement theory based trust management framework. It consists of three phases: trust modeling, trust inference, and decision making. Instead of proposing specific trust inference formulas, this dissertation proposes a fundamental framework which is flexible and can be adapted by many different inference formulas. Validation experiments are done on two data sets: the Epinions.com data set and the Twitter data set.

This dissertation also adapts the measurement theory based trust management framework for two decision support applications. In the first application, the real stock market data is used as ground truth for the measurement theory based trust management framework. Basically, the correlation between the sentiment expressed on Twitter and stock market data is measured. Compared with existing works which do not differentiate tweets' authors, this dissertation analyzes trust among stock

investors on Twitter and uses the trust network to differentiate tweets' authors. The results show that by using the measurement theory based trust framework, Twitter sentiment valence is able to reflect abnormal stock returns better than treating all the authors as equally important or weighting them by their number of followers.

In the second application, the measurement theory based trust management framework is used to help to detect and prevent from being attacked in cloud computing scenarios. In this application, each single flow is treated as a measurement. The simulation results show that the measurement theory based trust management framework is able to provide guidance for cloud administrators and customers to make decisions, e.g. migrating tasks from suspect nodes to trustworthy nodes, dynamically allocating resources according to trust information, and managing the trade-off between the degree of redundancy and the cost of resources.

1 INTRODUCTION

In our lives, all our social interactions are based on trust. On one hand, trust is an accumulated feeling based on the past social interactions. On the other hand, trust can help us to make future decisions that guide our future social interactions. For example, when people interact with others, they evaluate and/or update others' trustworthiness based on the interactions. At the same time, people make decisions based on trust assessment results they have, especially for those cases which are involved with high risk, i.e. stock investment, healthy decisions. Therefore, trust is an indispensable factor of many decision support systems.

1.1 Problem Statement

Although we know that trust plays an important role in decision making processes, there still exists some challenges in this field. First of all, trust itself is a subjective and complicated concept. Depending on circumstances and applications, trust has many different interpretations, and consequently, different representations and management principles [1]. Therefore, given various types of raw input data, we need to design a framework which is able to convert or map from raw input data into trust information. Besides this, for many large systems, it is very difficult for users or agents to manually evaluate trustworthiness like in our real lives, where we only have a limited number of acquaintances to deal with. Therefore, we need a computer framework which is able to efficiently handle trust information in a computerized way.

Secondly, it is very common that large systems are sparsely connected in real applications [2,3]. This is partly because that we can only evaluate targets with whom we have direct interactions/measurements. Given such sparsely connected systems or graphs, one way to alleviate this is to use the existing direct trust relationships to

infer indirect trust relationships for users who originally are not directly connected, which is also called "Friend of Friend (FOAF)" [2, 4].

Finally, given both direct and indirect trust information, our purpose is to provide additional information to help people make decisions in various applications. Decision making itself is a well developed and complex discipline. In this dissertation, we mainly focus on how trust can be used to help people or agents make better decisions compared with the scenarios where trust information is not available.

1.2 Dissertation Statement

To address the existing challenges, this dissertation presents a measurement theory based trust management framework. It simulates the trust evaluation process following the measurement theory. It is very flexible and can be adapted by various types of trust inference formulas.

We call the first challenge trust modeling. Basically, it maps the available trust related raw data from the field into computerized trust metrics which are defined in our trust management framework. Inspired by the similarities between the trust assessment process and physical measurements, this dissertation proposes a new trust metric, composed of trustworthiness and confidence, which captures both trustworthiness and its certainty.

We call the second challenge trust inference. It is composed of two types of trust inference operations: trust transitivity and trust aggregation [5, 6]. In our trust management framework, based on the error propagation theory, we are able to calculate confidence for inferred trust according to different trust transitivity and aggregation formulas as long as they are derivative.

We call the third challenge decision making. This dissertation presents two applications as examples to illustrate how we can use the trust information derived by our trust management framework to help making decisions. In the first application example, we use the real stock market data as ground truth. We derive Twitter users'

influence based on the trust network. Our results show that by taking into account trust information, we are able to enlarge the correlation between Twitter sentiment valence and the real stock market data. In the second application, we use our trust management framework to help to detect and prevent from being attacked in cloud computing scenarios. The simulation results show that it is able to provide guidance for the cloud administrators and customers to make decisions, and manage the trade-off between the degree of redundancy and the cost of resources.

We represent our trust management framework in Figure 1.1. The above three challenges are divided into three phases in Figure 1.1. All three phases of trust processing are dependent on the context, especially Trust Modeling and Decision Making [7]. For example, depending on the type of decisions, or the risks involved, we would map appropriately the raw trust data into defined trust metrics. Similarly, depending on the context, such as risks, we might use different formulas to aggregate and filter trust in Trust Inference. Finally, in Decision Making, for example, we might apply different levels of trust thresholds when we select a doctor for an important surgery, compared with when we select a movie. Furthermore, the three phases are interrelated. The accuracy of our Trust Inference, and its corresponding level of support in Decision Making, will depend on the availability and granularity of trust data from the field. While Trust Modeling and Decision Making can place constraints on the context, such as limitations from the raw data or the type of decisions, Trust Inference should not limit the potential of the raw data, but potentially increase it, by leading to more trustworthy decisions.

1.3 Dissertation Organization

The outline of the dissertation and a brief overview of the chapters are presented in this section.

In Chapter 2, we provide a literature review of existing trust management frameworks for online social communities. We also list four commonly seen types of attacks

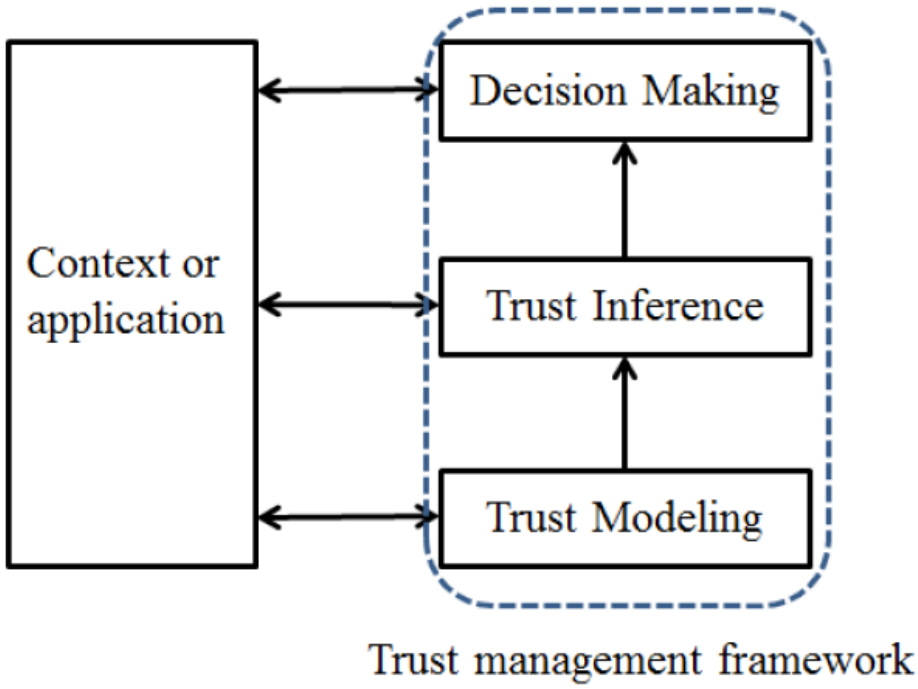


Figure 1.1. Three phases of trust management framework

in this field, and analyze existing frameworks' vulnerabilities to these four types of attacks. Specifically, this survey focuses on trust modeling, trust inference, and attacks in this field.

In Chapter 3, we propose a trust management framework based on measurement theory. Furthermore, based on the error propagation theory, we propose a method to compute confidence for inferred confidence according to different trust transitivity and aggregation formulas. We perform experiments on two real data sets, Epinions.com data set and Twitter data set, to validate our trust management framework. Also, we show that inferring indirect trust can connect more pairs of users.

In Chapter 4, we use the real stock market data as ground truth for our trust management framework. We apply our trust management framework to build a user-to-user trust network for Twitter users. Based on the user-to-user trust network, we measure Twitter users' influence in the field of stock investment. Our results show

that trust network based reputation mechanism can amplify the correlation between a specific firm's Twitter sentiment valence and the firm's stock abnormal returns.

In Chapter 5, we apply our trust management framework to help cloud vendors and customers to detect and prevent from being affected by potential attacks. We show that our trust management framework is able to provide guidance for the administrators to make decisions, e.g. migrating tasks from suspect nodes to trustworthy nodes, dynamically allocating resources, and managing the trade-off between the degree of redundancy and the cost of resources. In addition, it can be used to calculate systems' reliability based on the real-time trust information.

In Chapter 6, we conclude this dissertation and provide directions for future work.

2 A SURVEY OF TRUST MANAGEMENT FRAMEWORKS FOR ONLINE SOCIAL COMMUNITIES

2.1 Introduction

Due to the development of the Internet and computer-based devices, especially smart phones, people are now moving at least part of their social activities to online environments. In the last few years, many online social networks, such as Facebook and Twitter, have spread out around the world. Participants in such kinds of social networks can have a large number of claimed friends. Some of them may be well known, while some are not. One possible way to deal with this problem is to differentiate them by using trust metrics. In [8], authors differentiated "claimed friends" from "real friends" on Twitter by counting the number of interactive tweets that two users post toward each other. Besides social networks, many other online applications also exhibit social properties, for example e-commerce [9, 10], like eBay [11], Amazon and Epinions [12], and Peer-to-Peer file sharing networks [13, 14]. Here, we call them online social communities in which participants can be users, agents, devices, or others.

We have seen that trust plays an extremely important role in online social communities, as well as in people's lives; however, there are some challenges in applying trust in online social communities [15]. First of all, we have to represent trust in a computational model. Trust is not easy to model in a computational way because of its subjective property [1]. Also, it cannot be applied directly in online social communities due to different features that online social communities have from traditional social networks [16]. For example in real life, people only have a limited number of friends to evaluate, but this number explodes in online social communities. On Facebook and Twitter, users can have thousands of friends. Apart from this, in real life,

trust is developed slowly over time, based on face-to-face social experiences; however, this is very difficult in online social communities due to the large number of potential friends. Therefore, trust in online social communities must be computational such that it can be processed by computers [1, 16]. The difficulty is that trust is a subjective concept, and it has different meanings in different fields and applications [17, 18]. For example, in Amazon, participants use stars to represent to what extent they think others' reviews are useful. While in other cases, such as in Peer-to-Peer networks, trust measures the quality of downloaded files, downloading speed, and so on [13, 19]. Therefore, trust modeling should be dependent on applications or scenarios. In the remainder of this work, we use the term trust modeling to denote how to represent trust in a computational way.

Besides trust modeling, another challenge is how to infer indirect trust information among two unconnected participants. In many online communities, only a small number of participants are directly connected, compared with the potential number of pairs of participants. Many works have shown that online communities are sparsely connected [8, 16, 20–22]. Therefore, it is urgent to introduce mechanisms that can be used to infer indirect trust among participants who are not directly connected. Such type of framework is described as "Friend of a Friend (FOAF)". Basically, trust propagates along chains; however, how to propagate trust is still an open debate. Both general and application specific mechanisms are proposed by many researchers in this field [4, 23–31].

As shown in Figure 1.1, in this work we use the term trust management frameworks to denote the schemes dealing with how to represent, infer, and use trust. We provide a survey for existing trust management frameworks used in various online social communities. We mainly focus on two challenges – trust modeling and trust inference. Although there are several survey papers about computational trust [32–34] and global trust/reputation related attacks [35–37], the main contribution of this chapter includes:

- We provide a survey for trust inference problem, which takes into account inferring indirect trust relationship for not directly connected participants.
- We provide a survey for four types of local trust related attacks, and analyze existing schemes' vulnerabilities to them.

The rest of this chapter is organized as follows: in Section 2.2, we investigate various definitions of trust, and introduce some related works. In Section 2.3, we review how existing schemes deal with the first challenge – trust modeling. In Section 2.4, we illustrate the second challenge – trust inference, and survey several existing schemes in this field. In Section 2.5, we illustrate four types of attacks existed in trust management frameworks. In Section 2.6, we analyze existing schemes' vulnerabilities to the four types of attacks. In Section 2.7, we conclude this chapter.

2.2 Background and Related Works

2.2.1 Definition of Trust

Trust is a relationship existing between two participants. In this chapter, we use truster and trustee to denote them. Trustee is the participant being evaluated by the truster. For example, when we say A trusts B, A is the truster and B is the trustee.

Trust is studied and used in a number of disciplines, such as sociology, psychology, economics, computer science, and so on. As a result, there are many definitions for trust and no general consensus has been achieved so far [38,39]. Among them, one of the recent summarized definition is given by [39]:

> "Trust is the willingness of the trustor (evaluator) to take risk based on a subjective belief that a trustee (evaluatee) will exhibit reliable behavior to maximize the trustor's interest under uncertainty (e.g., ambiguity due to conflicting evidence and/or ignorance caused by complete lack of evidence) of a given situation based on the cognitive assessment of past experience with the trustee." [39].

In this definition, trust is explained as the probability of performing a specific action. In the field of computer science, besides probability, there are many other representations of trust, such as entropy [40, 41], similarity [42–44], and so on. We will see different types of representations of trust in the following.

Trust can be classified based on various criteria. In [45], McKnight classified it into three categories: impersonal/structural trust, dispositional trust, and personal/interpersonal trust. Impersonal/structural trust is determined by institutional properties rather than by participants themselves. Dispositional trust represents participants' bias trust preferences. Personal/interpersonal is the participant-to-participant trust relationship. Among them, personal/interpersonal trust has attracted ample attention from researchers. In this chapter, we mainly focus on personal/interpersonal trust. For simplicity, we call it trust in the following. Trust can be further divided into functional trust and recommender trust based on the types of behaviors [46]. Functional trust describes how trustworthy a person is when implementing functions, e.g. how good Alice is as a doctor. Recommender trust measures how reliable a person's recommendations are, e.g. how reliable Alice's recommendations are about doctors. The reason why some trust management frameworks separate them is that recommender trust is explicitly useful for trust inference.

Trust has many properties, such as subjective, dynamic, asymmetric, context dependent, transitive, composable, and so on [1, 17, 32]. Similar to its definition, different applications highlight different aspects of its properties. Here we list some properties that are very common in online social communities.

- Subjective. For the same trustee, different trusters can have different trust evaluations, even given the same observations [1, 32]. Also, the same trust level may have different meanings for different trusters. For example, A may think 80% as very trustworthy, while B may consider it as only a little bit better than neutral.

- Asymmetric. As trust is subjective, it is also asymmetric [1,17,34]. A trusts B, that does not necessarily mean B will trust A. Therefore, when representing online social communities in graph models, their links are directed.

- Context dependent. Usually the truster trust the trustee for specific domains [17,34]. For example, people think that computer scientists are professionals in computer networks, but not necessarily are they reliable in medicine.

- Dynamic. Trust is developed over time. In people's lives, good or positive experiences will increase trust levels, while trust levels will be decreased by bad or negative experiences [47]. There are some works proposing that it should take a large number of positive evidence to build up trust while a few negative evidence can destroy the trust immediately [48]. Apart from this, the effects of experiences also diminish over time. A new experience is more important for the truster to evaluate the trustee than old experiences [13,49].

- Propagative/transitive. Propagation means that trust can propagate along a chain [17,34]. For example, A trusts B, and B also trusts C. To some extent, A will also trust C, although A does not know C directly. This property is fundamental for trust inference that we are going to introduce in the following sections.

- Composable. Besides transitivity, trust is also composable [17]. Giving the truster multiple trust paths to evaluate the trustee, she/he should be able to combine all the information. Again there are also many schemes about how to combine the information.

2.2.2 Trust Management Frameworks

Trust management frameworks are designed to help participants to make better decisions based on trust information. Jøsang defined it as follows:

> "The activity of creating systems and methods that allow relying parties to make assessments and decisions regarding the dependability of potential transactions involving risk, and that also allow players and system owners to increase and correctly represent the reliability of themselves and their systems." [50]

According to [34], trust management frameworks can be divided into three phases: trust modeling, trust management, and decision making. Trust modeling mainly deals with how to represent trust relationships in computational models, and trust management is used to describe how to collect evidence and to do risk evaluation. Decision making is another important and complicated field, and can even be treated separately [34]. As trust modeling and trust management, together, mainly deal with how to represent trust in computational models using available raw data, we incorporate them together and use trust modeling to represent them. Apart from them, we also include trust inference into trust management frameworks as it is a very important component for trust management frameworks to work more intelligently and efficiently. Trust inference uses direct trust information among participants to infer indirect trust information. In this chapter, we mainly focus on trust modeling and trust inference.

We represent trust management frameworks in Figure 1.1. All three phases are dependent on context or applications, especially trust modeling and decision making. For example, depending on the type of available raw data, systems would map appropriately the raw data into defined trust metrics. Similarly, depending on context, such as risk, systems might use different methods to aggregate and filter trust, in trust inference. Finally, in decision making, for example, systems might apply different levels of trust thresholds when participants select a doctor for an important surgery, compared with when they decide whether or not to watch a movie. Furthermore, the three above phases are interrelated. So, the accuracy of trust inference, and its corresponding level of support in decision making will depend on the availability and granularity of raw trust data from the field.

2.2.3 Related Works

As online social communities are becoming more popular, there are also more works investigating trust relationships in this field of computer science. As a result, there are several survey papers in this field.

In [32], authors provided a survey for computational trust and reputation models. It also discussed their properties. In [33], authors provided a survey for trust in the field of E-commerce from economists' points of view. In [34], authors mainly focused on the classification of trust modeling. It reviewed how trust is represented and what is the semantic meaning of trust in different systems, e.g. rating, probability, fuzzy logic, etc. Jøsang provided a survey for trust's categories and semantic meanings in [11]. Besides trust, he also investigated another trust related concept – reputation. Furthermore, he gave some application examples in the paper, such as Amazon, Epinions, and Slashdot. Golbeck provided a comprehensive survey on trust modeling in [18]. It classifies trust based on its objects. Massa reviewed some challenges in trust management frameworks in [51]. It included how to represent trust in various types of online systems. Also, it mentioned a few identity related attacks, such as fake identities and multiple identities. In [17], authors provided a survey for trust in web-based social networks. It showed how trust is defined in different disciplines and also gave its definition for web-based social networks. It mainly focused on data collection, trust evaluation and trust dissemination. In [39], authors provided a survey for trust modeling in complex, composite networks. It included four layers of trust: communication trust, information trust, social trust and cognitive trust. It reviewed trust from multiple disciplines' points of view, such as sociology and psychology.

There are also a few works discussing the relationship between trust and security. In [35], authors discussed the concept of trust in the field of computer security. It mainly focused on determining initial trust metrics and updating trust metrics based on observed behaviors. It also described how trust can be used in computer security applications, such as authentication, intrusion detection, and so on. Similar to [35],

authors of [36] also discussed the potential usage of trust in E-commerce to counter attacks. In [52], authors combined social trust and quality-of-service trust for Mobile Ad Hoc Networks (MANETs). It also investigated several potential attacks; however, attacks discussed are application-oriented. They are specifically related to MANETs, such as routing loop attacks, replay attacks, and so on. In [53], authors listed several requirements of different security problems and potential attacks in trust management frameworks, but without examining existing schemes' vulnerabilities. In [37], Hoffman et al. provided a survey about potential attacks and defense techniques in reputation (global trust) systems. While in this chapter, we focus on attacks related with local trust.

Although there are several surveys existing for trust management frameworks [32–34], they rarely investigated trust inference. Many of them considered that trust management frameworks can be used to detect malicious users, but without considering trust management frameworks themselves can be the targets of attacks. Some surveys only considered attacks in specific applications or environments, such as [35], [36] and [52]. Therefore, in this chapter we provide a comprehensive survey for trust management frameworks for online social communities, which consider both trust inference and potential attacks.

2.3 Trust Modeling

In this section, we review how existing works deal with the first challenge we mentioned – trust modeling. As indicated in [34], trust modeling deals with how to represent trust in computational models using available raw data. In details, it includes the metrics they used to represent trust, how many dimensions they have, what is the trust information source, and what are the semantic meanings of trust.

2.3.1 Trust Metrics

Trust Scaling

As we stated, in order for computers to be able to process trust, it must be represented in a computational way. Metrics can be either numerical or categorical. Trust is always represented by numerical values. In the literature, there are two types of numerical values, discrete and continuous values, used to quantitatively measure trust. Discrete values come from raw data, such as ratings, scaled metrics, and so on [46]. Continuous values are also often used in trust management frameworks. For example, probability based, or similarity based trust metrics [13, 42], are always continuous. Besides numerical values, trust can also be represented by intervals [54, 55].

- Binary discrete values. One of the most straightforward ways for the truster to express her/his opinion about the trustee is to use binary metrics – trust or distrust. In many applications, it is also the final goal for the truster to make a binary decision. There is a large number of research works that model trust relations using binary metrics [56–59].

- Multinomial discrete values. Although binary metrics are easy for participants to use and understand, in some cases, trust and distrust may not be sufficient to represent the truster's opinions. With more scaled metrics, like "very trust", "trust", "distrust" and "very distrust", participants can evaluate others more accurately because they have more options [39]. Scaled metrics are commonly used in questionnaires. They can be converted to discrete values which can be used in computational models [46, 60].

- Continuous value. Continuous value is another popular way to represent trust. Due to the semantic meaning of many applications, such as probability and belief, continuous value is a straightforward way to represent trust. Many works belong to this category [16, 26, 61–64].

- Interval. Instead of representing trust using a single value, some works use intervals to represent trust, as in many cases trust is uncertain. Interval is used by many fuzzy logic-based trust models. Examples include [54] and [55].

Trust Dimension

In many works [4, 13, 24, 29, 59, 65], trust is represented by a single value; however, as trust has many properties, in some cases, two or more parameters are used to represent trust. In this section, we use the term trust dimension to denote the number of parameters that are used.

- Separated distrust. In systems that use a single trust value, distrust is considered as the complement of trust. In these systems, high value represents trustworthy, while low value represents untrustworthy [16, 24]. However, this is not always true [39]. [2] and [56] and separate distrust from trust and treat them independently. Besides distrust, [66] introduces untrust and mistrust into the system.

- Time stamps. As trust is dynamic, it is important for researchers to consider time stamps for trust status. By incorporating time stamps, trust can be updated and used to defend certain attacks [62]. Example considering time stamps include [30], [62], [67] and [68].

- Context. Trust is context dependent [39]. The trustee may exhibit different trust degrees or trustworthiness given various types of contexts. For example, a good babysitter is not necessary a good car repairer. Therefore, many works are context-aware [30, 46, 63, 69–71].

- Confidence/certainty. Confidence or certainty is used in trust management frameworks to measure to what extent the truster is certain about her/his trust assessment. It is considered as an important additional metric in many trust

management frameworks [2,16,72–74]. Therefore, we illustrate it in more details in Section 2.3.3.

Furthermore, there are many works that include other dimensions. For example, Subjective Logic [75] uses relative atomicity to denote the percentage of uncertainty contributing to the expected belief.

Trust Source

According to [17], trust can be derived from three sources: attitude, experience and behavior.

- Explicit attitude. Attitude represents the truster's opinion towards the trustee. It can be either trust/like/positive or distrust/dislike/negative. Although [17] indicates that attitude can be derived from interactions or experiences, in this chapter we only consider explicit attitude information. For example, in Epinions.com, users express either trust or distrust attitude [12]. Also, for those systems assuming trust values are directly and explicitly available, such as [2, 27, 56, 58, 59], we consider them as using explicit attitude.

- Evidence/feedback/experience. When the truster interacts or makes transactions with the trustee, the truster is able to evaluate the trustee's performance. For example, satisfactory transactions and unsatisfactory transactions are used to measure trust in [13]. Evidence is used in systems which consider belief theory [26, 74–76]. Also, rating is widely used in many systems to calculate trust [4, 16, 25, 77]. Note that although these systems all use evidence/feedback/experience, their semantic meanings are different. We illustrate this in Section 2.3.2.

- Behavior. Trust can also be evaluated based on behaviors [17]. In [78] and [79], authors used reply, forward and retweet behaviors to capture trust information. [80] also uses communication behaviors to measure trust. Besides these

application specific behaviors, we also consider similarity as one of the behaviors. Similarity measure how similar two agents are, for example, their purchasing behaviors [81,82], common communities they join [83], profile similarity [64,84], and so on.

2.3.2 Semantic Meanings of Trust

Trust has different semantic meanings in different scenarios and applications. We discuss some existing schemes' semantic meanings of trust in the following. Also, we note that some systems, such as [24, 56, 85], simply assume trust values are extant without digging into their semantic meanings.

Evidence or Experience Based Trust

In many cases, participants build up trust based on their prior evidence or experiences. The truster assesses all experiences she/he had with the trustee. Given those assessments of evidence, there are still multiple methods to model trust.

- Probability. As pointed out by [86], trust can be expressed by the probability that the trustee will behave as the truster expects. One of the most popular theories used in trust management frameworks is Dempster-Shafer Theory (DST). Based on DST, Jøsang et al. proposed a model which takes binary evidence as input and computes trust values [75], as well as many other researchers [26, 74, 76, 87, 88]. In [40], Sun et al. calculated trust based on probability's entropy. Probability based trust model is a very popular scheme in trust management frameworks.

- Mean. It is a straightforward way to calculate the mean of evidence as the trust value. For example, in [16], Zhang et al. used the average ratings (from 1 star to 5 starts) as trust values.

- Mode. Given the set of evidence, instead of calculating the mean value, an alternative way is to find the mode of the discrete evidence, such as [46].

- Difference. Trust value can also be calculated by the difference between positive and negative evidence. In [13], authors calculated the difference ($\#positive\ evidence - \#negative\ evidence$) first, and then for each participant p they normalized local trust values with the summation of p's all outgoing trust values.

Application Specific Behavior Based Trust

When calculating trust, some specific types of behaviors are especially important. Here we distinguish behaviors from evidence although evidence can be considered as one specific type of behavior.

- Conversational behaviors. For example, authors of [79] considered on Twitter that conversation and forwarding are two factors to determine trust. If two participants have balanced long term conversations, most likely they trust each other. Similarly, they assumed that if the truster forwards the trustee's messages very frequently, it means that the truster trusts the trustee. In [89], authors considered retweet and favorite as two trust-related behaviors on Twitter. Also, authors of [80] used conversational behaviors, such as conversation duration and frequency, to measure trust.

Similarity Based Trust

Similarity was first used in collaborative filtering (CF) recommender systems. They make recommendations based on the similarity between participants or items [81, 90]. Similarity can be an additional metric in trust management frameworks in determining trust [9, 63, 82, 84]. The assumption is that, for participants who are similar with each other, most likely they also trust each other.

Reputation

Reputation or global trust is different from local trust. We consider it as one type of trust, as in many systems it is considered in the decision making stage. Reputation is widely used in many systems, such as e-commerce systems [91] and Peer-to-Peer networks [92, 93]. Instead of asserting trust metrics for each pair of participants, each participant only has a unique value which represents how the whole community (centralized [87]) or part of the community (distributed [69, 94]) evaluates this participant. Furthermore, it can affect agents' personal/interpersonal trust. Examples of reputation systems include [65, 69, 92–100].

Fuzzy logic based trust

Because of trust's nondeterministic property, many works adopted fuzzy logic to model trust. Unlike traditional logic metrics, fuzzy logic is among completely true and completely false [101]. Schemes using fuzzy logic to represent trust include [102–105].

Comprehensive trust

Trust is a summarization of complicated human behaviors, and it can be affected by many factors. Because of its human-related properties, besides the above mentioned factors, some researchers tried to take into account several other factors when computing trust values [106]. For example, Marsh defined trust from the disciplines of psychology, sociology, biology and economics, and stated many rational principles and rules, which are adopted by later works [1]. In [64], Zhan and Fang concluded that trust is dependent on three components: profile similarity, information reliability and social opinions. Besides direct connections, authors of [67] also took into account users' susceptibility and others' contagious influence. In [83], friendship, social contact (based on frequently visited locations) and community of interest contribute to the trust. In [30], trust is divided into interpersonal trust and impersonal trust. It

further considers four aspects (benevolence, competence, integrity and predictability) for interpersonal trust. ReputationPro [70] uses a tree-like structure to compute trust. In [107], authors calculated trust from four aspects: prestige, familiarity, similarity and risk of trust.

From the above description, we can see that there exist several different representations and meanings for trust depending on specific scenarios and applications. It is very difficult to say which one is the best, or which one can outperform another, as the validation is also application dependent.

2.3.3 Trust and Confidence/Certainty

With the development of trust management frameworks, many researchers found that trust value itself is not enough to manage trust relationships. In many schemes [16, 46, 72, 87], researchers introduced another important concept – confidence (or certainty) into trust management frameworks. Confidence is used to measure how certain the truster is about her/his trust views about the trustee.

By using confidence, the truster can distinguish distrusted participants from unknown participants. Participants can have different levels of confidence even though they have the same level of trust. For instance, although both distrusted participants and unknown participants have very low trust levels, the confidence is different. Typically, distrusted participants have very high confidence due to their previous bad behaviors. While unknown participants have very low confidence since they are new in the communities.

Another important role of confidence is to imply the number of evidence or experiences based on which trust is evaluated. Confidence will increase as the total number of evidence increases. Confidence is also an important factor in the decision making stage. For example, when we are faced with high risk events, we may choose to cooperate with participants that have both high trust levels and high confidence level.

Like trust, there are also several ways to represent confidence. In [75], Jøsang et al. used a multi-tuple to represent belief, disbelief and uncertainty, which sum up to 1.0. In this case, uncertainty is dependent on belief and disbelief. Confidence in [16] is determined based on the uncertainty in measurement theory. There are also several works that use similarity based confidence [25, 108], as well as fuzzy theory [109].

In summary, in Table 2.1 we list some schemes and their corresponding representations, semantic meanings, as well as trust dimensions for trust. Also, we examine each scheme to see if they support trust inference (properties of transitive and composable).

2.4 Trust Inference

The goal of trust management frameworks is to provide participants with trust information and help them to make decisions; however, in many online communities, only a limited number of participants are directly connected. Therefore, using existing direct trust is not sufficient. It is urgent to introduce trust management frameworks which can infer indirect trust by making use of direct trust links [57]. In the field of computer science, there are many proposed trust inference schemes. Some of them were designed for specific applications, while some were proposed for general purposes. We review some existing schemes in this section.

There exist two very important operators in the trust inference schemes: transitivity/concatenation operator and aggregation operator [75, 110]. Transitivity operator is used to calculate trust propagation in a single chain. It helps participants to evaluate others even though they do not have any prior direct experiences. Aggregation operator is used for combining parallel trust paths between the truster and the trustee in case that there exist more than one trust path between them.

In the following, we classify some existing schemes based on the methods they used to calculate trust transitivity and aggregation. We list the methods they used to

Table 2.1.
Representations, semantic meanings and properties of trust

Schemes	Trust scaling	Semantic meaning	Trust dimension	Trust inference
Marsh [1]	Continuous $[-1, 1)$	Comprehensive	CT, TS	No
Abdul-Rahman and Hailes [46]	Discrete (multinomial)	Evidence based (mode)	CT, CF	Yes
Jøsang [75]	Continuous $[0, 1]$	Evidence (probability)	DT, CF	Yes
Falcone, Pezzulo et al. [102]	Continuous $[-1, 1]$	Fuzzy logic	NA	NA
Kamvar, Schlosser et al. [13]	Continuous $[0, 1]$	Evidence (difference)	NA	Yes
Guha and Kumar [56]	Discrete (binary)	NA	DT	Yes
Xiong and Liu [69]	Continuous $[0, 1]$	Reputation	CT, CF	Yes
Golbeck [4]	Continuous $[0, 1]$	NA	CT	Yes
Sun, Yu et al. [40]	Continuous $[-1, 1]$	Evidence (entropy)	TS, CT	Yes
Massa and Avesani [24]	Continuous $[0, 1]$	NA	NA	Yes
Sabestian [76]	Continuous $[0, 1]$	Evidence (probability)	CT, CF	Yes
Wang and Singh [72]	Continuous $[0, 1]$	Evidence (probability)	DT, CF	Yes
Uddin, Zulkernine et al. [63]	Continuous $[0, 1]$	Similarity	TS, CT, CF	Yes
Adali, Escriva et al. [80]	Continuous $[0, 1]$	Behavior	TS	Yes
Leskkovec, Huttenlocher et al. [58]	Discrete (binary)	NA	NA	NA
Nepal, Sherchan et al. [68]	Continuous $[0, 1]$	Comprehensive	TS, CT	No
Victor, Cornelis et al. [2]	Continuous $[0, 1]$	NA	DT, CF	Yes
Zhan and Fang [64]	Continuous $[0, 1]$	Comprehensive	NA	No
Liu, Wang et al. [85]	Continuous $[0, 1]$	Comprehensive	NA	Yes
Wang and Wu [73]	Continuous $[0, 1]$	Evidence (probability)	CF	Yes
O'Doherty, Jouili et al. [29]	Continuous	Comprehensive	NA	No
Zhang and Durresi [16]	Continuous $[0, 1]$	Evidence (mean)	CF	Yes
Kant and Bharadwaj [103]	Continuous $[0, 1]$	Fuzzy logic	DT	Yes
Fang, Zhang et al. [67]	Continuous $[0, 1]$	Comprehensive	TS,CT	Yes
Chen, Guo et al. [83]	Continuous $[0, 1]$	Comprehensive	NA	Yes
Shakeri and Bafghi [55]	Interval	Evidence	CF	Yes
Liu, Yang et al. [74]	Continuous $[0, 1]$	Evidence (probability)	DT, CF	Yes
Zhang and Mao [59]	Discrete (binary)	NA	NA	Yes
Aref and Tran [105]	Continuous	Fuzzy logic	TS	No
Fang, Guo et al. [30]	Continuous $[0, 1]$	Comprehensive	TS, CT	Yes

For trust dimension, DT=Separated Distrust, TS=Time stamp, CT=Context, CF=Confidence/certainty, NA=Not available.

propagate and aggregate trust separately; however, trust transitivity and aggregation in some schemes, such as the matrix factorization category, are combined together.

2.4.1 Multiplication for Transitivity and Weighted Mean of Evidence for Aggregation

Abdul-Rahman and Hailes

In [46], Abdul-Rahman and Hailes divided trust into two categories: direct trust and recommender trust. Trust in this case has four discrete values: very trustworthy, trustworthy, untrustworthy and very untrustworthy. The truster maintains a set of prior experiences with the trustee. To determine the trust value, it returns the mode of four trust degrees. If there are more than one returned trust degrees, it assigns a uncertainty value. Furthermore, trust propagates through recommendations. The truster compares her/his own experiences with the recommender's suggestions and then adjusts the recommender trust accordingly. Experiences are aggregated by weighted mean, where weights are intermediary participants' recommender trust. Similarly, aggregated trust is the mode of four trust degrees.

Jøsang

In [75], Jøsang proposed a model called Subjective Logic that considers trust as a term of uncertain probability. Trust is represented in two spaces – opinion (or belief) space and evidence space. Following Dempster-Shafer Theory (DST), Jøsang defined four important parameters: belief (b), disbelief (d), uncertainty (u) and relative atomicity (a) in the opinion space, and $b + d + u = 1$. In the evidence space, it focuses on binary events: positive evidence (represented by r) and negative evidence (represented by s). The posterior probability of binary events is represented by Beta distribution. Furthermore, there exists a mapping between the evidence space and the opinion space.

It uses discounting and consensus operators to propagate and aggregate trust correspondingly. Intermediary participants' recommendations about the trustee are discounted by their trustworthiness. In trust transitivity, both belief and disbelief

decrease, while uncertainty increases. This makes sense in real life that uncertainty increases when introducing more intermediary participants within a chain. Consensus operator adds evidence together from multiple parallel trust paths and converts them into the opinion space. Jøsang extended [75] to a new version, which uses conditional belief reasoning in [111]. As we will see later, Subjective Logic is adopted by many other researchers in this field.

Sabestian

In Sabestian's model, which is called CertainTrust [76], trust is also represented in two spaces. Human Trust Interface (HTI) contains trust and certainty. The second representation focuses on the evidence domain. To determine certainty, it sets a maximal number (E_m) of expected evidence for each context. Certainty increases when evidence increases; however, it does not increase linearly. In the beginning, few evidence can make certainty increase a lot. While there are already a large amount of evidence, certainty increases not as fast as before. When the number of evidence is greater than or equal to E_m, certainty is normalized to 1. In the evidence domain, similar to [75], it also uses Beta distribution to model the posterior probability of binary events. There exists a map between the above two representations. Trust is equal to the mode of Beta distribution.

For trust transitivity and aggregation, two operators – consensus and discounting, are defined. Both of the operators first calculate in the evidence domain and then convert to HTI.

Wang and Singh

As in [75], Wang and Singh also represented trust in the evidence space and the belief space; however they defined certainty differently in [72]. It has an another important parameter – evidence conflict, which represents the ratio of positive evidence

to the total evidence. In this definition, certainty is dependent on both the conflict ratio and the number of evidence.

Operators for trust transitivity and aggregation are similar to [75]. Apart from transitivity and aggregation operators, authors added another operator – selection, in [112]. Selection operator is used to select reliable trust paths among multiple trust paths between the truster and the trustee.

Liu, Yang et al.

Apart from belief, disbelief and uncertainty used in Subjective Logic, ASSESS-TRUST [74] incorporates another metric: posterior uncertainty. Relatively, it calls uncertainty defined in Subjective Logic the prior uncertainty. In the evidence space, it includes three types of evidence: positive, neutral and negative evidence. Mapping exists between the opinion space and the evidence space using Dirichlet distribution. Aggregation operator has the same idea as in Subjective Logic, except for extending from binary evidence to tri-nary evidence. In the transitivity operator, instead of transferring evidence to the prior uncertainty, it transfers evidence to neutral evidence, which in turn increases the posterior uncertainty. In a recursive manner, ASSESS-TRUST calculates trust from the truster to the trustee using transitivity and aggregation operator.

2.4.2 Multiplication for Transitivity and Weighted Mean of Trust Values for Aggregation

Kamvar, Schlosser et al.

EigenTrust [13] is mainly designed for Peer-to-Peer file sharing systems. It measures trust based on the number of satisfied and unsatisfied experiences. The truster's outgoing trust is normalized by the summation of all her/his outgoing links. It

uses multiplication to propagate trust and aggregates trust by weighted mean, where weights are intermediary participants' trust.

Xiong and Liu

PeerTrust [69] is another trust management system designed for Peer-to-Peer networks. It uses reputation based trust metrics. Also, it allows participants to propagate recommendations to their neighbors. It uses weighted mean to aggregate trust; however, weights are dependent on personalized similarity. Similarity is determined by two participants' feedback, number of transactions, credibility of feedback, transaction context factor and community context factor.

Golbeck 2005

In [4], Golbeck proposed a trust management framework – TidalTrust. It uses weighted mean to combine trust from multiple trust paths. In order to improve accuracy, it only takes recommendations from trustworthy neighbors, which means that their trust value is greater than a pre-defined threshold. Also, it sets a limitation for path lengths because Golbeck believed that inferred trust from a long path is not as reliable as that from a short path. It is evaluated in a social network called FilmTrust.

Sun, Yu et al.

Sun et al. proposed a trust model in [40] based on entropy – an important measure of uncertainty in information theory. p denotes the probability that the trustee will perform the action as the truster expected. Trust is defined by the entropy of p. Trust is positive when $p > 0.5$, and it is negative when $p < 0.5$. When $p = 0.5$, trust is equal to 0, which means that the truster is uncertain about the trustee. It uses weighted mean and only considers recommendations from trustworthy intermediary participants (whose trust is positive) to aggregate trust.

Massa and Avesani

In [24], Massa and Avesani proposed a trust model called MoleTrust. It takes two steps to propagate and aggregate trust. In the first step, it takes input the truster, trust network and trust propagation horizon, and outputs a modified trust network. Here the input trust network includes the whole community. Trust propagation horizon limits the maximum number of hops (path length). In the second step, it infers indirect trust within the modified trust network (the outcome of the first step). It computes indirect trust in an iterative way, in which the trustworthiness of a node at distance k only depends on the nodes at distance $k-1$. The inferred trust is the weighted mean of all the accepted incoming links. When selecting incoming links, only those links whose trust is greater than, or equal to, a threshold will be taken into account.

Liu, Wang et al.

In [113], Liu, Wang et al. used the product of links' trust as the prior probability for trust inference in a single path. The posterior probability is adjusted by the Bayesian network. It considers social intimacy degree and recommendation role in the Bayesian network. When there are multiple trust paths between the truster and the trustee, it uses weighted mean to combine them, where weights are assigned according to social intimacy degree and recommendation role and adjusted by the Bayesian network as well. Apart from social intimacy degree and recommendation role, preference similarity is also taken into account in [85].

Zhang and Durresi

In [16], Zhang and Durresi proposed a trust management framework based on measurement theory. It considers social interactions among participants as "measurements". Trust (impression) is similar to "measured value of object", and confi-

dence represents the certainty of a "measurement". So in this model, confidence is related to the "error" in measurement theory. For trust transitivity, there are three principles in [16]. Guided by these principles, it uses multiplication to calculate trust transitivity and weighted mean to calculate trust aggregation. In their work, weights are trust paths' confidence.

2.4.3 Selection for Transitivity and Average for Aggregation

Golbeck 2006

Golbeck proposed two algorithms – Rounding algorithm and Nonrounding algorithm, to infer indirect trust for a binary trust network [57]. Participants in [57] are labeled as either "trusted" or "not trusted". Good participants refer to those agreeing with the truster (source) with a certain probability, while bad participant refers to those who are always opposed to the truster. To infer indirect trust, the truster directly takes her/his good neighbors' recommendations, without discounting them. When there are multiple paths, the truster averages the recommendations. In the Rounding algorithm, all the participants round the average ratings to {0,1} in each step. While in the Non-rounding algorithm, all intermediary participants hold continuous average values, and only the truster does the final rounding.

2.4.4 Matrix Propagation

Guha and Kumar

Guha and Kumar took both trust and distrust into account in their work [56]. It is the first work which considers the propagation of distrust. Compared with trust, propagation of distrust is much more complicated. It defines two matrices, matrix of trust T and matrix of distrust D. Matrix of belief B can have two formats, $B = T$ or $B = T - D$, depending on applications. It includes four atomic propagation operators in this scheme: direct propagation (B), Co-citation ($B^T B$), transpose trust (B^T) and

trust coupling (BB^T). Direct propagation means that if A trusts B and B also trusts C, then A trusts C as well. If $A1$ trusts $B1$ and $B2$, and $A2$ trusts $B1$, it is probable that $A2$ also trusts $B2$ because $A1$ and $A2$ have the same views on $B1$. This is defined as Co-citation. Transpose trust means that if A trusts B, then B may trust A back. In trust coupling, if B and C both trust D, and A trusts B, it implies that A may trust C. These four operators are combined together forming a propagation matrix $C_{(B,a)} = a_1B + a_2B^TB + a_3B^T + a_4BB^T$, where a_1, a_2, a_3, a_4 are the weights of four operators.

There are three models to propagate trust: trust only, one-step distrust and propagated distrust. The trust only model ignores distrust, the one-step distrust model discounts judgments made by distrusted neighbors, and both trust and distrust can be propagated in the propagated distrust model. All three models have a limitation on the chains' length.

Zhang and Mao

In [59], trust is propagated similar to [56]. But it reduces to two atomic operators: transposition and forwarding, as other two (co-citation and coupling) can be deduced from transposition and forwarding. Instead of propagate trust deterministically, it assumes that transposition and forwarding happen with some probabilities. It also assumes that there can be a probability that two random participants can be connected without through transposition and forwarding. Given all the information, the posterior probability of inferred links can be calculated. It uses the factor graph to represent the dependence between variables (links) and functions (probability functions). In such a way it calculates the posterior probability using belief propagation algorithm (also known as sum-product algorithm). Final prediction of binary trust is based on the sorting of the probabilities.

2.4.5 t-norm for Transitivity and Weighted Mean for Aggregation

Victor, Cornelis et al.

In [2], Patricia Victor et al. derived trust from bi-lattices. In the definition, similar to [56], trust includes both trust degree (t) and distrust degree (d), which are independent with each other. It means that even if the trust degree is very high, e.g. $t = 0.9$, distrust degree can also be very high, e.g. $d = 0.9$. In this case, $t + d > 1.0$. It indicates information contradictory and knowledge defect $kd(t, d) = |1 - t - d|$. Certainty can be derived from knowledge defect.

With regard to trust propagation, it only uses trustworthy paths, as distrust information is very complicated and difficult to use. Unlike others, instead of proposing one trust propagation operator, it lists several operators. It uses weighted mean to aggregate trust from parallel trust paths. It also proposes several operators based on how to set weight for each path.

Wang and Wu

In [73], Wang and Wu computed trust and certainty by collecting evidence and using Dempster-Shafer Theory as in [75]; however they considered multi-dimensional evidence and trust. They also proposed several selection strategies, such as selecting primitive dimensions and subsets. To propagate trust, they used the parameterized family of Frank t-norm [114], in which discounting rate is controlled by the input parameters. Multiple trust paths are combined by weighted mean, where weights are derived from certainty of trust paths. They also tackled the problems caused by shared links (links shared by two or more paths between the truster and the trustee) and crossing links (links cross two paths) in trust networks.

Verbiest, Cornelis et al.

Authors of [115] adapted the framework from [2]. It represents trust using bi-lattices approach; however, to aggregate multiple paths between the truster and the trustee, [115] uses weighted mean approach where weights are dependent on paths' length. As increasing paths' length can decrease inference's accuracy, it weights paths' influence based on the order of paths' length. Also, it proposes a dynamic horizon search strategy, in which it sets a global threshold for the length of path; however, when the shortest paths' length is less than the global threshold, it only considers those shortest paths. By incorporating paths' length into trust inference, it tries to optimize the trade-off between coverage and accuracy.

2.4.6 Multiplication for Transitivity and Maximum for Aggregation

Zhao and Li

VectorTrust [116] provides a local trust management framework for Peer-to-Peer file sharing systems. It uses a single value to represent trust degree/level. To propagate trust, trust degrees/levels are multiplied together along the chains. And when there are more than two paths between the pair of users, it selects the most trustworthy path. Note that, only when the truster has no direct trust towards the trustee, indirect trust will be inferred and used.

Hao, Min et al.

MobiFuzzyTrust [107] models trust in a comprehensive way. It considers prestige-based trust, familiarity-based trust, similarity-based trust and risk, and combines them to calculate trust value; however, instead of using the numerical values, MobiFuzzyTrust represents trust with linguistic terms. Fuzzy membership functions are defined to convert trust from numerical values to linguistic terms. To infer indirect trust, it first multiplies numerical values. If there exist more than one path between

the truster and the trustee, it chooses the path which has the maximum trust value. Finally, the numerical trust values are converted back to linguistic terms using the fuzzy membership functions.

2.4.7 Social Theories Based Method

Huang, Kimming et al.

In [117], Huang, Kimming et al. proposed a trust framework based on Probabilistic Soft Logic (PSL). It uses soft truth values as trust degrees. To infer indirect trust for unconnected truster and trustee, it follows two social theories – balance theory and status theory, and develops two rules correspondingly. Specifically, following the balance theory, only triangles which contains one or three strong/positive links are considered as balanced. In status theory, if the truster trusts the trustee, it means that the trustee has higher status than the truster. Also, it takes reciprocation of trust into account as another rule for two social theories.

2.4.8 Machine Learning Based Method

As machine learning becomes more popular, there are also many works using machine learning techniques to predict social links for online social communities [58, 106, 118–120]. In such types of works, each link is labeled as positive or negative. In [121], authors combined behavior based methods, such as weighted mean and min-max aggregation, and machine learning method – reinforcement learning, together. In this chapter, we mainly focus on trust management frameworks which are behavior based.

2.4.9 Social Theories and Machine Learning Combined Method

Tang, Gao et al.

Tang, Gao et al. proposed a low rank matrix factorization method – hTrust [122] to predict trust relationships. Besides considering latent factors, it also considers homophily effect which is widely existed in online social networks. Basically, similar users are more likely to trust each other than others. Therefore, in the objective function, it includes the similarity of two users' latent vectors as one regularization term.

Yao, Tong et al.

Matri [27] treats trust aspects as latent factors and uses matrix factorization to predict trust values. Similar to the classic collaborative filtering algorithm, there are two matrices in Matri: truster matrix and trustee matrix. It also adapts four trust propagation operators from [56]. Besides these four operators, it also takes global bias, truster bias and trustee bias into account. It combines four social trust propagation operators with the matrix factorization method.

We can see from above that many schemes used weighted mean to aggregate trust from multiple trust paths; however, their weights were assigned differently. We summarize their weights in Table 2.2.

2.5 Attacks in Trust Management Frameworks and Corresponding Defense Mechanisms

Security is now a very hot topic in many fields of computer science [123–125]. Trust management frameworks can help to mitigate the damage in many applications, such as access control, authentication, secure service provision and secure routing [53]; however, they themselves can be the targets of malicious attackers, too [126, 127]. In this section, we discuss several potential attacks in trust management frameworks.

Table 2.2.
Weights in weighted mean schemes

Schemes	Weights
Abdul-Rahman and Hailes et al. [46]	Recommender trust
Jøsang [75]	Trusters' direct trust
Sabestian [76]	Product of trust and confidence
Wang and Singh [72]	Trusters' direct trust
Liu, Yang et al. [74]	Trusters' direct trust
Kamvar, Schlosser et al. [13]	Trusters' direct trust
Xiong and Liu [69]	Similarity
Golbeck2005 [4]	Trusters' direct trust
Sun, Yu et al. [40]	Recommender trust
Massa and Avesani [24]	Trusters' direct trust
Liu, Wang et al. [113]	Trusters' direct trust
Zhang and Durresi [16]	Trusters' direct trust

Attackers in trust management frameworks are malicious participants who are motivated either by selfish or malicious intentions [128]. Selfish attackers launch attacks for their own benefits, while malicious attackers aim to degrade others' trust and then affect the system's performance [37, 129]. According to [37], attackers can be classified into insiders and outsiders. Insiders are those who can get access to the systems and participate in the systems as normal participants, while outsiders are not authorized by the systems. Obviously, attackers inside the systems can cause more damage than outsiders. Therefore, many traditional approaches focus on authenticating participants' identities by using cryptography primitives [130, 131]. In today's life, identity authentication is not sufficient. It is very easy for attackers to get into the systems in many open environment applications [37], which include online social communities. Authorized participants in online social communities may behave badly, e.g. providing misleading information. In such situations, trust is introduced for the purpose of helping participants to avoid cooperating with potential malicious attackers. In [132], Rasmussen and Jansson used hard security to refer identity au-

thentication, and treated social control mechanisms, e.g. trust, as soft security. In this chapter we only focus on soft security.

Attackers can behave in various ways for different purposes. Based on this, Hoffman et al. classified attacks in reputation systems into self-promoting, whitewashing, slandering, orchestrated and denial of service [37]. In [53], authors listed misleading feedback attack, discrimination attack, on-off attack, Sybil attack and new comer attack. There are even more types of attacks in [133]. Some of those attacks, such as self-promoting and slandering, are considered for reputation systems or global trust only. While some of them are application dependent, e.g. imbalance value attack, denial of service. In this chapter, we mainly focus on potential attacks that can happen to local trust. We list four types of attacks based on attackers' behaviors. Note that we consider Sybil attack [134] as an auxiliary method for attackers to achieve their goals. So, it can be launched with any of the following attacks.

2.5.1 Naive Attack

As pointed out by [53], attackers may provide misleading recommendations to their neighbors. Dishonest recommendations can affect users' decisions. Also, it can be used in reputation systems to launch the self-promoting attack and slandering attack by providing negative feedback for good participants and positive feedback for their conspirators. In the naive attack, attackers blindly provide dishonest recommendations and have no knowledge about the systems. They do not realize that their dishonest recommendations may not be considered if they are untrustworthy to other participants.

To defend against the naive attack, when considering intermediary participants' recommendations, many systems only take into account recommendations from trustworthy neighbors [4, 16, 34, 135]. Using weighted mean, attackers' recommendations will be weighted by their own trust levels. Some schemes, such as [4] and [34], set certain thresholds to select trustworthy paths. In order for the recommendations to

be considered, trust paths' trust levels must be higher than the thresholds. In such cases, if participants do not trust attackers, the dishonest recommendations have no or very little impact. There exist few other mechanisms to defend against the naive attack, e.g. clustering [53]; however, we do not consider them in this chapter as we only focus on trust-based mechanisms.

2.5.2 Traitor Attack

As we discussed, if attackers' trust levels are low, their recommendations can only have very little impact on other participants' decisions. This is intuitive and can also be learned by attackers. Therefore, it is possible that before attackers begin to disseminate dishonest recommendations, they will provide honest recommendations for a period of time in order to become trustworthy neighbors of normal participants. Such an attack is called the traitor attack (or On-off attack) in [53, 136] because attackers can suddenly change their behaviors.

If we only consider a single attacker's behavior, the traitor attack cannot be completely eliminated. Before the first malicious behavior happens, attackers have good trust levels because of their previously disguised behavior. There is no way to predict attackers' first bad behavior based on their former trust levels. Therefore, when we discuss the defense to the traitor attack, we refer to defending against attackers' following consecutive bad behavior. The purpose of defense mechanisms is to detect attackers and remove them or mitigate their impact as soon as possible. One straightforward way is that bad behavior is given more weight than good behavior [136]. This means that participants have to behave good for a long time in order to become trustworthy, while their trust can decrease dramatically even if they only behave badly one time [1, 69]. It requires systems to update trust in a timely manner. Under this situation, attackers' sequential dishonest recommendations will not be accepted as their trust decreases immediately after the first dishonest recommendations. Apart from this, systems can put higher weights on recent evidence than

previous evidence such that trust is mainly determined by recent behavior (also called forgetting factor) [69, 137]. To summarize, these strategies aim to reduce attackers' trust immediately once they behave badly.

2.5.3 Whitewashing Attack

Attackers having very low trust levels may be interested in discarding their current identities and re-enter the systems. This is called the whitewashing attack since attackers can behave as new comers and hide their bad histories [138]. Whitewashing attack is a very common phenomenon in many online social communities because participants are able to create identities and re-enter the systems very easily [139]. Whitewashing attack is especially attractive in systems where bad history can lead to negative trust levels. For example, attackers' trust is negative because of their previous malicious behaviors. Then, they only need to re-enter the systems, and their trust becomes zero, which is better than before.

Defense mechanisms for the whitewashing attack can be divided into two aspects. First, systems can prevent participants from creating multiple identities or make it expensive. For instance, some systems require users to provide social security numbers or biometrics to register for identities. This kind of defense mechanism is related to hard security, which is beyond the scope of this chapter. On the other hand, systems can assign the lowest trust levels to the new comers such that there is no incentive for participants to re-enter the systems again. In those systems which consider confidence, attackers will lose their former confidence if they re-enter the systems [16]. Of course, it is a challenge for normal new comers to become trustworthy, which is known as the cold start problem [140].

2.5.4 Collusion Attack

Attacks mentioned previously can be launched together by either a single attacker or several attackers. We refer to the collusion attack the combination of multiple attacks, and it can be launched by a number of attackers [128, 141].

In order to get identities in a system, malicious users can launch the Sybil attack first. Sybil attack is one of the most popular attacks in online systems. In the Sybil attack, a single user is able to create many identities and behave as if there were multiple participants. In some extreme cases, attackers can create millions of identities such that the system will be dominated by Sybil accounts.

Compared with the above three attacks, the collusion attack is more complicated and difficult to detect [128]. In the collusion attack, attackers can act in several different ways to achieve their goals. In addition, attackers can divide malicious identities into different groups, and each group has their own responsibility at a given time. For example, in reputation systems, one group of accounts rate their conspirators with high trust in order to increase their global trust. Their conspirators are responsible for disseminating dishonest recommendations. There can be many other tasks divided among groups. To make it more complicated, attackers can switch their roles during the process [128].

As the collusion attack is the combination of different types of attacks, defense mechanisms also need to employ several methods together. There are some works trying to find out colluded attackers. In [136], Sun et al. developed a defense mechanism with temporal and correlation analysis. In order to find approaches to defend against the collusion attack, it first analyzed one type of the collusion attack, which they called RepTrap attack. In RepTrap attack, attackers have several features and behavior patterns. TAUCA, which is the defense mechanism, has three components: change detection, user correlation calculation and malicious user group identification. Change detector is used to monitor the changing trend of behavior (rating). Then TAUCA analyzes the correlation among suspicious participants. Finally TAUCA can

identify malicious participants' groups. More details about TAUCA can be found in [136]. Colluded attackers can be considered as clusters in graph models as they are similar to each other. With this observation, clustering algorithms are used to find out groups of participants in the systems.

From what we discussed above, behavior of the collusion attack can be changed with different attackers' strategies. Although we have seen examples of defense mechanisms to defend against the collusion attack, we should note that they all have certain assumptions about attackers' behavior. For example, in order to develop defense mechanisms, they need to know attackers' behavior patterns in advance, which is a tough task in reality.

To summarize, we can see that attackers have many methods to damage the systems. For example, attackers can launch the Sybil attack and the naive attack together. Although we discussed some defense mechanisms to deal with such attacks, there is a great need for further research work in this field. More importantly, in many applications, the defensive strategies should be used together in order to defend effectively against attackers. Finally, remember that attackers can also learn defense mechanisms and become immune to them. Therefore, it is like an "Arms race" between attackers and defense mechanisms.

2.6 Analysis of Vulnerability to Attacks

In the above section, we listed four types of potential attacks in trust management frameworks. As the collusion attack is dependent on attackers' strategies which are different in applications, in this section, we analyze existing schemes' vulnerabilities to the naive attack, the traitor attack and the whitewashing attack. We examine existing schemes to see whether they have the defense mechanisms we mentioned in Section 2.5 to defend against corresponding attacks. For those systems which do not consider trust propagation, such as [1], we do not analyze their vulnerabilities to attacks. Also, for machine learning based methods, we do not analyze their vulnerabilities.

Abdul-Rahman and Hailes in [46] proposed a trust management system which is used in virtual communities. Their model propagates and aggregates trust by weighted mean, where weights are intermediary participants' recommender trust. Therefore, it is robust to the naive attack as naive attackers' dishonest recommendations will be discounted. Also, the truster updates recommender trust after each recommendation finishes. In such situations, if attackers suddenly change their behavior, their recommender trust will be decreased immediately. So it can defend against the traitor attack as well. But it is vulnerable to the whitewashing attack as new comers have neutral trust levels, which is better than a bad trust level, e.g. "very untrustworthy".

Subjective Logic [75] proposed by Jøsang defines trust following belief theory. In this scheme, new comers have the lowest trust, therefore, the whitewashing attack does not have any impact. In trust transitivity, as evidence is discounted by intermediary participants' trust, it is robust to the naive attack. Unlike [46], Subjective Logic does not compare recommendations with the truster's own experiences. Also, it does not take into account temporal information and forgetting factor, so it is vulnerable to the traitor attack. CertainTrust [76], which is built based on Subjective Logic, has the same characters as Subjective Logic, as well as [74] and [110].

[13] uses normalized local trust for each participant. New comers have the lowest trust levels, therefore, there is no incentive for attackers to re-enter the system. It uses weighted mean mechanism to defend against the naive attack. Unfortunately, [13] does not contain any defense mechanism for the traitor attack. Hence it is vulnerable to the traitor attack.

Xiong and Liu proposed PeerTrust [69] for Peer-to-Peer networks. It takes many factors into account in modeling trust, including time decaying, different weights for positive and negative evidence, which makes it robust to the traitor attack. Naive attackers' recommendations are discounted by their trust, so it is robust to the naive attack as well. There is no incentive for attackers to re-enter the system. Those

features, combined together, make PeerTrust more robust to the collusion attack compared with other schemes.

TidalTrust [4] is robust to the naive attack as it uses weighted mean for trust aggregation. Also, the whitewashing attack is avoided because new comers have the lowest trust levels; however it does not contain any defense mechanisms for the traitor attack.

Sun, Yu et al. used a probability based trust in [40]. They put penalties on bad behavior by dramatically decreasing trust. Also, trust can only increase gradually even though participants behave very good. Therefore, it is robust to the traitor attack. As it uses weighted mean for trust transitivity and aggregation, it is robust to the naive attack. It is vulnerable to the whitewashing attack as new comers have better trust levels than bad participants (negative levels).

In MoleTrust [24], Massa used one continuous value to represent trust. Because only trustworthy paths will be accepted in his model, it is robust to the naive attack. It is also robust to the whitewashing attack as new comers have the lowest trust. But it is vulnerable to the traitor attack.

[113] uses weighted mean as well, so it is robust to the naive attack. Although it takes recommendation roles into account, it does not update them after each recommendation. Therefore, it is vulnerable to the traitor attack. Because new comers have the lowest trust levels, it is robust to the whitewashing attack.

In [16], trust evaluation is considered as a "measurement". Trust is defined by rating values between participants and confidence is related to the number of ratings. Both of them are continuous values between 0 and 1. Their model is robust to the naive attack and the whitewashing attack as it uses weighted mean and assigns the lowest trust levels for new comers.

In [57], Golbeck proposed a scheme to infer binary trust in social networks. When considering recommendations, only trustworthy neighbors' recommendations are selected. Therefore, it is robust to the naive attack. It is also robust to the whitewashing

attack as new comers are not trustworthy in the beginning; however it is vulnerable to the traitor attack.

Guha [56] used four atomic operators to calculate trust and distrust matrices. In his model, both trust and distrust can be propagated. As he used distrust, the whitewashing attack is possible in his model. Trust is discounted when it propagates through the chains, so it is robust to the naive attack. Unfortunately, it is vulnerable to the traitor attack. [59] adopts Guha's work [56] and changes four atomic operators to two. But they have the same characters regarding attacks.

Victor used bi-lattice based trust in [2]. Knowledge defect captures to what extent participants are certain about their estimations. It is vulnerable to the whitewashing attack as new comers have better trust levels than bad participants. As it considers thresholds in trust transitivity, it is robust to the naive attack. It is vulnerable to the traitor attack as there is no defense mechanisms. [115] has the same properties.

[73] evaluates trust similar to [75], therefore, it is robust to the whitewashing attack. It uses the parameterized family of Frank t-norm to combine trust paths, where discounting rates are controlled by participants. So it provides opportunity to defend against the naive attack. It does not update the discounting rate, hence it is vulnerable to the traitor attack.

[117] propagates and aggregates trust following balance theory and status theory. In this cases, the inferred trust is determined by the corresponding triangles. Therefore, it is vulnerable to the naive attack and the traitor attack. It is unclear for the whitewashing attack as the lowest trust value is dependent on specific applications.

[116] and [107] only select the most trustworthy paths to aggregate trust paths. Therefore, they are robust to the naive attack. Also, as the new comer has the lowest trust degree, both of them are robust to the whitewashing attack.

We summarize the above analyzed results in Table 2.3. For each type of attack, if the scheme is robust to the attack, we list which mechanism is used accordingly. For those schemes which are vulnerable to the attacks, we represent it by "No" in

the corresponding attacks. We can see that although the naive attack is considered in many schemes, only few schemes take the traitor attack into account.

Table 2.3.
Vulnerability to attacks

Schemes	Naive attack	Traitor attack	Whitewashing attack
Abdul-Rahman, Hailes et al. [46]	Weighted mean	Updating recommender trust	No
Jøsang [75]	Weighted mean	No	Lowest trust level for new comer
Sabestian [76]	Weighted mean	No	Lowest trust level for new comer
Wang and Singh [72]	Weighted mean	No	Lowest trust level for new comer
Liu, Yang et al. [74]	Weighted mean	No	Lowest trust level for new comer
Kamvar, Schlosser et al. [13]	Weighted mean	No	Lowest trust level for new comer
Xiong and Liu [69]	Weighted mean	Forgetting factor, time window	Lowest trust level for new comer
Golbeck2005 [4]	Weighted mean	No	Lowest trust level for new comer
Sun, Yu et al. [40]	Weighted mean	Forgetting factor	No
Massa and Avesani [24]	Weighted mean	No	Lowest trust level for new comer
Liu, Wang et al. [85]	Weighted mean	No	Lowest trust level for new comer
Zhang and Durresi [16]	Weighted mean	No	Lowest trust level for new comer
Golbeck2006 [57]	Threshold	No	Lowest trust level for new comer
Guha and Kumar [56]	Weighted mean	No	No
Zhang and Mao [59]	Weighted mean	No	No
Victor, Cornelis et al. [2]	Threshold	No	No
Verbiest, Cornelis et al. [115]	Threshold	No	No
Wang and Wu [73]	Weights adjusted by Bayesian network	No	Lowest trust level for new comer
Huang, Kimmig et al. [117]	No	No	Dependent on applications
Zhan and Li [116]	Most trustworthy path	No	Lowest trust level for new comer
Hao, Min et al. [107]	Most trustworthy path	No	Lowest trust level for new comer

2.7 Chapter Summary

In this chapter, we discussed the urgent need of trust management frameworks in many online social communities. We investigated how trust is defined by researchers from different disciplines and how can it be represented in the field of computer science. As we can see, it has various computational models depending on how

people understand it. The definitions and representations of trust are basics for trust management frameworks. Besides trust, confidence is another important concept in trust-based systems.

Furthermore, we presented different trust management schemes. Many of them have two important operators: transitivity and aggregation operators. This can largely increase the number of connected participants. Transitivity operator is used to infer indirect trust for two participants who originally are not directly connected. Aggregation operator, which always works together with transitivity operator, deals with the situation when there are more than one parallel trust path between the truster and the trustee.

Finally, we reviewed some potential trust attacks in trust management frameworks. We described four types of behaviors in these attacks. We analyzed existing schemes' vulnerabilities to the attacks. If they are robust to the attacks, we listed which defense mechanisms they use.

Compared with previous survey papers in this field, we provided a comprehensive survey that takes two challenges – trust modeling and trust inference, into account. In addition to that, we also discussed four types of potential attacks that can happen in trust management frameworks.

3 A MEASUREMENT THEORY BASED TRUST MANAGEMENT FRAMEWORK

3.1 Introduction

Trust is a complicated human behavior developed during our evolution. Depending on circumstances and applications, trust has many different interpretations, and consequently, different representations and management principles [1]. Trust has been a hot research topic in many fields, such as psychology, sociology, IT systems, and so on. For example, trust has been used in electronic markets, such as eBay [142], in Internet of Things [143], and in Peer-to-Peer systems [144]. In such applications, trust is constructed by algorithms through observing past events, such as positive or negative evidences or feedback [75, 144, 145].

In recent years, the explosive success of online social networks has encouraged the exploration of new directions for computerized trust representations and management of (cognitive) trust [110, 146–150]. Cognitive trust is especially useful in cases where it is difficult for computers to evaluate evidences; however, human trust, especially in large online social communities, such as Facebook, Twitter, Amazon, etc, needs support from computer systems. Due to the large amount of available data in today's information age, it is impossible for users to handle trust like in real lives, where people only have a limited number of acquaintances [16].

Therefore, we need a framework that can effectively, but also intuitively, let people express their trust, and enable the system to automatically and securely summarize the massive amounts of trust information, so that a user of the system can make "educated" decisions, or at least not blind decisions. In this chapter, we focus on two perspectives of trust: how to represent trust, and how to manage trust in online social communities. A lot of research has been done in this field [3, 4, 56, 110, 151].

For example, Subjective Logic [75,152] has been developed to express and manipulate subjective trust based on the Dempster-Shafer belief theory [153].

Trust has turned out to be very helpful for users to make decisions [154]; however, in many online social communities, existing user-to-user trust relationships are very limited when compared with the number of all potential pairs of users [2, 3]. As in real life, users can only evaluate others with whom they have direct interactions. Unfortunately, such user-to-user direct trust relationships are not sufficient, which always results in sparsely connected online social communities. One common way to alleviate this is to use existing user-to-user direct trust relationships to infer indirect trust relationships for users who are not directly connected [2, 4].

We develop our approach based on the similarities between human trust and measurements [155]. They are both evaluations of some values, enhanced by repeating the evaluations. Furthermore, the "error", which is used to express the certainty in measurement theory and statistics [155], is similar to humans' confidence when people assess trust relationships. Basically, the larger is the error, the smaller is the confidence. For variable x, given a range of estimation $[\bar{x} - \delta_1, \bar{x} + \delta_2]$, there is a certain probability that the true value $\hat{x}$ lies in this range [155]. For example, in Normal distribution, $[\bar{x} - \delta, \bar{x} + \delta]$ (here δ is the standard error) corresponds to 68% confidence level. In addition, when we propagate trust, we must take into account the corresponding confidence, similar to the theory of error propagation, which integrates single step errors in a chain of measurements.

We adapt our framework to several specific trust inference formulas. Besides Epinions.com, we also collect another data set from Twitter and establish trust network within it. To infer indirect trust relationships, we use different formulas in two online social communities. And we find that different communities or data sets have their own trust inference patterns. Our main contributions include:

- Establish user-to-user trust networks for two real online social communities: Epinions.com and Twitter;

- Propose a general trust management framework which is based on measurement theory to study and infer indirect trust relationships;
- Our framework is flexible and can be adapted to various trust inference formulas;
- Show that online social communities have different patterns such that selecting trust inference formulas for different applications is important;
- Show one benefit of inferring indirect trust – mitigating the sparsity problem in online social communities.

The rest of this chapter is organized as follows: In Section 3.2, we introduce background about trust processing as well as some related works. In Section 3.3, we state the similarity between trust and measurement theory, and define the trust metric. In Section 3.4, we then describe our measurement theory based trust management framework and two important trust inferring operations. In Section 3.5, according to existing works, we list several transitivity and aggregation formulas. In Section 3.6, we do experiments on two data sets in order to validate our framework, and then analyze results . In Section 3.7, we show one of the main benefits of using the trust management framework to infer indirect trust relationships. Finally, we conclude this chapter in Section 3.8.

3.2 Background and Related Works

3.2.1 Trust Processing in Online Social Communities

The goal of trust management systems is to provide users with trust information and help them make decisions. As shown in Figure 1.1, we divide trust processing into three major phases. Trust Modeling deals with mapping the available trust related raw data from the field into trust metrics. For example, in Epinions.com, users have reviews and propositions; in Facebook, users have likes and dislikes, and so on. Such data has to be translated into trust metrics, which are intrinsic components

of the trust management framework. Trust Inference focuses on propagating and aggregating the obtained trust metrics over the whole network, or over the part of interest. Finally, in Decision Making, the produced trust knowledge obtained by trust management is used to support various decision making.

All three phases of trust processing are dependent on the context and are interrelated. The accuracy of our Trust Inference, and its corresponding level of support in Decision Making, will depend on the availability and granularity of trust data from the field. While Trust Modeling and Decision Making can place constraints on the context, such as limitations from the raw data or the type of decisions, Trust Inference should not limit the potential of the raw data, but potentially increase it, by leading to more trustworthy decisions.

3.2.2 Related Works

Trust in online social communities has been attracting more attention from computer scientists. Consequently, many trust management frameworks have been proposed in recent years [156, 157].

A. Jøsang proposed a model called Subjective Logic in [75] that considers trust as a term of uncertain probabilities based on the Dempster-Shafer belief theory [153]. It has two spaces: opinion (or belief) space and evidence space. In the opinion space, there are four metrics: belief, disbelief, uncertainty and relative atomicity. In the evidence space, it includes positive and negative evidence. Metrics in two spaces can be converted into each other. When considering trust transitivity, transitive belief will be discounted by multiplying beliefs along the chain. In the cases that there are more than one path between two users, evidence will be added together first and then be converted into the opinion space. However, it does not consider conflict ratio when calculating confidence. Confidence in Subjective Logic is only related with the amount of evidence. Y. Wang et al. [72] proposed a framework based on Subjective Logic, as well as S. Ries [76]. Shin [158] takes into account

unreachable witness, which is based on common acquaintances. A selection operator is introduced in [112]. Both [72] and [158] define confidence based on the deviation between evidence's distribution and uniform distribution. Although they take both the amount of evidence and conflict ratio into account in calculating confidence, it is more difficult than Subjective Logic to convert between evidence space and trust space. As indicated by [72], there is no closed-form solution for conversion function. Therefore they are more computationally expensive than Subjective Logic. Our framework captures both the amount of evidence and conflict ratio. At the same time, it is simple to derive confidence from error.

[159] models trust using Hidden Markov Model. It considers reputation for two users when local trust is not available. TRAVOS [160] models trust using Beta distribution. It considers third-parties' opinions only if the direct confidence is below a threshold. However, it can only be applied in cases where evidence is binary. Our framework can be applied in cases no matter that evidence is represented as binary or continuous.

MoleTrust [20] first selects a sub-graph of the whole network, which contains the source user's (truster) contacts that are reachable within the limited number of hops. It then calculates trust in an iterative way in the sub-graph, using the weighted mean. It sets a trust threshold during computing; only those edges whose trust values are greater than, or equal to the threshold, will be taken into account for transitivity and aggregation. Introducing threshold can improve accuracy, but it reduces the number of pairs of users that can be connected. H. Tosun and J. W. Sheppard adapted MoleTrust in [135] for a better trade-off between them. Similar to MoleTrust, TidalTrust [4] also uses the weighted mean to calculate trust transitivity and aggregation. However, both MoleTrust and TidalTrust do not consider confidence. Y. Sun et al., [40] proposed a trust model based on entropy. They defined three axioms to infer indirect trust. Also, [40] does not consider confidence. To propagate trust, RATE [3] differentiates neighbors or recommenders based on four metrics: trustworthiness, expertise, uncertainty and cost.

R. Guha is the first one among computer scientists taking both trust and distrust into account [56]. It represents user-to-user trust relationships in matrices, and includes four operators. Indirect trust is predicted by multiplying trust matrix (or distrust matrix) with four operations. The final trust representation is more like reputation rather than personal trust perspectives. [122] predicts indirect trust using matrix factorization approach, with user homophily as a regularization term. These two works are more like machine learning approaches.

For those different schemes, [161] proposes a framework to evaluate and compare reputation systems' performance. The existing works mainly focus on trust modeling (based on evidence, probability, belief theory, and so on) and trust propagation formulas (i.e. multiplication, evidence accumulation, averaging, and so on). In this chapter, we propose a trust management framework which has two metrics: trustworthiness and confidence. We propose a simple measurement of confidence directly based on the error in the measurement theory, which is a well-accepted theory for general measurement purpose. More important, by using the error propagation theory which can be used for many general functions, confidence can be easily calculated for trust transitivity and aggregation formulas. The error propagation theory is a well-established theory in the field of error analysis [162]. Given input x and its error, it is easy to calculate the propagated error of $f(x)$, as long as f is derivative. Therefore, unlike other works that stick to specific trust transitivity and aggregation formulas, i.e. multiplication, our framework is flexible and can be adapted to various transitivity and aggregation formulas. Although there are some existing works that include confidence as well [26, 160], our computation of confidence is much simpler. Similar to [26], confidence in our framework captures both the number of measurements and their distribution (which reflects conflict ratio).

3.3 Trust Metric Inspired by Measurement Theory and Psychology

Measurement theory, which is a well developed and proven field of knowledge, quantifies the difference between the measured value and the corresponding objective value [163]. Additionally, a number of notations, categorized as approximation error (or "error" in general), are introduced to represent the accuracy, precision or uncertainty of a "measurement" [164], such as absolute error, relative error, confidence interval, and so on.

3.3.1 Psychology Implication

People develop their impressions about others based on their interactions and incidents. Furthermore, feedback is gathered and processed by the brain that revises the accumulated impression, which is generally called "trust" [1]. This repeating process makes our evaluation of trust regarding people or other entities more concrete: How trustworthy are they? For example, as indicated by [7], positive experiences will increase trust. This formed trust can be used later to support decision making.

Physical measurements possess similar characteristics of human trust evaluation. People get an initial evaluation about a given physical quantity by measuring it using the appropriate equipment. They can then improve the measurement accuracy by using more precise equipment, combining different measurement methods, or repeating the measurement. Such similarity inspires us to adapt the well established and proven measurement theory in representing and computing trust relations in online social communities.

3.3.2 Trust Metrics: Trustworthiness and Confidence

In our framework, we use two metrics – trustworthiness and confidence, together to represent trust. We first introduce the trustworthiness (or can be called impression in the scenario of human trust) metric m as a person's (say Alice's) comprehensive sum-

mary of multiple "measurements" on another person's (say Bob's) trustworthiness, through their real life experiences including personal direct and indirect contacts in their social context. Although the specific processing methods are different, this summarization is very similar to the averaging of sample measurements in statistics [165]; however, the concrete meaning of m depends on specific scenarios and applications. For example, m could be considered as a quality value (e.g., how good is Bob), a probability (e.g., how likely Bob will keep promises), and so on. Some widely used representations of trust are: binary metrics [56], scaled metrics [46, 60], probability based metrics [26, 62] and similarity based metrics [82, 84, 166]. Our framework can deal with both discrete and continuous metrics. In this framework, suppose that we have a set of measurement results $M = \{m_1, m_2, ...m_k\}$, then trustworthiness m is defined as in Equation 3.1.

$$m = \frac{\sum_{i=1}^{i=k} m_i}{k} \tag{3.1}$$

Similar to sampling in statistics, depending on the number of incidents and the intensity of each experience, Alice would have a distribution of measurements in a range around the summarized trustworthiness m. Such a distribution, which in fact shows to what extent Alice is confident about her trustworthiness assessment, is similar to "error" in physical measurements, which represents the variance of the actual value from the summarized value. Therefore, we introduce the second metric: confidence c. From the psychological perspective, confidence c represents how much a person is certain about his/her trustworthiness metric, and from the statistical perspective, c determines how far away from the "real" trustworthiness the "measured" one can be. Therefore, we associate c with "variance" or "error" of measurement theory, in an inversely proportional manner. It is intuitive that the smaller the "variance" or "error" is, the higher the confidence. Therefore, in our framework, a trust tuple contains trustworthiness m and confidence c, which can be represented as $T(m, c)$. Basically, trustworthiness/impression m measures how trustworthy the trustee is in the truster's point of view. And, confidence c measures how confident the truster is

about the evaluation of trustworthiness/impression m. In our framework, m and c together compose trust T.

3.3.3 Value and Interval of Trust Metrics

One way of asking people's opinions about other entities is to let them assign approximate values in a given interval, which is referred to as a "scaled question" in surveys and questionnaires. For example, "Likert-Scale" [167] lets users express their induction of past experiences, and then selected options or values that can be converted into predefined trustworthiness metrics m. In our framework, we define both the value of m and c as continuous values in $[0, 1]$. A higher trust value means that a person is more trustworthy. For example, 0 means most untrustworthy, while 1 refers to most trustworthy.

In order to utilize the error propagation theory to compute transitive and aggregated trust (discussed in the later section), we must be able to convert confidence c to error in a corresponding form. As a result, we further introduce another intermediate metric: range R, which is deduced from confidence c. If we consider c as the percentage of known fact, then the percentage of uncertain fact would be $1 - c$. Therefore, R is the total trustworthiness interval times the percentage of uncertain fact. Generally, for a trust tuple $T(m = 0.5, c = 0)$ which is the most neutral and uncertain trust, we would like the possible trustworthiness value $[m - \frac{R}{2}, m + \frac{R}{2}]$ to cover the whole interval, i.e., the "real" trustworthiness value could be any value in $[0, 1]$. On the contrary, when $c = 1$, which represents the highest confidence, we would like $R = 0$, which means both the worst and best expected trustworthiness equals to m. Following these guidelines, the relation between confidence and range can be simply defined as: $R = 1 * (1 - c) = 1 - c$.

To better fit the error characteristic, radius r, which is half of range R is introduced. r shows how far the best or worst expected trustworthiness can be from the summarized trustworthiness value m. Therefore, in this definition m is equivalent to

the measurement mean, and r is equivalent to the standard error of the mean [168]. Conversion between r and c can be written as Equation 3.2.

$$c = \begin{cases} 1 - 2 * r, & \text{if } r \leq 0.5 \\ 0, & \text{otherwise} \end{cases} \qquad \text{and} \qquad r = \frac{1-c}{2} \tag{3.2}$$

To illustrate the relationship among m, c and r, we give a Normal distribution example in Figure 3.1. Here, the black line represents the mean of measurements m_i, which is the trustworthiness m. The blue line represents the standard error r, and confidence $c = 1 - 2 * r$ can be represented by red line. Basically, more consistent are the measurement results, smaller is the standard error r, which results in higher confidence.

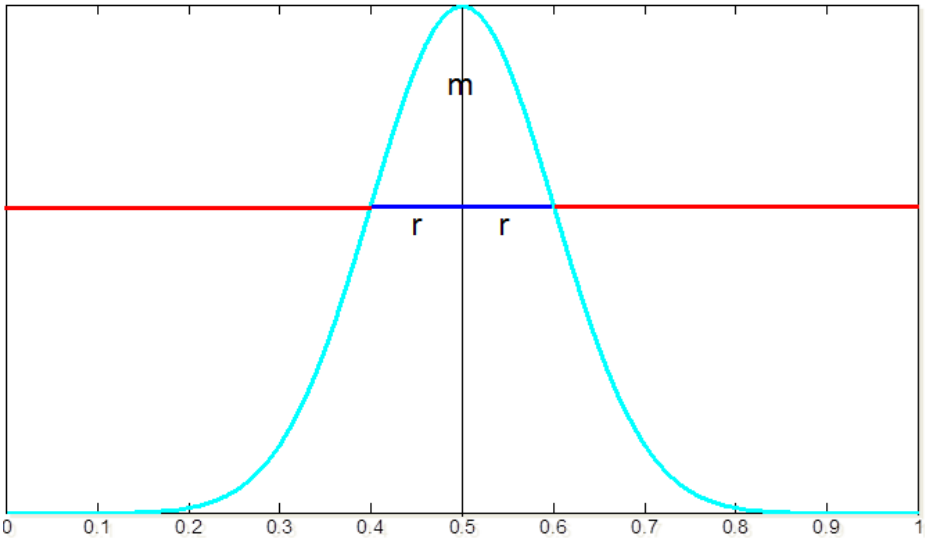


Figure 3.1. The relation among m, c and r

3.4 Trust Inference Framework

The fundamental assumption for frameworks inferring indirect trust is that trust is transitive. This is also supported by some psychologists and sociologists, such as Stanley Wasserman and Katherine Faust. They stated in their book [169]:

> "Holland and Leinhardt (1972) provide strong, statistical evidence that transitivity is a very important structural tendency in social networks." [169, 170].

In the previous example, we can call Alice the truster or evaluator who evaluates Bob's trustworthiness, and Bob on the other hand can be called the trustee or evaluation target whose trust is evaluated by Alice. If we treat the evaluator, the evaluation target, and all intermediate users as nodes in the graph, indirect trust relation builds a path that starts from the evaluator and ends at the evaluation target, connected by all intermediate users [5, 6]. For example, on Facebook, users (i.e. B) are able to recommend their friends (i.e. C) to other friends (i.e. A).

Error, which represents "uncertainty" in statistics, can be propagated and accumulated when a system is assembled from components each of which introduces different levels of error in measurement. The error propagation theory is then constructed to summarize the overall error of the system based on statistics theory. In this section we discuss the trust inference based on the error propagation theory using the trust metric m and c, and how we adapt them to comply with psychological implications.

There are two types of trust propagation operations: trust transitivity and trust aggregation [5,6]. We illustrate them using the scenario where node A is the evaluator, and node Z is the evaluation target. Node B is the intermediate node between node A and node Z. Node B can provide recommendations to node A, since node B knows node Z.

3.4.1 Trust Transitivity

Based on trust's transitivity property, in the above case, node A trusts node B and node B trusts node Z, and to some extent node A also trusts node Z. We denote

the operation of transitive trust as $\otimes$. Then node A's indirect evaluation of node Z via node B is represented as:

$$T_Z^{A:B} = T_B^A \otimes T_Z^B \tag{3.3}$$

This is a concatenation of trust path A-B and B-Z using node B as a connecting node for trust transitivity. T_B^A and T_Z^B can be either direct trust or an abstraction of transitive trust. In the case that T_B^A or T_Z^B are already indirect transitive trust, $T_Z^{A:B}$ extends to more than two hops.

Principles of Trust Transitivity

The formulas designed for computing transitive trust should comply with psychological observations. We list the following desired principles, similar to other previous works [6, 62, 171]:

- TPrinciple1: Trust transitivity will not increase confidence under all circumstances, i.e. $c_Z^{A:B} \leq min\{c_B^A, c_Z^B\}$.
- TPrinciple2: Trust transitivity will not increase the original trustworthiness under all circumstances, i.e. $m_Z^{A:B} \leq min\{m_B^A, m_Z^B\}$, because without other proof, the transitive trustworthiness would not be better than the original one. Note that, here we consider the scenario that node A only gets knowledge about node Z through node B. In the case that node A has additional paths to learn about node Z, it is possible that node A will have a higher trustworthiness about node Z than does node B.
- TPrinciple3: The closer the link to the evaluator, the stronger the influence it has on the transitive trust. This means c_B^A (m_B^A) has more weight in $c_Z^{A:B}$ ($m_Z^{A:B}$) than c_Z^B (m_Z^B).

3.4.2 Trust Aggregation

Trust aggregation is developed to summarize the propagated trust from multiple parallel trust paths. We use operator $\oplus$ to denote trust aggregation operation. For example, if node A has two parallel trust paths towards node Z, A-B-Z and A-C-Z, then the aggregated evaluation of node Z in node A's point of view via node B and node C is denoted as:

$$T_Z^{A:(B,C)} = T_Z^{A:B} \oplus T_Z^{A:C} \tag{3.4}$$

Principles of Trust Aggregation

Similar to trust transitivity, we list some desired principles for trust aggregation.

- APrinciple1: Aggregation may increase confidence if similar information is received from multiple paths, as it increases the volume of evidence; however, this principle may introduce vulnerability when a number of adversaries post the same misleading information to a victim.

- APrinciple2: Confidence may decrease if it contains contradictory information received from different paths. That is, a concrete positive trustworthiness and a concrete negative trustworthiness about the same target would produce a neutral but vague trust assessment.

- APrinciple3: Trustworthiness with higher confidence should have more influence on the aggregated trust than those with lower confidence.

Note that although we listed the above desired principles for trust transitivity and aggregation, as indicated by [2], not necessary all the principles will be satisfied by all the formulas we will demonstrate in the following section.

3.4.3 Calculating Uncertainty Based on Error Propagation Theory

Radius (or error) r of transitivity and aggregation operations can be calculated based on the error propagation theory. Given a set of variables which have error (or uncertainty), the error propagation theory (also called propagation of uncertainty) is used to calculate the error (or uncertainty) of a function of the variables [172]. Although we have not listed specific arithmetic functions or formulas for trust transitivity and aggregation here, we describe the general idea of how to calculate radius/error for them. Here we only take into account two trust tuples $T_1(m_1, c_1)$ and $T_2(m_2, c_2)$. It is easy to extend to more than two trust tuples. We represent transitivity or aggregation formulas in a general function as $f(m_1, m_2)$. Then the radius of function r_f can be computed as Equation 3.5 [172]:

$$r_f^2 = (\frac{\partial f}{\partial m_1})^2 r_1^2 + (\frac{\partial f}{\partial m_2})^2 r_2^2 + 2\frac{\partial f}{\partial m_1}\frac{\partial f}{\partial m_2} cov(m_1, m_2) \tag{3.5}$$

Here $cov(m_1, m_2)$ is the covariance between T_1 and T_2. In the case that T_1 and T_2 are independent, the covariance becomes zero.

We can see that the radius can be calculated for any format of arithmetic formulas using the error propagation theory. Therefore, our framework is very flexible and can be adapted to various transitivity and aggregation formulas. To summarize, error propagation is used to calculate confidence of trust propagation. In the later section, we will explore several different formulas for trust transitivity and aggregation.

3.5 Formulas for Trust Transitivity and Aggregation

In this section, we list some arithmetic formulas, which are widely used among computer scientists, for trust transitivity and aggregation. Remember that a trust tuple contains m and c in our framework. After defining formulas for m, confidence c and radius r can be computed accordingly by following the error propagation theory. Besides the formulas listed here, our framework can be adapted to other arithmetic formulas easily as long as they are derivable.

3.5.1 Transitivity Formulas

Following the principles inspired by human common sense, several formulas have been proposed to deal with trust transitivity. The three transitivity formulas listed below all have their specific focuses and make sense in some specific scenarios. Among them, it is very difficult to find out which one is the best, since trust propagation behavior patterns are different in many applications. We will show their performances on two real data sets in the later section.

Transitivity Formula One (TP1)

Multiplication is one of the most straightforward formulas used to compute trust transitivity in many existing works [2, 4, 13, 16, 62, 173]. We denote it as $TP1$ in this work, and represent it as:

$$m_B^A \otimes m_Z^B = m_B^A * m_Z^B \tag{TP1}$$

$$r_B^A \otimes r_Z^B = \sqrt{(m_Z^B)^2 * (r_B^A)^2 + (m_B^A)^2 * (r_Z^B)^2} \tag{TP1}$$

Note, when calculating the radius, we assume that m_B^A and m_Z^B are independent. This assumption applies for all the following formulas.

The idea of using multiplication for trust transitivity is that node B's recommendation about node Z will be discounted by node B's trustworthiness in node A's point of view. As $m \leq 1$, $m_B^A \otimes m_Z^B \leq min\{m_B^A, m_Z^B\}$, it satisfies TPrinciple1 listed in Section 3.4.1. Apart from discounting trust along the chain, multiplication can even filter out untrustworthy paths by setting a threshold. For example, in some cases, node A only considers suggestions from her/his trustworthy friends. This mechanism provides potential usage of defending attacks.

Transitivity Formula Two (TP2)

When considering trust transitivity, most likely friends of friends are also friends; however, it is more complex when considering an enemy's recommendations. In some

cases, the truster/evaluator discards those untrustworthy paths; however, distrust information may also be valuable for inferring indirect trust. One simple example is that enemies of enemies can be friends. To capture this idea, we list the second transitivity formula $TP2$ as in [62]. It is not guaranteed to satisfy any transitivity principles listed in Section 3.4.1.

$$m_B^A \otimes m_Z^B = m_B^A * m_Z^B + (1 - m_B^A) * (1 - m_Z^B) \quad \text{(TP2)}$$

$$r_B^A \otimes r_Z^B = \sqrt{(2 * m_B^A - 1)^2 * (r_Z^B)^2 + (2 * m_Z^B - 1)^2 * (r_B^A)^2} \quad \text{(TP2)}$$

In this case, friends of friends are still friends; however, if both m_B^A and m_Z^B are very low, which means A and B, B and Z are enemies, enemies of enemies result in friends too.

Transitivity Formula Three (TP3)

Another formula for trust transitivity is obtaining the minimum m of the chain as the transitive trust, which is represented as minimum t-norm [2]. Correspondingly, confidence associated with the minimum m is selected as the transitive confidence. If there are more than one link has the same minimum m, we select the minimum confidence among these links.

$$m_B^A \otimes m_Z^B = m_{min} = min(m_B^A, m_Z^B) \quad \text{(TP3)}$$

$$r_B^A \otimes r_Z^B = max(r_i \ \ where \ \ m_i = m_{min}) \quad \text{(TP3)}$$

The idea behind this formula is that trust will decrease as long as one of the links in the chain is very low. The minimum impression in this case is the bottleneck of the chain. Straightforwardly, it satisfies TPrinciple1.

3.5.2 Aggregation Formulas

In this section, we list five arithmetic aggregation formulas which are widely used in this field. Similar to transitivity formulas, their performances on various applications are also different.

Aggregation Formula One (AP1)

Given multiple parallel paths, one simple way to aggregate them together is to average them. It means that all paths are considered equally important, such as [57].

$$m_Z^{A:B} \oplus m_Z^{A:C} = \frac{m_Z^{A:B} + m_Z^{A:C}}{2} \tag{AP1}$$

$$r_Z^{A:B} \oplus r_Z^{A:C} = \sqrt{\frac{1}{2^2}((r_Z^{A:B})^2 + (r_Z^{A:C})^2)} \tag{AP1}$$

Note that, here we use two parallel paths as an example as well as in the following discussion. It is easy to extend them to more than two parallel paths cases.

Aggregation Formula Two (AP2)

Although averaging is very popular in several cases, it is not able to distinguish paths from each other. For example, paths can have different length and confidence. Under such situations, many researchers proposed to use the weighted mean [4, 16, 20, 62], in which paths are assigned with different weights accordingly. Generally, it can be written as:

$$m_Z^{A:B} \oplus m_Z^{A:C} = \frac{w_1 * m_Z^{A:B} + w_2 * m_Z^{A:C}}{\sum w_i} \tag{AP2}$$

$$r_Z^{A:B} \oplus r_Z^{A:C} = \sqrt{\frac{1}{(\sum w_i)^2}(w_1^2 * (r_Z^{A:B})^2 + w_2^2 * (r_Z^{A:C})^2)} \tag{AP2}$$

There are several ways to assign weights for paths. For example, weights are assigned according to confidence in [16]. Paths with higher confidence also have higher weights when compared with lower confidence paths. Also, weights can be assigned according to the value of trustworthiness m of the first hop, for example,

m_B^A and m_C^A in the above case [62]. The reason why the first hop is so important is that it is the only direct information that the evaluator has.

Aggregation Formula Three (AP3)

The third arithmetic aggregation formula is derived from the law of the probability of the union of two events [174]. The probability of the union event EA and EB, when EA and EB are independent, can be represented as:

$$\begin{aligned} P(EA \cup EB) &= P(EA) + P(EB) - P(EA \cap EB) \\ &= P(EA) + P(EB) - P(EA) * P(EB) \end{aligned}$$

Similarly, we list the third aggregation formula as in [40]:

$$m_Z^{A:B} \oplus m_Z^{A:C} = m_Z^{A:B} + m_Z^{A:C} - m_Z^{A:B} * m_Z^{A:C} \quad \text{(AP3)}$$

$$r_Z^{A:B} \oplus r_Z^{A:C} = \sqrt{(1 - m_Z^{A:C})^2 * (r_Z^{A:B})^2 + (1 - m_Z^{A:B})^2 * (r_Z^{A:C})^2} \quad \text{(AP3)}$$

This formula makes sense for applications which interpret trust as probability. It calculates the probability that at least one of two paths is trustworthy.

Aggregation Formula Four (AP4)

The forth aggregation formula is called the strongest path [175]. Among several parallel paths, the evaluator chooses the one which has the highest trustworthiness m. Correspondingly, that path's confidence will be selected as the aggregated confidence. In the cases where there are two or more parallel paths having the same highest m, it picks up the one having the highest c among them. Therefore, it is also called first trust then confidence.

$$m_Z^{A:B} \oplus m_Z^{A:C} = m_{max} = max(m_Z^{A:B}, m_Z^{A:C}) \quad \text{(AP4)}$$

$$r_B^{A:B} \oplus r_Z^{A:C} = min(r_i \ \ where \ \ m_i = m_{max}) \quad \text{(AP4)}$$

Aggregation Formula Five (AP5)

Instead of first trust then confidence, we can also aggregate trust first according to confidence, then trust [175]. In this scenario, the truster/evaluator prefers to select the path which has the highest confidence. Correspondingly, that path's trustworthiness m will be selected as the aggregated trustworthiness. This is a more conservative methodology when compared with the first trust then confidence methodology.

$$r_B^{A:B} \oplus r_Z^{A:C} = r_{max}, \;\; where \;\; c_{max} = max(c_Z^{A:B}, c_Z^{A:C}) \tag{AP5}$$

$$m_Z^{A:B} \oplus m_Z^{A:C} = max(m_i \;\; where \;\; c_i = c_{max}) \tag{AP5}$$

3.6 Validation Experiments and Results Analysis

In order to validate the accuracy and the potential usage of our trust management framework, we perform a series of experiments on two data sets. The first one was from a real world online social community – Epinions.com, which was collected by the authors of [176]. Another data set was collected from Twitter by us.

3.6.1 Data Sets Description

Epinions.com Data Set

Epinions.com is a general online customer review site. At Epinions.com, users can publish reviews regarding commercial products. Other users can rate the published reviews from 1 to 5 stars, which represent their opinions of the reviews from the least useful to the most useful respectively. Users are identified by IDs, so each user can only rate a review article, at most, one time. Also, users can express their propositions, i.e., trust judgment about other users with like, neutral, or dislike. Although other works [56,176] use propositions as trust, alternatively we use ratings to build up trust relationships in our experiments. We compare users' propositions with the average ratings between the corresponding pairs of users in Table 3.1. We can see that the

average ratings are coherent with subjective propositions. 72.6% of users rated their disliked users with low ratings (less than 3 stars). And to those whom users like, 95.2% of them gave very high ratings (more than 4 stars).

Table 3.1.
Average review ratings for three subjective propositions

Proposition	Total review	Average review ratings				
		(0,1]	(1,2]	(2,3]	(3,4]	(4,5]
Like	424336	4 (.0001%)	631 (.149%)	1667 (.393%)	18108 (4.267%)	403926 (95.190%)
Neutral	4365623	1429 (.033%)	180549 (4.136%)	452773 (10.371%)	992545 (22.735%)	2738327 (62.725%)
Dislike	45430	71 (.156%)	26357 (58.017%)	6525 (14.363%)	4288 (9.439%)	8189 (18.026%)

This data set contains 405, 154 distinct user IDs. Among them, 95, 318 users gave subjective propositions towards other users, and 120, 492 users rated review articles written by others. In total, 153, 265 users gave either ratings or propositions or both. On the other hand, 84, 601 users received subjective propositions from others, and 132, 586 users received ratings for their reviews. In total, 158, 143 users received either ratings or propositions or both. Based on our definition of trust for the Epinions.com data set (defined in Section 3.6.2), there are 78, 468 users having trust relationships.

Twitter Data Set

Twitter is a web-based micro-blogging service which has been in service since 2006. Many applications have been developed based on Twitter data, such as tracing disasters [177], stock market [79, 178], elections [179], and spam detection [180].

We collected our data set from Twitter in which all the users were the followers of a public stock market account named StockTwits. We first retrieved users' IDs and then used these IDs to retrieve their tweets, which were written in English. We developed an application using Twitter API as well as twitter4J library. Note that

Twitter API limits data collecting up to $3,200$ tweets from a single user's time line. The data set consisted of users' screen names, locations, tweets, and the date and time when they posted the tweets. We took a snapshot of users in that group in February, 2015, which had $401,052$ followers. And from the followers' time lines, we collected all the tweets posted before February 9th, 2015, for a total number of $38,748,723$ tweets. In addition to the followers, we also included users to whom the followers had posted interactive tweets. Based on our definition of trust for the Twitter data set (defined in Section 3.6.2), there are $2,067,284$ users having trust relationships.

3.6.2 Trust Modeling

The main goal of trust modeling is to evaluate trust metrics m and c from the raw data sets. As shown in Figure 1.1, this phase is context dependent. We separately deal with trust modeling for the Epinions.com and the Twitter data sets.

Trust Modeling for Epinions.com Data Set

As we indicated, we use ratings to synthesize the trustworthiness m. For a trust relation from user A to user Z, the trustworthiness m is the average of ratings that A rates Z's review articles. It is then converted into value in $[0, 1]$, as shown in Equation 3.6. Each rating is treated as one measurement. Following measurement theory, radius (or error) consists of two parts: Random Error (r_r) and Systematic Error (r_s). Random error is associated with the distribution of ratings around the mean. And systematic error is due to different components of the measuring system [155], such as external factors and measurement resolutions. In our framework, we only consider the measurement resolution. It is determined by the measurement scale. Finally, the error is combined in Equation 3.8.

$$m_Z^A = \frac{\sum_{i=1}^{i=N} rating_i}{5 * N} \tag{3.6}$$

$$r_r = \sqrt{\frac{\Sigma_{i=1}^{i=N}(x_i - \bar{x})^2}{N(N-1)}} \qquad r_s = \frac{scale}{2 * \sqrt{3}} \tag{3.7}$$

$$r = \sqrt{r_r^2 + r_s^2} \tag{3.8}$$

In our case, one star is equivalent to 0.2 when converted into the interval of $[0, 1]$. The *scale* in Equation 3.7 is 0.2. We use r to denote radius (or error), and confidence c can be derived based on r as defined in Equation 3.2. Also note that in Equation 3.7, N, which is the number of measurements, has to be greater than 1. For this reason we only consider the pairs of users which contains at least two ratings between them.

Trust Modeling for Twitter Data Set

To model trust for Twitter, like [181], as shown below, we will take into account several textual and behavioral features. Twitter allows users to post short tweets (140 characters maximum per tweet). Some types of tweets are designed for special purposes. They are mentions, replies, and retweets. One common feature of these three types of tweets is that they all contain the symbol "@", and all of them are used to tweet toward specific users and are considered as part of interactions (or conversations) among the users [182].

Since we want to know whether or not one user trusts another user, we need to evaluate her/his attitude towards the target user by analyzing the tweets that she/he posted towards the target user. Similar to Epinions.com, we treat each interactive tweet as one measurement. Tweets reflect users' opinions on persons, objects, or even aspects of objects. Similar to [183], we build up trust based on sentiment analysis results. There are some complex works about text sentiment evaluations [184]; however tweets are very short compared with regular documents. For simplicity, we assume that interactive tweets are always targeted at users they are posted towards. Tweets, based on their contents, can be divided into positive and negative tweets. We use SentiStrength [185] to analyze the sentiment result for each tweet in our data set. It

gives us a discrete value from -4 to $+4$ for each tweet. Then we convert it into the interval $[0, 1]$, using $\frac{sentiment+4}{8}$.

As we derive users' opinions based on sentiment analysis, it is very important for us to find an accurate sentiment analysis tool. Unfortunately such tools are very subjective. We select SentiStrength [185] for our experiments. We evaluate its performance on text reviews whose sentiments are known. We collected a data set from Yelp's 2014 competition, which provided both text messages and ratings [186]. In this data set, there were $1,125,458$ text reviews that users wrote towards the business (e.g. restaurants). Associated with the text reviews, users also gave ratings from 1 star to 5 stars ($335,022$ reviews which contain both text and ratings). It is reasonable to assume that the users' text reviews are consistent with their star ratings. For example, if a user writes a negative text review for a restaurant, most likely the rating associated with the corresponding text review is also very low (for example, 1 or 2 stars).

In order to compare the sentiment analysis results with the users' ratings, we convert them into the same interval $[0, 1]$. SentiStrength distinguishes sentiment results using eight discrete values, from -4 to $+4$. While users' ratings are expressed from 1 star to 5 stars, the conversion can be found in Equation 3.9. We denote v as the converted value, *rating* and *sentiment* represent star ratings and sentiment results correspondingly. As we can see, 1 star and -4 (in sentiment result) correspond to $v = 0$, 5 star and $+4$ correspond to $v = 1$. The result shows us that SentiStrength's sentiment results are very close to the users' ratings (the mean absolute error is equal to 0.8972 star).

$$v = \frac{rating - 1}{4} \qquad v = \frac{sentiment + 4}{8} \tag{3.9}$$

Having sentiment result for each tweet, we could now calculate the trustworthiness m by treating each tweet as one measurement. Instead of just averaging all the tweets, we divide them into different windows based on the time line since people interact with each other in different periods of time. We try to capture these time-based characteristics of human behavior. We cluster the tweets posted in the same month

into the same window. As we will see later, we treat these windows differently. In each window, we group tweets based on days. In each day, we calculate the mean trustworthiness m_d for that specific day, as well as c_d. Note that the *scale* in this case is 0.125, as SentiStrength returns discrete values from -4 to $+4$.

After calculated m_d and c_d for each day, we use the weighted mean to combine the results for each month in Equation 3.10.

$$m_{month} = \frac{\Sigma_{i=1}^{i=31} w_i m_i}{\Sigma_{i=1}^{i=31} w_i} \tag{3.10}$$

Here we use $w_i = \frac{1}{r^2}$, to assign higher weights to those which have higher confidence (or smaller errors). And the error of weighted mean is expressed as Equation 3.11.

$$e_{month}^2 = \frac{1}{\Sigma_{i=1}^{i=31} w_i} \tag{3.11}$$

Having calculated trustworthiness and error for each window for one month, we then combine them together. In each window, as in one period of time, the truster/evaluator has an impression on the target; however, this impression faded with time. For example, if the truster/evaluator just evaluated the target a few days ago, she/he may be quite sure about her/his trustworthiness assessment; however, if the evaluator "measured" the target several months ago, the impression has somehow faded. So we introduce a forgetting factor σ, where σ is less than 1, to capture this effect on the users' confidence. Also, because of the forgetting effect, we only focus on tweets which were posted in 2014. Therefore, we have 12 windows in total (from January to December). The confidence of December, which is the latest month, is not discounted. The confidence of November is discounted by σ, and the confidence of October is discounted by σ^2, and so on, as shown in Equation 3.12, where i is the number of the corresponding month (i.e. $i = 1$ for January and $i = 2$ for February).

$$c_i^{'} = c_i * \sigma^{12-i} \tag{3.12}$$

Similarly, we combine all the windows' results using the weighted mean where weights are their confidence $c^{'}$. We select one month as the length of time window and the forgetting factor $\sigma = 0.9$ in this chapter. Further refinement of these parameters will be part of our future work.

3.6.3 Validation Experiments

To measure the accuracy of our trust management framework, we use the leave-one-out method to compare predicted indirect trust with actual trust, as shown in Figure 3.2. To predict A's trust about Z, we remove the actual direct trust link from the network and keep all other trust paths. We then use trust transitivity and aggregation formulas to infer indirect trust and compare it with the removed actual trust.

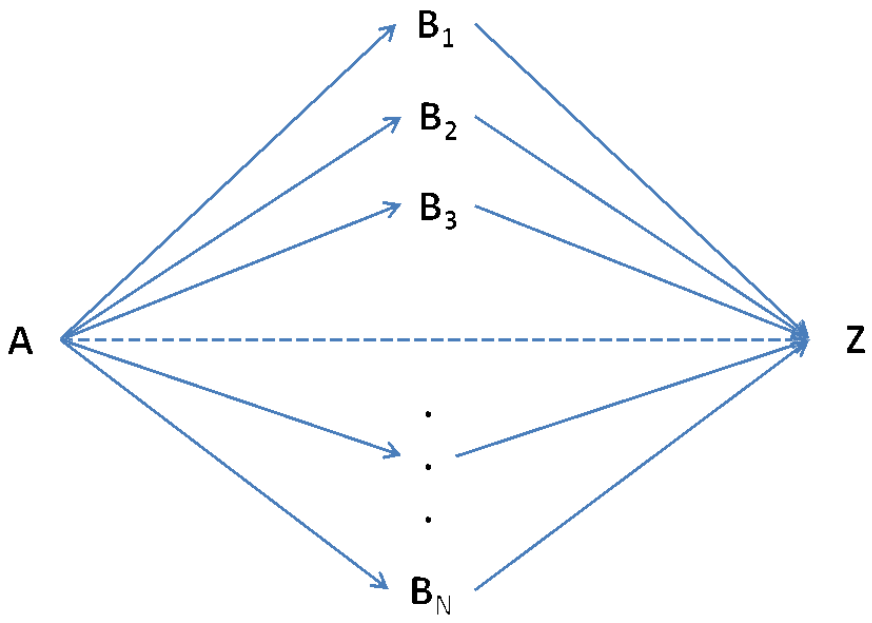


Figure 3.2. Predicting indirect trust with leave-one-out method

For $AP2$, we use $w = c$ in the following experiments, which is the same as in [16]. Accuracy is measured by classical mean absolute error (MAE) and classical root mean square error (RMSE). We use $diffm$ to represent the absolute difference between the inferred m and actual m, and $diffc$ for the absolute difference between the inferred c and actual c accordingly. Additionally, to consider $diffm$ and $diffc$ together, we

also measure MAE and RMSE for Manhattan distances, which is defined in Equation 3.13. Note, the interval of MAE of Manhattan distances is $[0, 2]$.

$$MAE(Man) = \frac{\sum_{i=1}^{n} |diffm_i| + |diffc_i|}{n} \tag{3.13}$$

3.6.4 Result Analysis

In this section, we show the performances of different combinations of transitivity and aggregation formulas using two data sets. As [4] points out, inferred indirect trust becomes unreliable when the length of the chains increases. We only take into account the chains containing two hops (contain only one intermediate node). We will see their performances in cases which have three hops in the later section. In addition to 15 possible combinations, we also add a baseline methodology, in which we randomly assign values in $[0, 1]$ for inferred m and c.

Table 3.2.
Formulas' performances on the Epinions.com data set (two hops)

Formulas	MAE (diffm)	RMSE (diffm)	MAE (diffc)	RMSE (diffc)	MAE (Man)	RMSE (Man)
TP1,AP1	0.0565	0.0827	0.1021	0.1170	0.1585	0.1433
TP1,AP2	0.0596	0.0924	0.0353	0.0630	0.0948	0.1118
TP1,AP3	0.0620	0.1250	0.1357	0.1476	0.1977	0.1934
TP1,AP4	***0.0511***	***0.1036***	***0.0514***	***0.0653***	***0.1026***	***0.1224***
TP1,AP5	0.1815	0.2497	0.0427	0.0614	0.2242	0.2572
TP2,AP1	0.0554	0.0808	0.1049	0.1197	0.1603	0.1444
TP2,AP2	0.0589	0.0906	0.0371	0.0659	0.0960	0.1121
TP2,AP3	0.0619	0.1248	0.1359	0.1476	0.1978	0.1933
TP2,AP4	***0.0510***	***0.1032***	***0.0501***	***0.0629***	***0.1012***	***0.1208***
TP2,AP5	0.2063	0.2656	0.0463	0.0705	0.2526	0.2748
TP3,AP1	0.0526	0.0783	0.1070	0.1215	0.1597	0.1446
TP3,AP2	0.0553	0.0870	0.0531	0.0750	0.1084	0.1149
TP3,AP3	0.0619	0.1249	0.1354	0.1478	0.1973	0.1935
TP3,AP4	***0.0513***	***0.1038***	***0.0265***	***0.0629***	***0.0778***	***0.1214***
TP3,AP5	0.0559	0.1106	0.0257	0.0621	0.0816	0.1268
Baseline	0.4526	0.5346	0.3823	0.4643	0.8349	0.7081

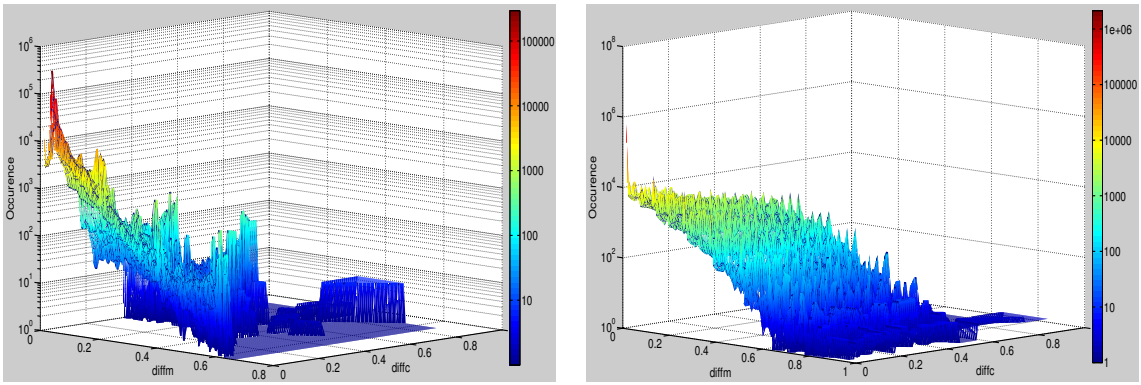
Table 3.3.
Formulas' performances on the Twitter data set (two hops)

Formulas	MAE (diffm)	RMSE (diffm)	MAE (diffc)	RMSE (diffc)	MAE (Man)	RMSE (Man)
TP1,AP1	0.2336	0.2527	0.1559	0.1946	0.3895	0.3189
TP1,AP2	0.2342	0.2533	0.1552	0.1937	0.3894	0.3189
TP1,AP3	0.2209	0.2475	0.1483	0.1838	0.3693	0.3083
TP1,AP4	0.2207	0.2409	0.1457	0.1800	0.3665	0.3008
TP1,AP5	0.2386	0.2590	0.1536	0.1923	0.3922	0.3226
TP2,AP1	0.1037	0.1456	0.2729	0.3250	0.3766	0.3562
TP2,AP2	0.1038	0.1457	0.2729	0.3252	0.3767	0.3563
TP2,AP3	0.1556	0.2198	0.2758	0.3279	0.4314	0.3948
TP2,AP4	0.1047	0.1468	0.2670	0.3192	0.3717	0.3514
TP2,AP5	0.1037	0.1460	0.2783	0.3305	0.3820	0.3613
TP3,AP1	***0.0692***	***0.1220***	***0.1224***	***0.2087***	***0.1916***	***0.2418***
TP3,AP2	***0.0694***	***0.1224***	***0.1238***	***0.2115***	***0.1932***	***0.2444***
TP3,AP3	***0.1133***	***0.1909***	***0.1267***	***0.2156***	***0.2400***	***0.2879***
TP3,AP4	***0.0669***	***0.1218***	***0.1185***	***0.2103***	***0.1854***	***0.2430***
TP3,AP5	***0.0701***	***0.1258***	***0.1255***	***0.2188***	***0.1955***	***0.2524***
Baseline	0.3620	0.4416	0.3665	0.4472	0.7285	0.6285

We find $1,449,750$ leave-one-out cases (or triads) in the Epinions.com data set, and $4,791,751$ triads in the Twitter data set. Table 3.2 shows their performances on the Epinions.com data set, and Table 3.3 shows their performances on the Twitter data set. Note that three transitivity and five aggregation formulas which are from existing works represent corresponding behavior patterns. Therefore, Table 3.2 and 3.3 compare different behavior patterns' performance in two online communities.

In both data sets, we can see that 15 combinations of formulas, inspired by some principles, perform much better than the baseline methodology. To better illustrate the prediction accuracy on two data sets, for the combination of $TP3$ and $AP2$ formulas, we show $diffm$ and $diffc$ in both data sets in Figure 3.3. We divide $diffm$ and $diffc$ into small cells. Each cell has its length ($\Delta diffm$) and width ($\Delta diffc$) equal to 0.01, i.e. $0.00 \leq diffm < 0.01$ and $0.10 \leq diffc < 0.11$, which results in total $10,000$ cells. We then count the number of triads in each cell. From

them we can see that most triads have very small $diffm$ and $diffc$ at the same time, i.e. $diffm < 0.1$ and $diffc < 0.1$.



(a) Epinions.com data set

(b) Twitter data set

Figure 3.3. Occurrence of diffm and diffc in two data sets using TP3 and AP2 formulas

These formulas achieve different prediction accuracy on two data sets. As we can see, overall, the performances on the Twitter data set is not as good as on the Epinions.com data set. One possible reason is that ratings in Epinions.com are given by users themselves, and they reflects the users' real preferences. While, in Twitter, we use a common sentiment analysis tool to evaluate all the users' tweets. First of all, it is very difficult to assess sentiment analysis tools as their outputs are subjective. Although we test SentiStrength on Yelp's data set, Twitter may exhibit different properties from Yelp. Second, we use the same criteria to assess all the tweets without considering their authors' different preferences. Remember that human trust is subjective, which means different users can have different feelings even when they write the same texts. This is captured in Epinions.com, because ratings are based on the users' own preferences.

Apart from this, we note one interesting phenomenon in our experiments. In the Epinions.com data set, the aggregation formula plays a dominant role in their performances. As long as we select the aggregation formula, the results change only a little even if we try different transitivity formulas. However, in the Twitter data set

transitivity formulas dominate the performances. This phenomenon, so far, is still unexplained.

Furthermore, one important observation from our experiments is that formulas perform differently on the two data sets. For example, $TP1$ and $AP4$ achieve a very good accuracy in the Epinions.com data set compared with other formulas. In the Twitter data set, their performance is worse than some other formulas. This indicates that formulas which have different views or focuses on different specific aspects of trust, such as a conservative view vs. an optimistic view, perform differently. This is because data sets, or applications themselves, have biased trends. It is possible that one formula's underlying meaning fits this application very well, but does not make sense for others. In other words, whether or not formulas can perform well depends on if they match the applications trust propagation patterns. Therefore, instead of proposing specific formulas, we propose a fundamental framework which can be adapted by many different formulas.

3.6.5 Filtering Paths by Confidence

As many existing works [4, 20] suggest, when there exist multiple paths between the truster and the trustee, it is important to select trustworthy paths to consider in trust aggregation. TidalTrust [4] selects the strongest paths. And MoleTrust [20] sets a threshold, and only paths whose trust values are above the threshold are taken into account in trust aggregation.

In this chapter, we propose a simple approach to calculate confidence based on error (Equation 3.2). Apart from trustworthiness evaluation, confidence provides information about how certain that evaluation is. To show one of the benefits of using confidence, as in TidalTrust and MoleTrust, we use weighted mean for trust aggregation. Besides using trustworthiness (m) as a selection criteria, we also use confidence (c) as a selection criteria. In Equation 3.14, j represents nodes which have

direct trust relationship with s, and are reachable for i. Basically, we add confidence as an additional selection criteria in our approach.

$$m_s^i = \frac{\sum_{m_j^i \geq th \ \& \ c_j^i \geq th} m_j^i * m_s^j}{\sum_{m_j^i \geq th \ \& \ c_j^i \geq th} m_j^i} \tag{3.14}$$

We compare our approach to TidalTrust and MoleTrust. As the Twitter data set is more sparse than the Epinions.com data set, here we only try our approach on the Epinions.com data set. Also, for space limitation, we only do experiment for two hops cases. Figure 3.4 shows us that by using confidence as an additional factor, predicted results are more accurate. Especially, in the area of high confidence, which is the reliable area for decision making in many applications, our approach performs better than TidalTrust and MoleTrust. Here, x axis is the value for different th.

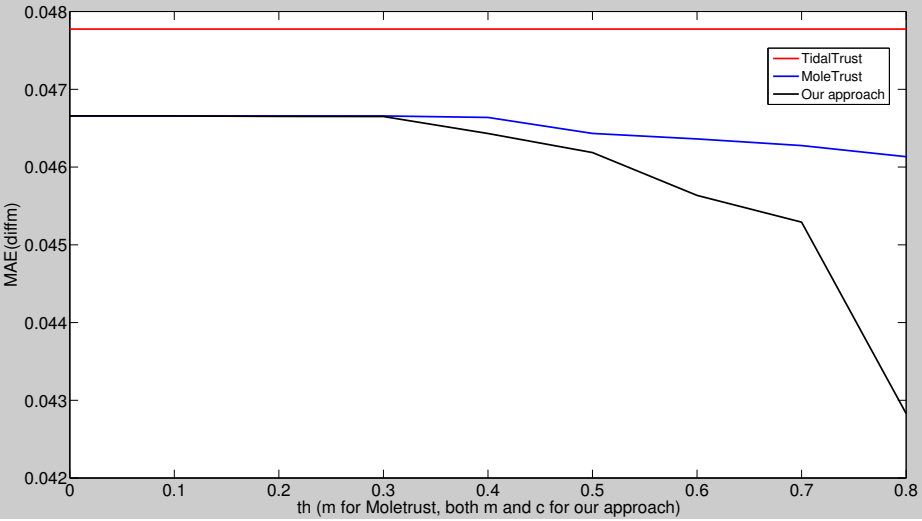


Figure 3.4. Prediction comparison among TidalTrust, MoleTrust and our approach on the Epinions.com data set (two hops)

There are some existing works which also consider confidence, such as [72], [112] and [26]. Similar to [72], the definition of confidence in our framework also captures two important intuitions or properties mentioned in [72]. However, we think that our definition of confidence is computationally cheaper than [72] and [112]. In [72], authors use a binary search algorithm to convert between evidence space and trust

space, as there is no closd-form solution for it. In our approach, we derive confidence from the error in the measurement theory, which is simpler than [72] and [112]. More important, for different types of transitivity (i.e. discounting) and aggregation formulas, it is easy to calculate the propagated error by following the error propagation theory [162].

3.7 Coverage

As we stated earlier, one of the main purposes of inferring indirect trust is to allow more pairs of users to be connected given the original sparsely connected networks. This is especially useful for many applications, such as recommender systems, in which many users have only a limited number of direct contacts. We use the term coverage to measure how many pairs of users are connected within a specific number of hops.

3.7.1 Coverage vs. Number of Hops

It is obvious that more pairs of users can be connected if we predict indirect trust for a larger number of hops. In this section, we quantitatively show how the number of hops can affect the coverage. Due to time limitation, we only calculate the coverage within one hop (directly connected users), two hops, and three hops. Table 3.4 shows the coverage results of two data sets. Note that, the coverage within two hops also contains pairs of users within one hop, and the same rule applies to three hops.

Table 3.4.
The coverage in two data sets

Triads	connected pairs in the Epinions.com	connected pairs in the Twitter
one hop (direct)	1,530,103	6,829,998
two hops	152,795,175	65,131,606
three hops	977,171,805	833,540,419

We can see that in the Twitter data set, the coverage within two hops only increases by less than 10 times compared with the coverage within one hop. It increases 120 times when we extend it to three hops. In the Epinions.com data set, when we increase the length of the chains to three hops, the coverage increases more than 630 times compared with the one hop case. This difference is caused by the different topologies of the two online social communities. We measure the density of two communities by $\frac{|E|}{|V|(|V|-1)}$ [187]. Here $|E|$ is the number of edges, and $|V|$ is the number of nodes in communities. We can see that the Epinions.com data set (density is $2.4851 * 10^{-4}$) is much denser than the Twitter data set (density is $6.8967 * 10^{-8}$). Apart from this, in the Twitter data set, there are $4,447$ connected sub-communities or sub-graphs (composed by inter-connected nodes using trust relationships), and most sub-communities only contain 2 or 3 users. Although its average size is 464.87, among $2,067,284$ users, $2,018,469$ of them are leaf nodes in the Twitter data set. This is because we include the users to whom StockTwits's followers posted interactive tweets; however these users' tweets are not collected as they are not part of the stock group. After removing the leaf nodes, its average size becomes 10.98. In other words, the Twitter data set is very sparsely connected; however, in the Epinions.com data set, we find 390 sub-communities with their average size equal to 201.19 (it does not contain leaf nodes), where the sub-communities' average size is much larger than in the Twitter data set (after removing the leaf nodes). Details about sub-communities statistics in two data sets can be found in Table 3.5.

Table 3.5.
Sub-communities' statistics of two data sets

Community	Number of communities	Maximum community size	Minimum community size	Average community size	Number of leaf nods	Average community size without leaf nodes
Epinions.com	390	77,540	2	201.20	0	201.20
Twitter	4,447	2,055,406	2	464.87	2,018,469	10.98

In conclusion, coverage is affected not only by the length of the chains, but also by the networks topologies. Within densely connected networks, inferring indirect trust can help to cover more pairs of users than in sparsely connected networks.

3.7.2 Coverage vs. Accuracy

By increasing the length of the chains, more users become connected; however, on the other hand, it may sacrifice prediction accuracy. To see how the length of the chains can affect the prediction accuracy, we do leave-one-out experiments on two data sets for three hops cases, in which each triad contains exactly three hops. In other words, user A and user Z are only connected by paths which contain two intermediate users.

Table 3.6.
Formulas' performances on the Epinions.com data set (three hops)

Formulas	MAE (diffm)	RMSE (diffm)	MAE (diffc)	RMSE (diffc)	MAE (Man)	RMSE (Man)
TP1,AP1	0.0840	0.1089	0.1322	0.1441	0.2162	0.1806
TP1,AP2	0.1016	0.1465	0.0328	0.0622	0.1344	0.1591
TP1,AP3	0.0642	0.1322	0.1400	0.1510	0.2043	0.2007
TP1,AP4	***0.0566***	***0.1158***	***0.0768***	***0.0848***	***0.1334***	***0.1435***
TP1,AP5	0.5775	0.6104	0.0428	0.0701	0.6203	0.6144
TP2,AP1	0.0841	0.1094	0.1324	0.1442	0.2165	0.1810
TP2,AP2	0.1014	0.1463	0.0.0332	0.0621	0.1346	0.1589
TP2,AP3	0.0643	0.1314	0.1404	0.1515	0.2047	0.2005
TP2,AP4	***0.0564***	***0.1163***	***0.0773***	***0.0857***	***0.1337***	***0.1444***
TP2,AP5	0.5748	0.6069	0.0464	0.0734	0.6212	0.6113
TP3,AP1	0.0724	0.0954	0.1342	0.1461	0.2066	0.1475
TP3,AP2	0.0865	0.1274	0.0554	0.0786	0.1419	0.1497
TP3,AP3	0.0642	0.1316	0.1402	0.1513	0.2045	0.2005
TP3,AP4	***0.0567***	***0.1158***	***0.0268***	***0.0637***	***0.0835***	***0.1322***
TP3,AP5	0.0847	0.1513	0.0260	0.0632	0.1107	0.1640

Now, we increase the length of the chains to three hops. Table 3.6 and Table 3.7 show the formulas' performances on two data sets separately. For time efficiency, we

randomly select $10,000$ three hops triads (repeated 10 times) from the Epinions.com data set. From these two tables, we can see that formulas follow the same performance patterns in three hops triads as they do in two hops triads. But obviously, their prediction accuracy is not as good as in two hops triads.

Table 3.7.
Formulas' performances on the Twitter data set (three hops)

Formulas	MAE (diffm)	RMSE (diffm)	MAE (diffc)	RMSE (diffc)	MAE (Man)	RMSE (Man)
TP1,AP1	0.3656	0.3921	0.2734	0.3347	0.6390	0.5155
TP1,AP2	0.3671	0.3941	0.2522	0.3131	0.6193	0.5033
TP1,AP3	0.3292	0.3785	0.2415	0.3004	0.5707	0.4832
TP1,AP4	0.2922	0.3296	0.2026	0.2521	0.4948	0.4149
TP1,AP5	0.4010	0.4273	0.2667	0.3271	0.6677	0.5382
TP2,AP1	0.2547	0.2869	0.2962	0.3575	0.5510	0.4584
TP2,AP2	0.2555	0.2876	0.2826	0.3450	0.5381	0.4491
TP2,AP3	0.3357	0.3881	0.2930	0.3543	0.6287	0.5255
TP2,AP4	0.2174	0.2541	0.2324	0.2901	0.4498	0.3856
TP2,AP5	0.2587	0.2911	0.3082	0.3688	0.5669	0.4698
TP3,AP1	***0.1440***	***0.1843***	***0.2367***	***0.2981***	***0.3806***	***0.3505***
TP3,AP2	***0.1465***	***0.1883***	***0.2177***	***0.2764***	***0.3642***	***0.3345***
TP3,AP3	***0.3550***	***0.4098***	***0.2692***	***0.3332***	***0.6242***	***0.5282***
TP3,AP4	***0.1319***	***0.1765***	***0.2143***	***0.2746***	***0.3462***	***0.3265***
TP3,AP5	***0.1467***	***0.1930***	***0.2508***	***0.3167***	***0.3976***	***0.3709***

3.7.3 Coverage vs. Confidence

In this section, we explore the relationship between the coverage and confidence. As the Epinions.com data set is denser than the Twitter data set, we only do experiment on the Epinions.com data set, which contains $158,143$ users. Among them, there are $158,143 * 158,143 = 25,009,208,449$ possible pairs of users.

The relationship between the coverage and confidence is shown in Figure 3.5. Note that the y axis is in logarithmic, and the x axis denotes the desired inferred confidence. For example, if we have $x = 0.5$, only those triads whose inferred confidence

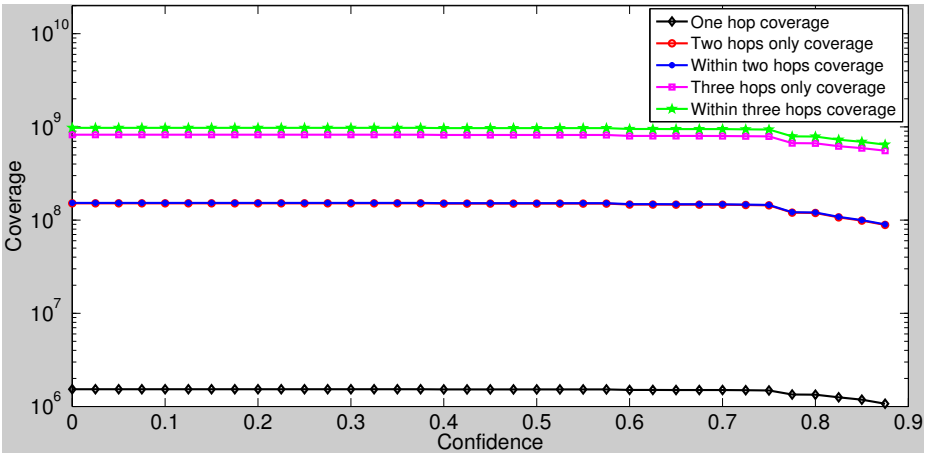


Figure 3.5. Relation between desired confidence levels and the coverage

is greater than or equal to 0.5 will be counted. Overall, we can see that when requiring higher confidence levels, less pairs of users can be connected. Our results show that the two hops only cases (does not include one hop) coverage is two magnitudes higher than the one hop cases coverage. The coverage of the three hops only cases is one more magnitude higher than the two hops only cases coverage. Such results could be used by various applications in online social communities to explore tradeoffs between the coverage and corresponding levels of confidence. For example, we could increase the number of receivers of a given recommendation depending on the desired level of confidence, from which we will determine the chances of success of that recommendation.

3.8 Chapter Summary

We developed a measurement theory-based trust management framework that aims to provide an intuitive way to represent and manage cognitive trust. For cognitive trust, we introduced two trust metrics: trustworthiness/impression and confidence. On one hand, these metrics are intuitive and on the other hand, they are similar to measured value and the error used in measurement theory. Using the cog-

nitive trust concept, we established trust networks among users in two real online social communities.

Based on the proposed trust management framework, we adapted some widely used transitivity and aggregation formulas to our scheme. Our framework associated with these formulas can be used to infer indirect trust relationships among unconnected users in sparsely connected networks. We showed with experiments on two real online social communities data sets the validity of our framework, as well as its enormous potential usage in various social network applications. Our results showed coherence with [2], that no single formula can guarantee very good performances in all applications, as users in different communities and applications have different behavior patterns. Our framework is significantly important because it serves as an underlying fundamental for other schemes which focus on specific formulas. Also, we showed that by using confidence as additional information, our approach can perform better than two existing works.

4 USING TWITTER TRUST NETWORK FOR STOCK MARKET DATA ANALYSIS

4.1 Introduction

Online social media (e.g. Twitter) is becoming more popular, as it is easier for users to post and spread information than with traditional media. With more users joining in online social networks, more data is available. Therefore, many data-driven applications, such as disaster detection [177], election predictions [179, 188], information filtering [189], opinion mining [190–192] and so on benefit from this trend. Among them, financial market analysis is one of the most attractive fields and has attracted a lot of attention [79, 178, 193–195].

The stock market is a very hot topic in the field of finance and economics. Many researchers try to analyze and predict stock returns based on various types of theories [196, 197]. For example, Chartist theory [198] assumes that the stock market's past behavior patterns will recur in the future. Thus we can predict future stock returns by using historical data. In contrast to Chartist theory, Random Walk theory [199] considers stock returns as identical independent variables. Although these theories' assumptions are different, many existing works use historical stock market data, such as open price, close price, daily trade volume and so on, to predict future stock returns.

Besides historical stock market performance, investors' decisions can be affected by news [200] and media [193, 194, 201–203]. Also, public mood or sentiment which is reflected in media plays an important role in investors' decision making processes [204, 205]. Investors' decisions in turn can affect stock market. Therefore, stock market is related with public mood in news or media.

With the popularity of Twitter and its easy-to-use open Application Programming Interfaces (APIs), there exist many works that use Twitter as a platform to analyze and predict stock market activities, including both indicator-level and firm-level analysis [79, 178, 195, 206, 207]. In addition to academic researchers, firms are also paying attention to Twitter for their commercial purposes. Many firms use Twitter to interact with their investors and customers [208]. Compared with traditional media, Twitter is efficient. To use Twitter to analyze stock market, typically Twitter feeds (tweets) are first analyzed by sentiment analysis tools to extract their sentiment, then tweet sentiments are aggregated together. Aggregated Twitter sentiment valence is then used for financial market analysis. Most widely used sentiment analysis tools generate binary results (positive or bullish vs. negative or bearish), although some sentiment analysis tools can generate more complicated results, such as multi-level sentiment results.

The main hypothesis of this work is that the users' reputation, built by the inter trust among them, using our trust management system, helps in making better decisions of the stock market investors. To verify this hypothesis and to validate our trust management system, we collect stock market-related data from Twitter to see the correlation between Twitter sentiment valence and abnormal stock returns. Therefore, the correlation between Twitter sentiment valence, filtered by our trust management system, and abnormal stock returns served as ground truth for our trust management system. We select eight firms which are the top eight mentioned firms (which have the largest number of tweets) in our data set. The reason we select these eight firms is that, for other firms, the average number of daily tweets is low. Based on only a small number of tweets, we think that the analysis result is not reliable. For the selected eight firms we collect their stock market data correspondingly from Yahoo! Finance. As indicated in [209], the source (users) of tweets is also an important factor. Therefore, unlike many existing works which treat all the authors equally important or ignore authors' identities, in addition to analyzing tweets' sentiment, we also take into account tweets' authors. We adapt our measurement theory based trust

management framework [210] and construct a user-to-user trust network for Twitter users based on their tweeting behaviors. Then, users are differentiated by their reputation or power in the whole community, where reputation or power is determined by the user-to-user trust network. Furthermore, to aggregate tweets together for Twitter sentiment valence, each tweet is weighted by its author's power.

To compare our approach to other ones, we use the Pearson correlation tests among results for eight months time (the trading days from 01/01/2015 through 08/31/2015). Compared to treating all the authors equally important or weighting them by their number of followers, our trust network based reputation mechanism amplifies the correlation between a specific firm's Twitter sentiment valence and the firm's stock abnormal returns. To further consider the possible auto-correlation property of abnormal stock returns and to test the relation between Twitter sentiment valence and abnormal stock returns, we construct a linear regression model, which includes historical stock abnormal returns. Again, our results show that by using our trust network power based method to weight tweets, Twitter sentiment valence reflects abnormal stock returns better than other two methods, that is treating all the authors equally important or weighting authors by their number of followers.

The remaining portion of this chapter is organized as follows: in Section 4.2, we introduce some background knowledge and literature works in this field. In Section 4.3, we introduce our trust management framework and adapt it to Twitter. Also, we propose a simple method to calculate for users' power or reputation. In Section 4.4, we illustrate how we aggregate Twitter sentiment valence for the firms. And we propose our trust network power based method as well as other two baseline methods. In Section 4.5, we give detailed information about the data sets we used in this work. Also, we compare our trust network power based method with other two baseline methods regarding Pearson correlation coefficients and a linear regression model. In Section 4.6, we conclude this chapter and list several limitations of applicability of this work as well as some potential future work.

4.2 Background and Related Works

Twitter, as one of the most popular online social media platforms, provides its users the ability to share and spread their opinions. It also enables users who have the same interests to form groups. The stock market is among one of the hottest topics among Twitter users. There are many stock market-related groups or gurus on Twitter, such as StockTwits, FinancialTimes, MarketWatch, and so on. Recent research has shown that investors are likely to post financial news or articles and share their opinions on Twitter [211]. Compared with traditional media, Twitter feeds can be incorporated instantly into stock prices. Therefore, Twitter has become a widely used platform for researchers to analyze and predict stock returns.

As [193, 201, 202] pointed out, investors' emotions or sentiments can be reflected by the stock market. Negative sentiment or pessimism on social media might induce a stock price to drop. Positive sentiment is more likely to induce stock prices to increase than neutral or pessimism sentiment. Therefore, given users' text (tweets), natural language processing methods are needed to analyze investors' emotions. There exist many sentiment analysis tools. Roughly, they can be divided into two categories: word count analysis strategy and machine learning based strategy. Word count analysis strategy uses dictionaries to determine sentiment for each word and then aggregate words' sentiment together. Most commonly used dictionaries in this field include Harvard-IV dictionary [212] and Loughran and McDonald's financial dictionary [213]. Among machine learning methods, most of them are classifiers, such as Naive Bayes classifier, SVM classifier, and so on. One of the problems of the machine learning based strategy is that it requires a set of labeled training data, which might need a huge load of manual work. In this work, we use an existing sentiment analysis tool – SentiStrength [185], which is designed for short informal text.

Twitter sentiment valence is then measured based on the detected positive and negative tweets. Various Twitter sentiment valence measurements are used in the literature [193, 195, 211, 214]. In principle, Twitter sentiment valence measures the

ratio of positive tweets to negative tweets. To investigate the linear relation between Twitter sentiment valence and stock prices or stock returns, Pearson correlation coefficients [79, 211, 215] and beta coefficients of linear regression models [178, 195, 214] are widely used in the literature.

Existing works in this field can be divided into two categories based on their focus. Indicator-level works mainly focus on indicators, such as Dow Jones Industrial Average Index, NASDAQ, S&P 500 index, and so on. This type of work focuses on the whole industry. Indicator-level works include [178], [193], [216], [217], and so on. More recently, researchers are also paying much attention to firm-level works; as the name itself indicates, instead of investigating the whole industry, this type of work focuses on specific firms. [215], [195], [214], [211], [218] and [219] belong to firm-level works. In this chapter, we focus on specific firms.

Bollen et al., used OpinionFinder and Google-Profile of Mood States (GPOMS) to measure sentiment for tweets [178]. Rather than outputting binary sentiment results (OpinionFinder), GPOMS measures sentiment in six dimensions, which includes calm, alert, sure, vital, kind, and happy. And it showed that only calm is related to Dow Jones Industrial Average Index. Tetlock [193] did experiments with Wall Street Journal, and mainly focused on the pessimism score of the media. It showed that high media pessimism scores caused the drop in stock market prices. In [217], authors classified tweets into fear, worry, and hope based on the corresponding words. It showed that Twitter sentiment (fear, worry, and hope) is negatively correlated with Dow Jones Industrial Average Index, NASDAQ and S&P 500 index. Similarly, [216] measured anxiety, worry, and fear in LiveJournal, and it turned out that they were negatively related to the S&P 500 index.

Smailovic et al., [214] calculated positive sentiment probabilities by dividing the number of positive tweets by the total number of tweets. It then analyzed eight firms' stock returns and their positive sentiment probabilities by using the Granger causality test [220]. Instead of focusing on stock returns, Ranco et al., [215] measured the Pearson correlation coefficient between 30 firms' abnormal returns and their Twitter

sentiment valence. Its focus was on event detection and events' relation with abnormal returns. Similarly, Sul et al., [195] measured the relation between Twitter sentiment valence and abnormal returns via analyzing beta coefficients of linear regression models. In [211], besides investigating the relation among Twitter sentiment valence, stock returns, message volume and trading volume, Sprenger et al., showed that tweets from users who always provide good advice are more likely to be retweeted more than other users.

However, none of those mentioned above works took into account tweets' authors. Even in [211], it only investigated the relation between the advice quality and the number of retweets. It did not differentiate their authors. In the remainder of this chapter, given historical Twitter data, we adapt our trust management framework [210] to measure user-to-user trust relationships and then construct a trust network for the whole community. Based on the trust network within the community, we then derive users' reputation or power, which is used later as weights in the process of Twitter sentiment aggregation. Through weighting each tweet by its author's reputation or power, we can amplify the correlation between specific firms' Twitter sentiment valence and their abnormal stock returns. Also, we show that to get reliable analysis, a sufficient number of tweets must be available.

4.3 Trust Network for Twitter

Trust, which is a subjective concept, plays an extremely important role in people's daily lives. Actually, we use our trust estimations or trust networks to make decisions in our lives [9, 154]. For example, among several service providers, we might choose the one who has the highest rating. On Twitter, given a huge number of subscribers, trust is very important for users to differentiate among other users. A user might have thousands of followers or friends on Twitter; however, not all of them are acquaintances. Huberman et al., [8] differentiated "claimed friends" from "real friends" on Twitter by counting the number of interactive tweets that two users post towards

each other. To handle trust on large online social media sites, such as Facebook and Twitter, we need support from computer systems. Therefore, we have to represent trust computationally, which we call trust modeling in this work. A lot of research has been done in this field [111, 121, 157].

Typically, only very few users are directly connected through trust network on Twitter [22]. Therefore, besides representing trust in a computational way, we also need a framework that can effectively infer Friend of a Friend (FOAF) relationships, as user-to-user direct trust relationships are not sufficient in sparsely connected online social networks. In this work, we call this trust inference. In [210], we proposed a measurement theory based trust management framework which addresses both trust modeling and trust inference. We represent our trust management framework in Figure 1.1.

4.3.1 Trust Components and Trust Modeling for Twitter

Our trust management framework is based on measurement theory. As we do in Section 3.6.2, we treat each interactive tweet as a measurement that the truster has towards the trustee. By treating interactive tweets as measurements, we can calculate trustworthiness m by following Equation 4.1. Instead of just averaging all the tweets, we divide them into different time windows based on their posted date. We cluster tweets posted within the same month into the same window. And we treat these windows differently. In each window, we group tweets based on their posted dates. On each day, we treat them equally and calculate the mean impression m_d for that specific day by following Equation 4.1, as well as c_d.

$$m_d = \frac{\sum_{i=1}^{i=N} m_i}{N} \quad and \quad c_d = 1 - 2 * r_d \quad where \quad r_d = \sqrt{\frac{\sum_{i=1}^{i=N}(m_i - m)^2}{N * (N-1)} + \frac{scale^2}{12}} \tag{4.1}$$

After calculating m_d and c_d for each day, we use the weighted mean to combine the results for each month (time window) in Equation 3.10. Here we use $w_i = \frac{1}{r_{di}^2}$,

to assign higher weights to those who have higher confidence. Correspondingly, the error of the weighted mean is expressed in Equation 3.11.

Similar to [40], we assume that trustworthiness can fade with time. We use forgetting factor σ, where σ is less than 1, to capture this effect on the truster's confidence. As we are going to use tweets posted from 01/01/2015 through 08/31/2015 to analyze stock market, we use tweets which were posted in the year of 2014 (before the stock market analysis period) to construct the trust network. Therefore, we have 12 time windows in total. The confidence of December, which is the most recent month, is not discounted. The confidence of November is discounted by σ, and the confidence of October is discounted by σ^2, and so on. We show this effect of forgetting factor in Equation 3.12, where i is the number of the corresponding month (i.e. $i = 1$ for January).

Similar to Equations 3.10 and 3.11, we combine 12 time windows' results using the weighted mean where weights are confidence $c^{'}$. In this chapter, we select one month as the length of the time window and the forgetting factor $\sigma = 0.95$. Further refinement of these parameters will be part of our future work.

4.3.2 Trust Inference

In this chapter, we use the same 15 combinations of transitivity and aggregation formulas as in Section 3.5. These are all commonly used formulas in literature, and each formula has its meaning. For the meanings and sources of these formulas, please refer to [210].

As we indicated in [210], the user-to-user Twitter trust network typically is sparsely connected. We use the mentioned 15 combinations of formulas, to infer indirect trust relationships. We call the number of links from the truster to the trustee hops. For example, if A knows Z through B, in this case, we say that it has two hops. On the one hand, by increasing the number of hops, we can have more pairs of users being connected. On the other hand, the larger number of hops, the lower is the accuracy

of the inferred indirect trust [4]. For the trade-off between them, in this chapter, we only infer indirect trust by two hops.

4.3.3 Users' Power/Reputation

Based on the built user-to-user trust network on Twitter, we can calculate users' power or reputation. If a user is trusted by a large number of other users, she/he will have a high influence in the whole community. In other words, in order for a user to have a higher power or reputation, first of all, she/he needs to have a large number of friends or incoming trust links. For example, she/he is a celebrity and followed by a larger number of people on Twitter. Besides this, those incoming trust links have to be trustworthy. Remember that we represent trustworthiness in the range of $[0, 1]$, and 0.5 represents neutral sentiment. In other words, most of the positive incoming trust links' trustworthiness should be larger than or equal to 0.5. Also, we want these incoming trust links to be confident as well as trustworthy. To consider the number of incoming trust links, their trustworthiness and confidence together, in this work, we define a simple method to calculate power for users as in Equation 4.2.

$$P_u = \sum_{ui \in IN_u \ \& \ m_{ui} \geq 0.5} m_{ui} * c_{ui} \tag{4.2}$$

Here, IN_u is the set of users who have trust links toward user u. In other words, IN contains all the users that have trust assessments towards user u. Considering trustworthiness and confidence together, we use the product of them $(m * c)$. In such a way, even given the same trustworthiness m, higher confident incoming trust links contribute more to users' power than lower confident trust links. By setting a threshold of 0.5 for trustworthiness, we only count trustworthy incoming trust links. For negative or neutral links, we do not want them to contribute to the power.

Note that, in this work, we use a very simple definition for users' power since the main purpose of this work is to show that trust network power based method is able to improve stock market analysis, but not to construct a complicated model to

predict abnormal returns. Exploring more complicated reputation algorithms, such as PageRank [221] and Peertrust [69], will be part of our future work.

4.4 Twitter Sentiment Valence

4.4.1 Sentiment Analysis for Tweets

As many existing works [79, 178, 195, 204, 206, 207, 211, 214, 215] suggested, social media's emotional valence can be a helpful and important factor for stock market analysis. Given a tweet, its sentiment can be analyzed in two dimensions: valence or polarity and arousal [195, 222]. As did many existing works, in this work we only use tweets' sentiment valence. In other words, for each tweet, we only analyze whether it is positive or negative. In some works in the field of stock market analysis, positive is also called bullish and negative is called bearish [202]. In this work, we use the terms of positive and negative.

In literature, there are two types of sentiment analysis tools: word count analysis strategy based tools and machine learning strategy based tools. For simplicity, many works use word count analysis strategy [178,195]. In this category, based on the given positive and negative dictionaries, each word is mapped into the positive, neutral, or negative tag. To aggregate sentiment valence, there exists two methods: document-level method or tweet-level method. In the document-level works, the numbers of positive, neutral and negative words are counted together for all the interesting tweets in the document. While in the tweet-level works, each tweet is first tagged as positive or negative, based on the number of positive and negative words that it contains. Then, sentiment tagged tweets are accumulated for document-level sentiment valence. In machine learning strategy based tools, words are first used to construct features, i.e., TF-IDF [223], which can be later used along with other features by classifiers [215, 224].

In this work, we use an existing sentiment analysis tool – SentiStrength [185] which was also used by [224], to do tweet-level sentiment analysis for each tweet.

SentiStrength has shown its good performance for informal short text [225]; however, it is not specially designed for financial text analysis. Therefore, in addition to its default lexicon, we also add Loughran and McDonald's financial dictionary [213], which is widely used for financial text sentiment analysis, into SentiStrength.

Note that SentiStrength has different types of output results. In the previous trust modeling phase, we used SentiStrength's multi-scales output results, which can provide more grained information about users' attitudes. While in the field of analyzing Twitter sentiment and stock market, only binary output results (positive or bullish vs. negative or bearish) are used [178, 195, 202, 211, 215, 224]. Following this, we use SentiStrength's binary output results in the stage of stock market analysis.

4.4.2 Aggregation of Twitter Sentiment Valence

By using SentiStrength, we analyze sentiment for each tweet. To investigate the relation between the Twitter sentiment valence and stock returns, we need to accumulate tweets' sentiment valence on a daily basis. To aggregate daily sentiment valence, there are three widely used variables in literature [195, 218]. Following [211], in this work, we select to use the log of the ratio of the number of positive tweets to the number of negative tweets, which is shown in Equation 4.3. Here, P is the number of daily positive tweets, and N is the number of daily negative tweets.

$$TSV = \log(\frac{1+P}{1+N}) \tag{4.3}$$

Among existing works, all tweets are considered equally important regardless of their authors. Each tweet contributes to either the number of positive tweets or to the number of negative tweets in Equation 4.3. However, in reality, the source of information is also very important [209]. In the case of stock market analysis, we assume that users who have a higher power in the community should have more influence than users with lower power. Therefore, we adjust Equation 4.3 by incorporating users' power, which is calculated in Section 4.3.3, into calculating Twitter sentiment valence. Instead of considering all tweets equally important, we weight tweets by

their authors' power as in Equation 4.4. Here, PS is the set of positive tweets, and NS is the set of negative tweets. up and un are the authors who post positive and negative tweets correspondingly.

$$TSV = \log(\frac{1 + \sum_{p \in PS} Power(up \mid up \ \ posts \ \ p)}{1 + \sum_{n \in NS} Power(un \mid un \ \ posts \ \ n)}) \tag{4.4}$$

Note that, it is possible that a single user appears two or more times in Equation 4.4. For example, a user might post two positive tweets about a specific firm on the same day. Or she/he can even post one positive tweet and one negative tweet about the same firm on the same day. In such cases, this user appears multiple times in Equation 4.4.

To compare with existing works, we introduce two baseline methods to calculate Twitter sentiment valence. In the first one, authors' information is ignored such that all the tweets are considered equally important as in Equation 4.3. We use TSV_{equal} to denote this method. Actually, TSV_{equal} is widely used by many existing works, including [195], [211], [214], [215] and [218]. In the second method, instead of using users' power as weights in Equation 4.4, we use the number of followers that the users have as weights. This is a straightforward way to differentiate users' influence, as the number of followers that a user has is directly available on Twitter. We denote this method as $TSV_{followers}$, and show it in Equation 4.5. Correspondingly, we denote our trust network power based method (Equation 4.4) as TSV_{power}.

$$TSV = \log(\frac{1 + \sum_{p \in PS} Numberoffollowers(up \mid up \ \ posts \ \ p)}{1 + \sum_{n \in NS} Numberoffollowers(un \mid un \ \ posts \ \ n)}) \tag{4.5}$$

4.5 Results

4.5.1 Data Sets

To investigate the relationship between Twitter sentiment valence and stock returns, we collected two sets of data: financial data set and Twitter data set.

Financial Data

To have sufficient information for firms that we are going to investigate, we select eight firms that have the largest number of tweets in our Twitter data set. On average, each firm have more than 40 daily tweets. Also, these eight firms are selected from the S&P 500 index. They are Apple Inc (AAPL), Amazon.com Inc (AMZN), Alphabet Inc Class C (GOOG), Facebook Inc (FB), Netflix Inc (NFLX), Gilead Sciences Inc (GILD), General Electric Corp (GE), and Microsoft Corp (MSFT). For the period that we are interested, we download their daily stock market data from Yahoo!Finance, which include open price, highest price, lowest price, close price, adjusted the close price and trading volume. We analyze Twitter and stock market information from 01/01/2015 through 08/31/2015, which include 167 trading days in total.

Like many other works [178, 195, 215] did, we focus on stock returns, which is defined in Equation 4.6. Here, we use the adjusted close price in Equation 4.6. Therefore, stock returns reflect stock price's change compared with the previous trading day.

$$R_d = \frac{Price_d - Price_{d-1}}{Price_{d-1}} \tag{4.6}$$

In the field of finance, people are more interested in abnormal returns than stock returns [215]. Abnormal returns are defined as the actual stock returns minus the expected stock returns (also called normal returns) [215, 226], as shown in Equation 4.7. Here, we use $E[R_d]$ to denote the expected returns or normal returns. From this definition, we can see that abnormal returns somehow reflect external events or news' influence on the stock portfolios. In other words, abnormal returns are more sensitive to external events and news than stock market price itself.

$$AR_d = R_d - E[R_d] \tag{4.7}$$

In literature, there are many alternative methods and models used to calculate the expected stock returns [227]. To evaluate the difference and performance of these models is beyond the scope of our work. As in [215], we use the market model to

estimate the expected returns. It assumes that a firm's stock returns have a linear relation with the whole industry's stock returns. We use a linear regression model to represent it and show it in Equation 4.8. In our case, we use the S&P 500 index as the independent variable RSP. In Equation 4.8, α is the intercept, and β is the linear coefficient. As in [215] and [228], we use the previous 120 days as the training set to estimate α and β. α and β are estimated following the ordinary least squares (OLS) procedure. Therefore, although we only investigated from 01/01/2015 through 08/31/2015, we also collected part of 2014's stock data to calculate the expected returns.

$$E[R_d] = \alpha + \beta * RSP_d \tag{4.8}$$

Twitter Data

To collect stock market-related tweets, we find three official certificated accounts on Twitter. They are StockTwits, FinancialTimes, and MarketWatch. All of them are stock market-related companies or organizations. We consider them as three groups, and their followers discuss stock market within the groups. We also collect all the followers of these three groups and combine them into a single group or community. Note that, compared with Chapter 3, we have a larger data set in this chapter, which is composed of three stock market related groups.

We develop an application using Twitter 's open API as well as twitter4J library to collect data from Twitter. We first retrieve users' IDs and then use these user IDs to retrieve their tweets, which are written in English. Note that Twitter's open API limits data are collecting up to $3,200$ tweets from a single user's timeline. The dataset consists of users' screen names, locations, tweets, and the date and time when they posted the tweets. We take a snapshot of the group in September 2015. At that time, it had $2,898,756$ users in total. And from users' timelines, we collect all the tweets posted before September 2015, for a total number of $775,928,121$ tweets. In addition to their official accounts' followers, we also include users towards whom the followers

had posted interactive tweets. To build the trust network among users, we use all the collected interactive tweets. Based on our definition of trust, there are $20,916,112$ pairs of users having trust relationships. And based on the definition of users' power, $3,929,933$ users have their power calculated. So, we only consider tweets that were posted by these $3,929,933$ users in the later stock market analysis stage.

After building the trust network, we filter out tweets that are not related to the stock market in the stage of stock market analysis. Similar to many other works [178,195,206,215], we use the dollar sign (e.g. $AAPL), to select stock market related tweets, since the dollar sign is commonly used on Twitter to tag stock market related tweets. For the eight firms that we have selected above, we collect their daily tweets from 01/01/2015 through 08/31/2015. All the tweets are grouped on a daily basis for each firm. We list the number of tweets on trading days for each firm in Table 4.1.

Table 4.1.
Number of tweets on trading days from January 1st 2015 through August 31st 2015

Firm	Total number of tweets	Average number of daily tweets	Maximum number of daily tweets	Minimum number of daily tweets
AAPL	61,807	370.1018	2,653	101
FB	24,047	143.9940	1,089	37
GOOG	19,461	116.5329	704	29
NFLX	15,964	95.5928	665	13
AMZN	13,943	83.4910	912	14
GE	9,091	54.4371	491	10
MSFT	8,087	48.4251	567	9
GILD	7,329	43.8862	483	4

4.5.2 Trust Inference Validation Experiment

To infer indirect trust relations among users on Twitter, we collect three transitive formulas and five aggregation formulas from literature in Section 3.5, which results in 15 possible combinations. Each of them might fit well for specific applications. As [2] and [210] indicated, in different applications, users might exhibit different

trust propagation behavior patterns. Therefore, among 15 combinations, we need to select a combination that works for our Twitter application. To measure their trust inference accuracy, we use the leave-one-out cross-validation method [229]. Basically, in the leave-one-out cross-validation method, we compare the difference between the actual trust expressed by the truster and the inferred trust calculated by combinations of transitive formulas and aggregation formulas. For each leave-one-out case, we first hide the actual direct trust link (dash line in Figure 3.2) from the truster towards the trustee and use all the remaining direct trust links (solid line in Figure 3.2) to infer indirect trust by trust transitivity and trust aggregation formulas. An example of leave-one-out case is shown in Figure 3.2, where there exists N indirect paths from truster A to trustee Z through B_1, B_2, ... B_N. In our dataset, we have $499,327$ leave-one-out cases.

As we did before, for $AP2$, we use confidence as weights, $w = c$. Accuracy is measured by classical mean absolute error (MAE). We use $diffm$ to represent the absolute difference between the inferred m and actual m, and use $diffc$ to represent the absolute difference between the inferred c and actual c accordingly. Additionally, to consider $diffm$ and $diffc$ together, we also measure MAE for Manhattan distances, which is defined in Equation 3.13.

As [4] pointed out, inferred indirect trust becomes unreliable when the length of the chains (the number of hops) increases. Therefore, we only take into account the chains containing two hops. We list the performance results of 15 combinations in Table 4.2.

From Table 4.2, we can see that $TP1$ and $AP3$'s performance is significantly worse than other formulas, which is consistent with our previous work [210]. Although many applications use multiplication ($TP1$) as the transitivity formula [2, 4, 173], in this application, it is not the best one. To consider trustworthiness m and confidence c together, we select to use the combination of $TP3AP2$ in the remainder of this chapter, which has the smallest $MAE(man)$ among 15 combinations. By selecting $TP3$, it means that we considered the minimum m in a trust path as the bottleneck.

Table 4.2.
Comparison of formulas' performances

Formulas	MAE(diffm)	MAE(diffc)	MAE(Man)
TP1,AP1	0.2449	0.0793	0.3242
TP1,AP2	0.2452	0.0795	0.3247
TP1,AP3	0.2237	0.0786	0.3023
TP1,AP4	0.2133	0.0816	0.2949
TP1,AP5	0.2520	0.0811	0.3331
TP2,AP1	0.0728	0.0946	0.1674
TP2,AP2	0.0728	0.0947	0.1676
TP2,AP3	0.2294	0.0961	0.3255
TP2,AP4	0.0730	0.0907	0.1636
TP2,AP5	0.0733	0.0977	0.1710
TP3,AP1	0.0780	0.0851	0.1631
TP3,AP2	**0.0777**	**0.0850**	**0.1627**
TP3,AP3	0.2252	0.0893	0.3145
TP3,AP4	0.0788	0.0919	0.1707
TP3,AP5	0.0797	0.0916	0.1713

To aggregate trust paths, we use the weighted mean method $AP2$, where weights are trust paths' confidence. In other words, we assume that higher confident trust paths are more important than lower confident trust paths in aggregating trust paths.

4.5.3 Users' Power Distribution

As stated above, in addition to the direct trust links, we also infer indirect trust relations for users who originally are not directly connected by using $TP3AP2$ formulas. Given the trust network which includes both direct and indirect inferred trust relations among users, we calculate users' power/reputation by following our definition presented in Section 4.3.3. In Figure 4.1, we show the distribution of the number of users for 100 bins of power. We normalize users' power into the range of $[0, 1]$ using feature scaling. So each bin has a length of 0.01. Also, note that we use the log scale for the number of users in each bin. From Figure 4.1, we can see that as in many online communities, this distribution follows the power law distribution [230].

Only a few users have high influence in the community. These powerful users can be professional investors or gurus in the field of stock market.

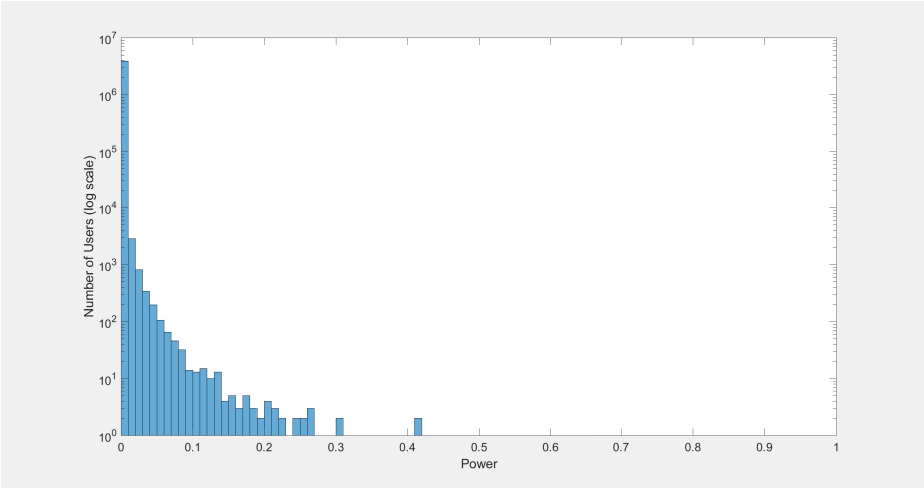


Figure 4.1. Distribution of the number of users with regard to users' power

4.5.4 Pearson Correlation

Pearson correlation [231] is widely used to measure the linear relationship between two variables, including time series variables. In this work, we use Pearson correlation coefficients (PCC) to measure the linear relation between the abnormal stock returns (AR) and Twitter sentiment valence (TSV). Remember that from 01/01/2015 through 08/31/2015, we have 167 trading days. Therefore, AR and TSV are two $167 * 1$ vectors. Given these two vectors, Pearson correlation coefficients can be calculated as shown in Equation 4.9, where E stands for the expectation value of the variable.

$$PCC = \frac{E[AR * TSV] - E[AR] * E[TSV]}{\sqrt{E[AR^2] - E[AR]^2} * \sqrt{E[TSV^2] - E[TSV]^2}} \tag{4.9}$$

In Table 4.3, we list the Pearson correlation coefficients between the selected eight firms' abnormal returns and their Twitter sentiment valence. In addition to Pearson correlation coefficients, we also test the p-values for them. We compare our trust network power based method TSV_{power} with other two baseline methods TSV_{equal}

(a widely used method by many existing works, such as [195], [211], [214], [215] and [218].) and $TSV_{followers}$ that we mentioned in Section 4.4.

Table 4.3.
Comparison of Pearson correlation coefficients for eight firms

Firms	TSV_{equal}		$TSV_{followers}$		TSV_{power}	
	PCC	p-value	PCC	p-value	PCC	p-value
AAPL	0.3370	$8.4 * 10^{-6}$	0.3969	$1.1 * 10^{-7}$	**0.4644**	$\mathbf{2.6 * 10^{-10}}$
FB	0.0662	0.395	0.0544	0.485	**0.0962**	**0.216**
GOOG	0.1830	0.018	0.1295	0.095	**0.2883**	$\mathbf{1.6 * 10^{-4}}$
NFLX	0.1416	0.068	0.1758	0.023	**0.4036**	$\mathbf{6.4 * 10^{-8}}$
AMZN	0.1314	0.091	0.3949	$1.3 * 10^{-7}$	**0.5318**	$\mathbf{1.4 * 10^{-13}}$
GE	0.0401	0.610	0.0043	0.956	**0.1530**	**0.048**
MSFT	0.0533	0.494	0.1035	0.183	**0.3812**	$\mathbf{3.7 * 10^{-7}}$
GILD	0.0969	0.213	0.0305	0.696	**0.1702**	**0.028**

In Table 4.3, among TSV_{equal}, $TSV_{followers}$ and TSV_{power} we use bold font to represent the most linearly correlated method with the stock abnormal returns. We can see that our method TSV_{power} performs better than other two methods for all eight firms. By weighting tweets' sentiment by their authors' power, TSV_{power} has higher PCC (and correspondingly lower p-value) than other two methods. For many firms, such as AMZN, GE, MSFT and GILD, by using $TSV_{followers}$ or TSV_{equal}, the Pearson coefficient between their Twitter sentiment valence and abnormal returns is weak (p-values are greater than 0.05), which means that Twitter sentiment valence might not have linear relation with abnormal stock returns. However, by using our trust network power based method, Twitter sentiment valence is significantly linearly related to abnormal stock returns for all the firms except FB. This confirms that the source of information (tweets) is an important factor to consider in this field of study. Compared with $TSV_{followers}$, TSV_{power} not only takes the number of trust links into account, but it also considers the quality of trust links. Our trust network power based method highlights powerful users' tweets and opinions, such that the

accumulated Twitter sentiment valence is more linearly related to the firms' abnormal returns.

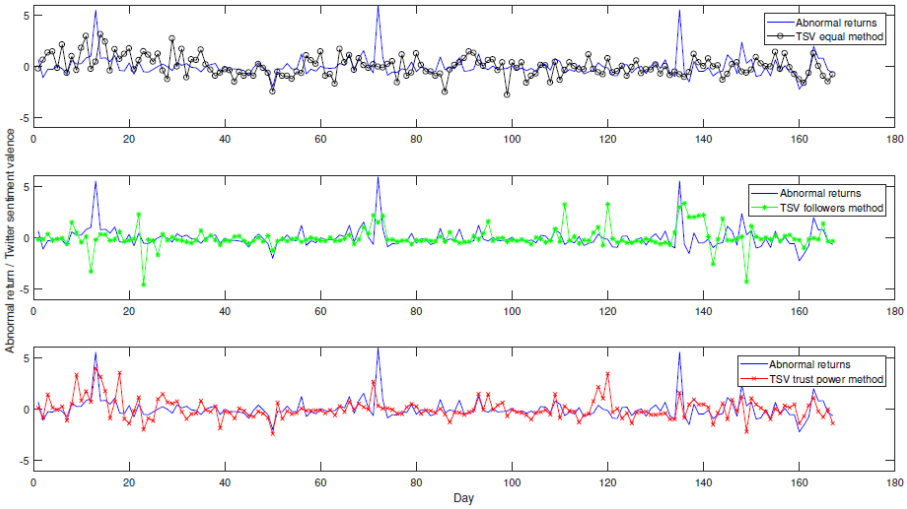


Figure 4.2. Comparison of three methods for NFLX

To illustrate this, we compare three methods' of performance for NFLX's abnormal returns and its Twitter sentiment valence in Figure 4.2. We can see that our trust network power based method reflects NFLX's abnormal returns much better than other two methods. For example, for the abnormal returns' peak at day 13, our method TSV_{power} follows the peak, while other two methods are not able to follow. Note that, in Figure 4.2, in order to compare abnormal returns and Twitter sentiment valence in the same scale, we convert both of them to Standard scores (also called z-scores) as shown in Equation 4.10, whose means are 0 and standard deviations are 1. In Equation 4.10, z is a Standard score, x is the original score, μ and σ are the mean and standard deviation of the population respectively.

$$z = \frac{x - \mu}{\sigma} \tag{4.10}$$

To see how good Twitter sentiment valence is linearly related to the firms' abnormal returns, we select AMZN as an example, which has the largest Pearson correlation coefficient among eight firms. We illustrate the relation between AMZN's abnormal returns and our trust network power based method in Figure 4.3. We can see that

our method captures abnormal returns' fluctuation very well, especially for three abnormal returns' big peaks in Figure 4.3. Such kind of linear correlation can be used for other advanced analysis, for example, event study [215, 232] and stock price prediction.

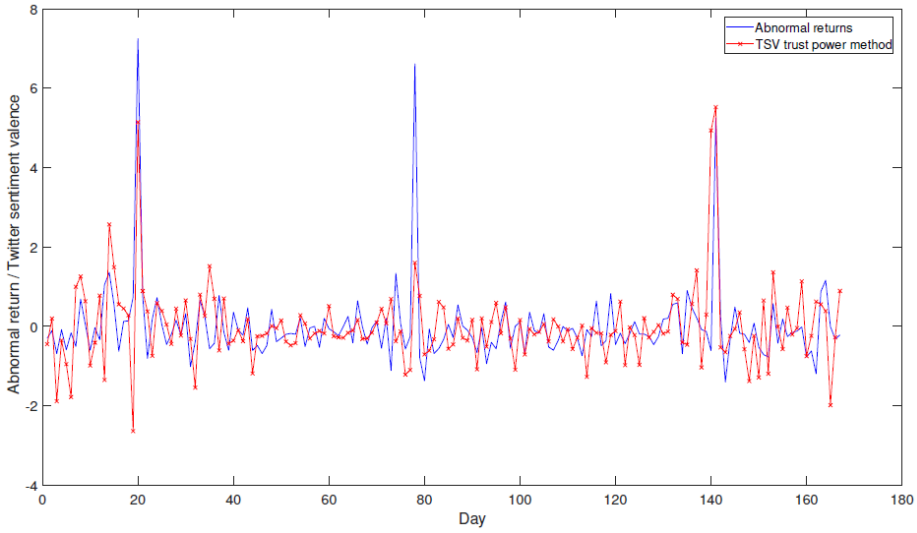


Figure 4.3. Pearson correlation between AMZN's abnormal returns and trust network power based Twitter sentiment valence

4.5.5 Linear Regression Correlation

In the above, Pearson correlation coefficient is used to measure the pairwise linear correlation between abnormal stock returns and Twitter sentiment valence. In addition to that, by taking into account that abnormal stock returns might exhibit auto-correlation property [209], we also construct a linear regression model which includes both Twitter sentiment valence and historical abnormal returns, as in Equation 4.11.

$$AR_d = \alpha + \beta * TSV_d + \gamma * CV + \varepsilon_d \tag{4.11}$$

Here, α is the intercept. β is the coefficient that we are going to investigate, and ε denotes a zero mean disturbance term. CV stands for control variables. Although there are many factors (i.e., trading volume, volatility) that can be considered as

control variables [193,209], in this work, we consider the previous three days' abnormal returns as control variables. Thus, we can rewrite the regression Equation 4.11 as in Equation 4.12.

$$AR_d = \alpha + \beta * TSV_d + \sum_{i=1}^{i=3} \gamma_i * AR_{d-i} + \varepsilon_d \tag{4.12}$$

We test Equation 4.12 with TSV calculated by three methods we mentioned above. We list estimated coefficient β, standard error of the estimation SE, t statistic for a test that the coefficient is zero $tStat$, p-value for the t statistic $pValue$, and adjusted R-square $adjR^2$ in Table 4.4.

Table 4.4.
Regression results of abnormal returns for eight firms

Firms	TSV methods	coefficient β	SE	$tStat$	$pValue$	$adjR^2$
AAPL	TSV_{equal}	0.0127	0.0026	4.9243	$2.07 * 10^{-6}$	0.1241
	$TSV_{followers}$	0.1103	0.0187	5.9106	$1.95 * 10^{-8}$	0.1717
	TSV_{power}	0.0365	0.0051	**7.1752**	$\mathbf{2.45 * 10^{-11}}$	**0.2359**
FB	TSV_{equal}	0.0016	0.0020	0.7919	0.4296	0.0372
	$TSV_{followers}$	0.0166	0.0246	0.6751	0.5006	0.0361
	TSV_{power}	0.0113	0.0086	**1.3162**	**0.1900**	**0.0437**
GOOG	TSV_{equal}	0.0066	0.0029	2.3166	0.0218	0.0115
	$TSV_{followers}$	0.0700	0.0407	1.7200	0.0873	-0.0030
	TSV_{power}	0.0398	0.0105	**3.8031**	$\mathbf{2.02 * 10^{-4}}$	**0.0624**
NFLX	TSV_{equal}	0.0077	0.0045	1.6902	0.0929	0.0094
	$TSV_{followers}$	0.2252	0.0991	2.2725	0.0244	0.0231
	TSV_{power}	0.1388	0.0244	**5.6836**	$\mathbf{5.99 * 10^{-8}}$	**0.1595**
AMZN	TSV_{equal}	0.0048	0.0028	1.6936	0.0923	0.0204
	$TSV_{followers}$	0.3276	0.0565	5.7987	$3.41 * 10^{-8}$	0.1744
	TSV_{power}	0.1228	0.0154	**7.9768**	$\mathbf{2.57 * 10^{-13}}$	**0.2842**
GE	TSV_{equal}	0.0007	0.0013	0.5824	0.5611	-0.0133
	$TSV_{followers}$	0.0014	0.0347	0.0400	0.9681	-0.0154
	TSV_{power}	0.0298	0.0150	**1.9911**	**0.0481**	**0.0089**
MSFT	TSV_{equal}	0.0010	0.0016	0.6095	0.5430	-0.0184
	$TSV_{followers}$	0.0588	0.0423	1.3908	0.1662	-0.0087
	TSV_{power}	0.0822	0.0155	**5.2984**	$\mathbf{3.77 * 10^{-7}}$	**0.1300**
GILD	TSV_{equal}	0.0023	0.0017	1.3519	0.1783	-0.0050
	$TSV_{followers}$	0.0603	0.1161	0.5189	0.6045	-0.0146
	TSV_{power}	0.0453	0.0189	**2.3970**	**0.0177**	**0.0185**

From Table 4.4, we observe the same performance pattern as in Pearson correlation coefficient test. In other words, after considering the stock abnormal returns' possible auto-correlation property, still, our trust network power based method outperforms other two methods. Similarly, in Table 4.4, we highlight the lowest p-value and the highest adjusted R-square among three methods with bold font. Therefore, the main hypothesis of this work that the users reputation built by using our trust management system, helps in making better predictions of the stock market is confirmed.

4.5.6 A Limitation – Number of Tweets

Although in the above experiments our trust network power based method outperformed other two baseline methods for all eight firms we selected, we find that to achieve this each firm must have enough number of daily tweets available. Remember that all eight firms we selected have more than 40 average daily tweets.

To see the influence of the number of daily tweets, we select another firm – Bank of America Corp (BAC), as an example. BAC is the 9th most mentioned firm in our Twitter dataset. And it has an average of 31.4192 daily tweets during our testing period. As before, we do the Pearson correlation test for BAC with three methods for all the 167 trading days. We list the results in Table 4.5. From Table 4.5 we can see that, in this case, TSV_{equal} performs better than our method TSV_{power}.

Table 4.5.
Pearson correlation coefficients of BAC

Testing period	TSV_{equal}		$TSV_{followers}$		TSV_{power}	
	PCC	p-value	PCC	p-value	PCC	p-value
All 167 trading days	**0.1877**	**0.015**	0.0244	0.755	0.1589	0.040
Subset40	0.1471	0.464	-0.1773	0.376	**0.4378**	**0.022**

Since BAC has only a few tweets on many trading days, instead of testing for all the 167 trading days, we also select a subset of trading days on which BAC has more than 40 tweets available and we call it Subset40. By setting a threshold of 40 for

the number of daily tweets, Subset40 has 27 trading days. Also, we test the Pearson correlation coefficient for Subset40 and include its results in Table 4.5. We can see that if we have enough number of tweets (in this example more than or equal to 40 daily tweets) to infer Twitter sentiment valence for BAC, still our method can outperform other two methods. Besides performance, we think that in order to get reliable analysis results, it is necessary to have sufficient tweets. Note that, compared with the difference between TSV_{equal} and TSV_{power} for all 167 trading days, our method performs much better in Subset40. Also, we compare three methods' performance of BAC in Subset40 in Figure 4.4. For example, our method TSV_{power} can capture day 10's drop, while other two methods are not able to capture it.

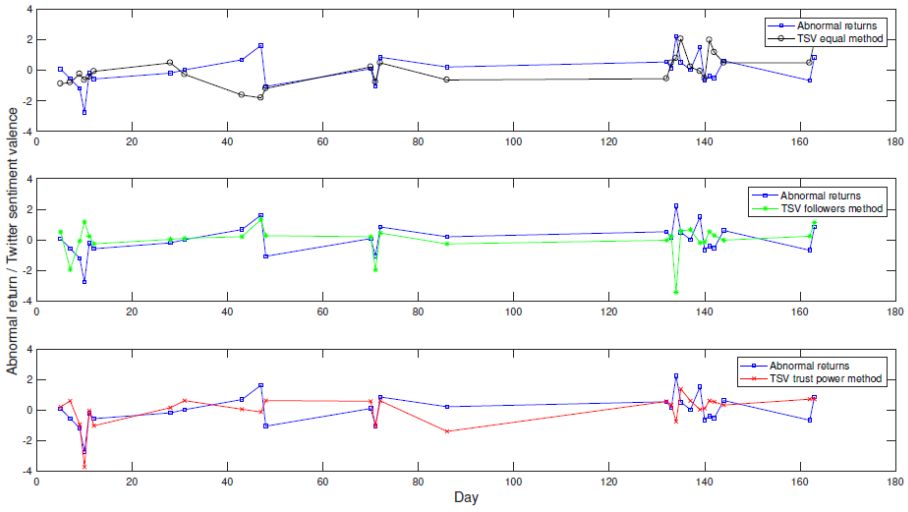


Figure 4.4. Comparison of BAC's performance in Subset40

4.6 Chapter Summary

In this work, we used the abnormal stock returns as ground truth for our trust management system. For this reason we verified the hypothesis that the users reputation, built by the inter trust among them, using our trust management system, helps in making better predictions of abnormal stock returns. So, we collected a group of users who were interested in stock market activities from Twitter. Based

on tweets posted by the users, we selected eight firms which were the top eight mentioned firms in the data set. Correspondingly, those eight firms' stock market data was collected from Yahoo! Finance. For the users on Twitter, we adapted our trust management framework [210] and constructed a user-to-user trust network. Based on this user-to-user trust network, we calculated for users' power or reputation in a simple way.

To see whether or not Twitter sentiment information could help to analyze stock market, for each firm, we analyzed Pearson correlation coefficients between Twitter sentiment valence and the firm's abnormal returns. Compared with existing works, when accumulating Twitter sentiments, we took into account tweets' authors. Authors were weighted and differentiated by their reputation or power in the whole community. Compared with treating all the authors equally or simply weighting authors by the number of their followers, we could see that our trust network based reputation mechanism could amplify the correlation between a specific firm's Twitter sentiment valence and the firm's stock abnormal returns.

To further consider the auto-correlation property of abnormal stock returns, we also constructed a linear regression model, in which the previous three days' abnormal returns were considered as control variables. Again, our results showed that by using our trust network power based method to weight tweets, we did linear regression better than other two methods.

However, our work also has some limitations. First of all, we did experiments only for a period (from 01/01/2015 through 08/31/2015). It is possible that the relation pattern we found here does not apply to other periods of time [209]. Therefore, testing our method on multiple data sets and periods of time is part of our future work. Furthermore, our study showed that when the number of tweets about a firm was very small; the Twitter sentiment valence might not be able to reflect the stock market. So, in the future, we will consider collecting more data or think of how to use available data more effectively. Finally, we will further tune the used reputation algorithms.

5 A TRUST MANAGEMENT FRAMEWORK FOR CLOUD COMPUTING PLATFORMS

5.1 Introduction

Nowadays, cloud computing platforms are becoming more and more widely used and welcomed in many fields, including e-commerce, web applications, data storage, healthcare, gaming, mobile social networks, and so on [233, 234]. Cloud computing platforms are able to provide customers with Internet-based services, without requiring customers to purchase a large number of hardware [235]. However, security and privacy are still two big concerns for cloud computing platforms and applications [236, 237]. For example, data confidentiality and auditability are two important properties for cloud vendors to convince customers to put their sensitive information in cloud [233]. Also, it is important for cloud vendors to provide available and reliable services, which is called business continuity and service availability in [233], to customers.

According to [233], cloud can be classified into public cloud and private cloud depending on their owners and serving objects. Public cloud is generally developed by big companies, e.g. Google and Amazon, and is designed to be accessible to public customers in a pay-as-you-go manner, such as Amazon EC2. While private cloud is usually owned by private companies or organizations. And only internal users have the access to use private cloud. In reality, as cloud can be owned by different owners, it is possible that a single mission or task will involve or be distributed over multiple clouds. In this work, we call this scenario multi-clouds environment as in [238].

In cloud computing platforms, on one hand, a single task might be distributed over multiple computer nodes. For example, one computer node pre-processes the data, the second computer node might do the data mining tasks, and the third one

visualizes the results to end users. On the other hand, a single computer node may be shared by multiple tasks. In such cases, it is possible that tasks are shared with some other untrustworthy tasks or organizations.

Faced with such new challenges, the old security model that consisted in defending the perimeter of the system is not valid anymore. We have to assume that whatever defense mechanisms we deploy in the systems, sooner or later will be breached by attackers. We have to design systems that can survive various attacks, with a calculated and acceptable degradation in performance by using additional resources planned for such conditions. Therefore, besides traditional security measures, such as cryptography, access control policies, and so on, more measures should be taken in cloud computing platforms. For example, when multiple cloud computing platforms are involved, not only load balance and redundancy should be taken into account, but also the trustworthiness of computing nodes, groups of nodes, tasks and cloud computing platforms should be taken into account.

In this chapter, we apply our measurement theory based trust management framework for cloud computing platforms, which addresses three levels of trust measurement: flow level trust, node level trust and task level trust. Both of the node level trust and the task level trust are dependent on the flow level trust. Although packets information is more detailed than flow information and may also be available in some cases, typically the amount of packets is much higher than the amount of flows such that it is very difficult to handle packets information [239]. Flows, which are the aggregation of packets, somehow also exhibit traffic features between the sender and the receiver. Therefore, in this work, we use flow level measurements rather than packet level measurements. To summarize, we estimate trustworthiness based on the network flow traffic.

We show that, by using trust metrics - trustworthiness and confidence, we are able to help cloud vendors or cloud customers to estimate both computing nodes' and tasks' trust. Based on the evaluation of trust, in cases that there are attacks, it could help cloud administrators to migrate tasks from suspect nodes to trustworthy

nodes. Also, it can help cloud administrators to dynamically allocate resources to tasks by the guidance of our trust management framework.

The rest of the chapter is organized as follows: we introduce literature works in Section 5.2. We illustrate the usage of our trust management framework in Section 5.3. We show the usage of our trust management framework by an attack example in Section 5.4. We propose a trust-reliability assessment algorithm and show its usage for resource configuration in Section 5.5. Finally, we conclude this chapter in Section 5.6.

5.2 Background and Related Works

As security is an very hot research topic in cloud, there are many works have been proposed to detect attacks and diminish their damage [237, 240].

There exist several works talking about trust between cloud vendors and cloud customers. For example, in [241], author explored the role of mutual trust between cloud providers and cloud customers in data storage systems. In [242], authors listed several factors which need to be considered in estimating trust, such as ownership, control, transparency, and so on. Therefore, we can see that there is a big need in cloud computing platforms for cloud vendors to be able to provide trust information to their customers.

On the other hand, there are also some works focusing on trust or risk assessment in distributed systems. In [243], authors defined risk using the concept of fuzzy belief to deal with risk's uncertainty property. In [244], authors established a network for hosts, connected with flows among them. And they explored both PageRank and HITS algorithms in their work. Similarly, in [245], authors assessed hosts' risk based on their flows and host network.

In this work, we adapt and apply our measurement theory based trust management framework to fulfill the gap between the need of trust and analysis of trust in cloud computing platforms. Basically, we provide an approach for cloud vendors

or administrators to assess trust of nodes and tasks in cloud environment. Also, it provides cloud vendors guidance for dynamically allocating resources. Compared with other existing works, in addition to the trustworthiness, we also had confidence included in our trust management framework. Confidence can be used to measure how certain the trustworthiness evaluation is. Furthermore, we develop a reconfiguration capability of tasks over elements of the system, such as tasks, computing nodes, networks, based on their trust values and the required trust by various tasks [246].

5.3 Trust Management in Cloud Computing Platforms

Trust has been shown to be very helpful in many decision making fields, such as IT systems, sociology [78], electronic commerce [142], Inter of Things [143], and so on. Therefore, there are many proposed trust management frameworks in literature [157].

In this work, we apply our measurement theory based trust management framework [210] in cloud computing platforms. Our trust management framework has two metrics: trustworthiness and confidence, as defined in Section 3.3.2.

5.3.1 Measurement of Flows

As indicated in Section 5.1, we measure trust based on network flows among computing nodes in cloud computing platforms. In our approach, we treat each network flow as an atomic measurement.

We assume that source and destination of flows are known such that we know the truster and trustee correspondingly. And flows between the truster and the trustee are treated like conversations or observations between them. In order to know the trust relationship, we need to analyze traffic flows. Basically, anomalous flows can decrease trustworthiness. To distinguish anomalous traffic from normal traffic, there exist many methodologies, such as machine learning-based method [247–249], rule-based method [250], and so on.

Traffic anomaly detection is beyond the scope of this paper. Instead, we use anomaly detection results as our trust management framework's input. For example, we can assume that we have profiles which specify both normal and abnormal traffic patterns for each flow, i.e., average and peak rates, banned destinations, and so on. For a given traffic, we compare it with the profile. Besides continuous anomaly scores, our trust management system can also handle binary cases. In some cases, the output of anomaly detection is a binary result (normal and abnormal) rather than a continuous value [247]. For example, classification algorithms and clustering algorithms will classify traffic into two categories. Depending on the input from anomaly detection, corresponding distributions can be applied. For example, we can use Beta distribution for binary inputs. For other discrete cases, we can use Dirichlet distribution.

In the following of this chapter, we assume the output of anomaly detection is continuous and is normalized into the range of $[0, 1]$. So, we use Normal distribution as an example to illustrate our trust management framework and its usage in cloud computing platforms. And we use the set of measurements $M = \{m_1, m_2, ...m_k\}$ to denote the anomaly detection results.

5.3.2 Trust Modeling: Trustworthiness and Confidence

We evaluate flow trust based on the flows' anomaly detection results. As defined in Section 3.3.2, we calculate m as the mean of $M = \{m_1, m_2, ...m_k\}$, and confidence is derived from M's error. As indicated by [72], confidence should have two important properties. First, given a fixed conflict ratio of evidence or measurements (i.e. positive vs. negative), confidence should increase as the amount of evidence or measurements increases. Second, given a fixed amount of evidence or measurements, confidence increases when the conflict ratio decreases.

We show these two properties in Figures 5.1 and 5.2. In this example, we only consider two possible anomaly scores $\{0, 1\}$. In other words, we consider that the

anomaly detection results are either positive or negative. We can see that when the total number of measurements is fixed, confidence achieves smallest when the ratio of positive and negative measurements is $1:1$. Also, in Figure 5.2, we fix the conflict ratio equaling to $1:1$, which means that we have the same number of positive and negative measurements. We can see that confidence monotonically increases with the number of total measurements. In other words, more measurements can make the trustworthiness estimation more confident.

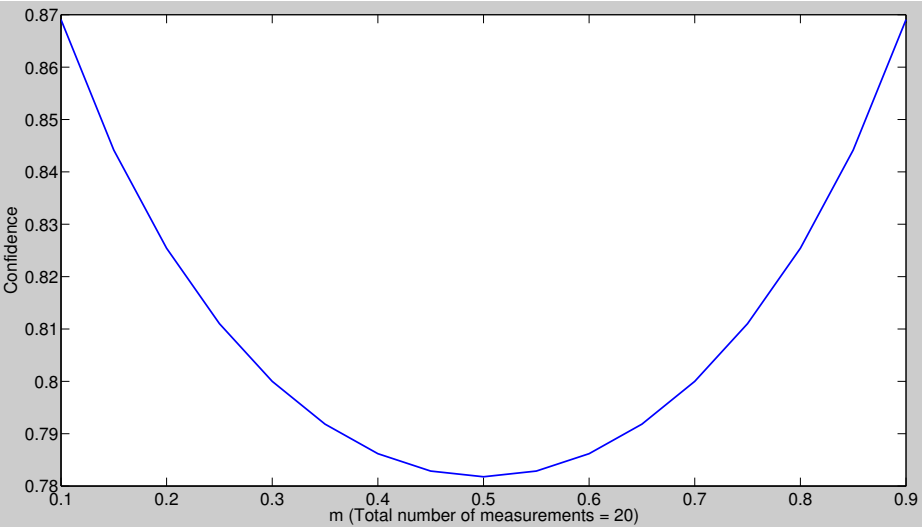


Figure 5.1. The conflict ratio's effect on confidence

As nodes or objects' behavior may change over time, trust assessment should be dynamically updated as well. Therefore, similar to the Twitter application, we divide flows based on time. For example, flows collected within one hour can be considered as a measurement window. The length of the time window will be tuned in future works.

Besides this, trust assessment should also highlight more on recent measurements than old measurements, as recent measurements are more likely to reflect the real time situation. Therefore, we forget the previous measurements by a forgetting factor σ, where $\sigma \leq 1$. Instead of treating previous measurements as important as current measurements, for each time window, we discount previous measurements by σ. The

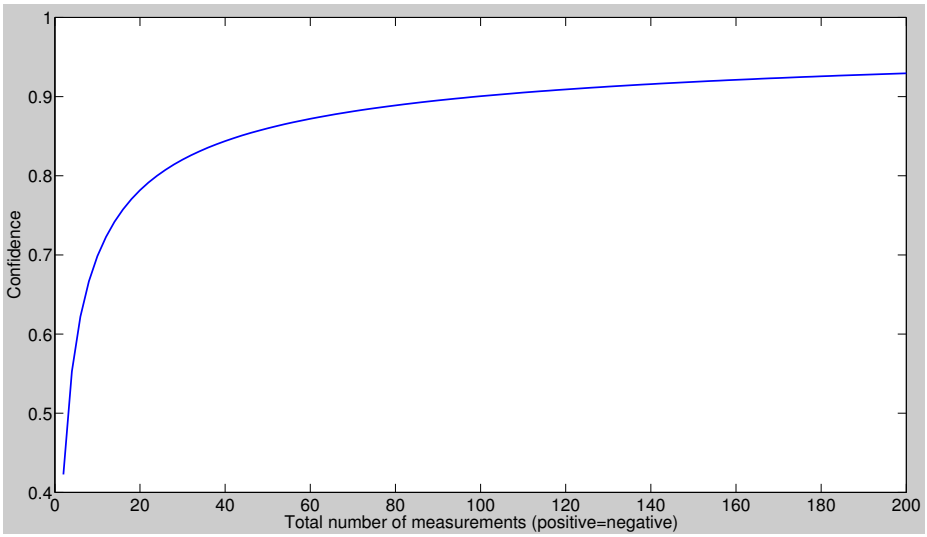


Figure 5.2. The total number of measurements' effect on confidence

larger is σ, the more important old measurements are. When $\sigma = 1$, we consider old measurements as equally important as current measurements. When $\sigma = 0$, we only use the most recent measurements to estimate trustworthiness. Note that in each new window, old measurements is discounted by σ. So the old measurements will be discounted by σ, σ^2, σ^3, and so on, as time goes on.

By discounting the old measurements, instead of using mean, we use weighted mean as trustworthiness m. Correspondingly, we use weighted sample variance to calculate confidence. We show them in Equation 5.1. Here, for the most recent measurements, weights are 1. And for previous measurements, weights are discounted, i.e. σ. Note that, for each m_i in a single time window, all the anomaly detection results are treated equally important and it follows the definition of Equations 3.1 and 3.2.

$$m = \frac{\sum_{i=1}^{i=k} w_i * m_i}{\sum_{i=1}^{i=k} w_i} \qquad r = \sqrt{\frac{\sum_{i=1}^{i=k} w_i * (m_i - m)^2}{\sum_{i=1}^{i=k} w_i}} \tag{5.1}$$

There exists a trade-off between the amount of available measurements and timely trust information. If we set σ too large, we will have more measurements available but lose the relative importance of the recent measurements. On the other hand, if we set σ too small, we can track trust estimation in real time, but with limited amount of

measurements. We show the effect of forgetting factor in Figure 5.3. In this example, we divide time into 20 time windows. And each time window has 5 new measurements. For the first 10 time windows, we assume they have 4 positive measurements and 1 negative measurement. Therefore, for the first 10 time windows, $m = 0.8$. For the next 10 time windows, we assume that the object changes its behavior, and each time window has 1 positive measurement and 4 negative measurements. We can see that, given smaller forgetting factors (forget more rapidly), m decreases more rapidly. On the other hand, smaller forgetting factors result in lower confidence.

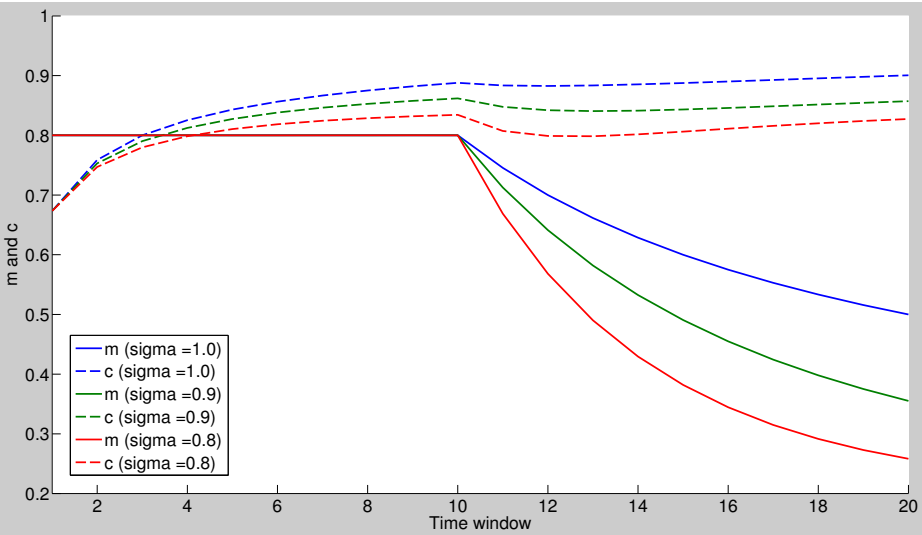


Figure 5.3. The effect of forgetting factor

5.3.3 Trust of Nodes

In [245], authors argued that the risk of a node/host is determined by both its incoming and outgoing links/flows. It is reasonable to assume that if a node sends/receives a large amount of anomalous flows, it may execute some malicious missions or it may be compromised. In addition to that, in cloud computing platforms, we consider that nodes' trust will also be affected by the tasks that are executing on the nodes. For example, if we know that a malicious task is running on a specific

node, although the node's incoming and outgoing flows have not exhibited anomaly yet, we will still treat this node as a suspect.

In summary, we take all the incoming and outgoing flows into account. Similarly, all the tasks running on the node will be considered. We represent measurements of all the incoming flows and outgoing flows as $flow_I$ and $flow_O$ correspondingly. And all the tasks running on a node are denoted as $Task = \{task_1, task_2, ...task_n\}$. For some types of attacks, incoming and outgoing flows are of different importance, we might consider using weighted mean of them. However, in the following simulated example, we consider that incoming and outgoing flows are equally important. And we use w_{flow} to denote the weight of incoming and outgoing flows.

By considering flows' trust and tasks' trust as two factors to determine trust for computing nodes, we represent it as in Equation 5.2. Here, w_{flow} and w_{task} control the relative weight of flows' trust to tasks' trust. By following the error propagation theory, confidence of the node can be calculated as in Equation 5.3.

$$m_{node} = \frac{w_{flow} * m_{flow} + \sum_{i=1}^{i=n} w_{task} * m_{taski}}{w_{flow} + \sum_{i=1}^{i=n} w_{task}} \tag{5.2}$$

$$\begin{aligned} c_{node} = 1 - 2 * [(\frac{w_{flow} * (1 - c_{flow})}{2 * (w_{flow} + \sum_{i=1}^{i=n} w_{task})})^2 \\ + \sum_{i=1}^{i=n} (\frac{w_{task} * (1 - c_{taski})}{2 * (w_{flow} + \sum_{i=1}^{i=n} w_{task})})^2]^{\frac{1}{2}} \end{aligned} \tag{5.3}$$

5.3.4 Trust of Tasks

Similar to trust of computing nodes, as tasks are involved with both flows and nodes (a set of nodes $Node = \{node_1, node_2, ...node_N\}$), we consider both of them in evaluating tasks' trust. However, compared with trust of nodes, where we consider all the incoming and outgoing flows, here we only take flows that belong to the corresponding tasks into account. In other words, a task's flow trust is only derived from its own flows (both incoming and outgoing flows). Similarly, we assume that incoming and outgoing flows are equally important in the following simulated example.

Similar to trust of nodes, we first calculate flow trust for each task. Also, we use the weighted mean of flow trust and nodes' trust to calculate trust for tasks, as shown in Equations 5.4 and 5.5.

$$m_{task} = \frac{w_{flow} * m_{flow} + \sum_{i=1}^{i=N} w_{node} * m_{nodei}}{w_{flow} + \sum_{i=1}^{i=N} w_{node}} \tag{5.4}$$

$$\begin{aligned} c_{task} = 1 - 2 * [(\frac{w_{flow} * (1 - c_{flow})}{2 * (w_{flow} + \sum_{i=1}^{i=N} w_{node})})^2 \\ + \sum_{i=1}^{i=N} (\frac{w_{node} * (1 - c_{nodei})}{2 * (w_{flow} + \sum_{i=1}^{i=N} w_{node})})^2]^{\frac{1}{2}} \end{aligned} \tag{5.5}$$

5.4 A Simulation Example

In this section, we show how to use our trust management framework in cloud computing platforms. We show an example of possible attack in cloud computing platforms.

5.4.1 An Attack Example in Cloud Computing Platforms

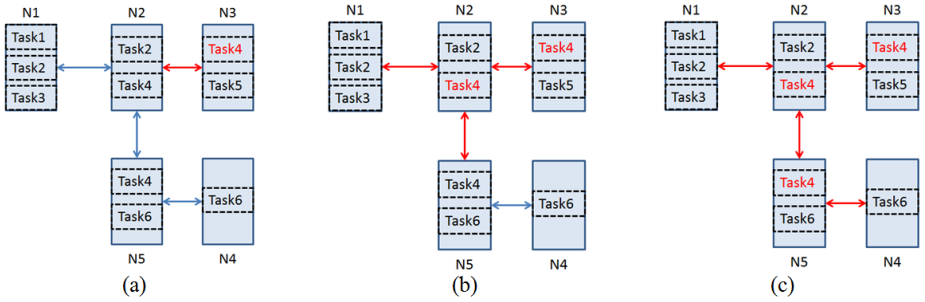


Figure 5.4. An attack example in cloud computing platforms

In Figure 5.4, we show an example of attack in cloud computing platforms. In this example, we have 6 tasks which are running on 5 computing nodes. Among 6 tasks, tasks $T2$, $T4$ and $T6$ distribute over multiple nodes and have incoming and/or outgoing flows among these nodes. For tasks $T1$, $T3$ and $T5$, we assume that they can be accomplished in a single node such that there is no flow for them. Also, we

assume that each node has profiles for all the tasks running on it, and then it is able to justify anomalous and normal flows.

We assume that task $T4$ is a malicious task. Figure 5.4 (a) to (c) show the process of the attack. In (a), it begins to launch attack on node $N3$. In the next step, node $N2$ is compromised and begins to send malicious flows to node $N5$, which is also running task $T4$. Also, as node $N2$ is compromised, flows between node $N2$ and node $N1$ will be anomalous as well. Finally in (c), node $N5$ is also compromised. In this example, we assume that nodes are compromised and not only task $T4$ is affected, but also other tasks running on the same nodes will be affected. In Figure 5.4, red lines represent anomalous flows, and blue lines represent normal flows. We will see how the malicious task (task $T4$) will affect other nodes and tasks.

In this example, we have 3 time windows ($TW1$, $TW2$, $TW3$), which correspond to the scenarios of Figure 5.4 (a), (b) and (c). In addition to $TW1$, $TW2$, $TW3$, we assume that there exists a prior time window $TW0$, which includes the prior knowledge. Initially, in $TW0$, we assume that all the nodes, tasks, and flows are normal. Therefore, we let $m = 1$ and $c = 1$ for all the nodes and tasks. Regarding flows, in $TW0$, we assume that there are 10 normal flows for each link in Figure 5.4. For example, there exist 10 normal flows between node $N1$ and node $N2$. Obviously, all the flow trust initially has $m = 1$ and $c = 1$ as well.

For links among each pair of nodes, we assume that it contains 10 flows in each time window. Therefore, node $N2$ has 30 incoming and outgoing flows in total in each time window, as it has three links with nodes $N1$, $N3$ and $N5$. As we indicated before, for simplicity, we consider incoming flows as important as outgoing flows. In other words, 10 is the total number of flows between a pair of nodes, no matter how many of them are incoming flows or outgoing flows. For time windows $TW1$, $TW2$, $TW3$, we assume that each link contains 10 normal flows (for each measurement $m_i = 1$) if the link is not affected (blue links). Otherwise, we assume that all 10 flows are anomalous (red links), which means that their measurement results are $m_i = 0$.

In Table 5.1, we list the trust information for all the nodes and tasks for 4 time windows. As we assume that initially all the nodes and tasks are not affected, we assign $m = 1$ and $c = 1$ for them. In this example, we let the forgetting factor $\sigma = 0.8$. To consider the importance of flow trust relative to tasks and nodes' trust, we let $w_{flow} = 2 * w_{task}$ and $w_{flow} = 2 * w_{node}$ correspondingly. Also, note that within each time window, we update nodes and tasks' trust using the previous time window's results as prior knowledge.

Table 5.1.
Trust information for all the nodes and tasks in 4 time windows

	$TW0(m,c)$	$TW1(m,c)$	$TW2(m,c)$	$TW3(m,c)$
T1	$(1,1)$	$(1,1)$	$(1,1)$	$(0.84,0.92)$
T2	$(1,1)$	$(1,1)$	$(0.76,0.90)$	$(0.57,0.91)$
T3	$(1,1)$	$(1,1)$	$(1,1)$	$(0.84,0.92)$
T4	$(1,1)$	$(0.86,0.93)$	$(0.65,0.93)$	$(0.50,0.95)$
T5	$(1,1)$	$(1,1)$	$(0.72,0.88)$	$(0.60,0.91)$
T6	$(1,1)$	$(1,1)$	$(1,1)$	$(0.79,0.91)$
N1	$(1,1)$	$(1,1)$	$(0.84,0.92)$	$(0.71,0.92)$
N2	$(1,1)$	$(0.87,0.93)$	$(0.65,0.93)$	$(0.47,0.94)$
N3	$(1,1)$	$(0.72,0.88)$	$(0.60,0.91)$	$(0.43,0.92)$
N4	$(1,1)$	$(1,1)$	$(1,1)$	$(0.77,0.88)$
N5	$(1,1)$	$(1,1)$	$(0.85,0.93)$	$(0.65,0.93)$

From Table 5.1, we can see that although initially only task $T4$ is malicious, it can affect other nodes and tasks as well. First of all, as task $T4$ is distributed over nodes $N2$, $N3$ and $N5$, their trustworthiness will decrease a lot, which means that nodes $N2$, $N3$ and $N5$ are compromised by malicious task $T4$. In addition to that, we can see that it also affects tasks $T2$, $T5$ and $T6$, as they are running on the affected nodes $N2$, $N3$, and $N5$. Finally, in $TW3$, we can see that the malicious effect spread to all

the nodes and tasks in this example. They all decrease their trust from the initial status ($m = 1, c = 1$).

From Table 5.1, we show that our trust management framework is able to derive trust information for nodes and tasks in cloud computing platforms. The derived trust information is very helpful for cloud administrators to make decisions. When trust decrease is detected in any nodes, cloud administrators might need to monitor or investigate efforts on those suspect nodes. After investigation, corresponding measures should be taken to diminish potential damage. For example, tasks $T2$, $T5$ and $T6$ can be migrated in advance if we find that nodes they are running on are decreasing their trust. Or, at least alarms should be arisen for further investigation. Alarms and administrator's decision making will be part of our future works.

5.5 Trust, Redundancy and Reliability

5.5.1 Trust-Reliability Assessment

In Section 5.3, we have introduced how to evaluate trustworthiness and confidence for tasks and nodes. It is important for both cloud vendors and customers to monitor trustworthiness (m) and confidence (c). Compared with existing works, in addition to the trustworthiness itself, we also measure how certain the trustworthiness evaluation is with confidence. Since trust is related with system reliability, to consider trustworthiness and confidence together we call it trust-reliability assessment. And we propose an algorithm to assess trust-reliability in Algorithm 1.

In Algorithm 1, we first shift trustworthiness by $(m-0.5)$, since 0.5 means neutral trustworthiness. To consider m and c together, we multiply shifted m with c. And then it is normalized into range of $[0, 1]$. Finally, we use an exponential function to assess trust-reliability, in which λ is related with m. Basically, if both m and c are high, we want the corresponding trust-reliability assessment result being high as well (controlled by $\lambda 1$). Otherwise, if m is low, we want that the trust-reliability assessment result decreases dramatically (controlled by $\lambda 2$). Therefore, typically,

Algorithm 1: Trust-Reliability Assessment Algorithm

Input: m; c; lamda1; lamda2; mthreshold;

Output: Trust-Reliability

```
if m ≥ mthreshold: ;
 then
    lamda = lamda1 ;
end
else
    lamda = lamda2 ;
end
Normalizedmc = (2 * (m - 0.5) * c + 1) / 2 ;
Trust-Reliability = exp(- lamda * (1 - Normalizedmc)) ;
return Trust-Reliability ;
```

we have $\lambda 1 \leq \lambda 2$. To better illustrate this, we plot Figures 5.5 and 5.6. Here, $mthreshold = 0.5$, $\lambda 1 = 4$, and $\lambda 2 = 8$.

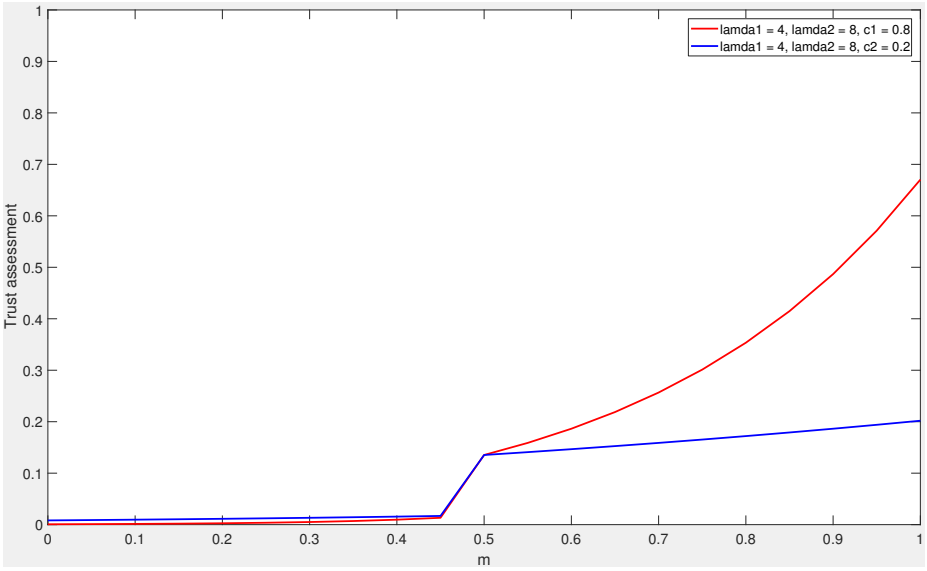


Figure 5.5. Trust-reliability assessment results vs. m

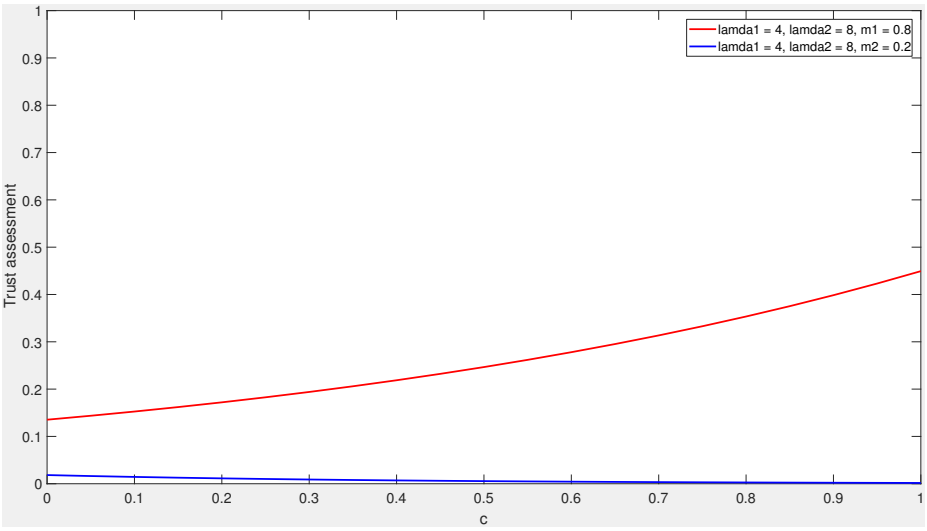


Figure 5.6. Trust-reliability assessment results vs. c

In Figure 5.5, we fix the confidence ($c = 0.2$ and $c = 0.8$ correspondingly). We can see that when m is low, the trust-reliability assessment result is always low. On the other hand, for high m (when m is greater than $mthreshold$), the trust-reliability assessment result increases dramatically when confidence increases. For low c, the trust-reliability assessment result does not increase too much even if we increase m. Similarly, we fix the trustworthiness ($m = 0.2$ and $m = 0.8$) in Figure 5.6. We can see that if trustworthiness m is low, the trust-reliability assessment result is always low no matter how confident it is. If trustworthiness m is high, then increasing confidence can help to increase the trust-reliability assessment result as well. In summary, to get a high trust-reliability assessment result, both m and c must be high.

We also show the trust-reliability assessment results for the attack example (Figure 5.4) in Table 5.2. Similarly, we can see that task 4 can potentially affect all the nodes and other tasks in this example.

5.5.2 Redundancy

Redundancy is a basic requirement in many networking frameworks to provide reliable services. On the one hand, it increases services' reliability by providing back-

Table 5.2.
Trust-reliability assessment results for all the nodes and tasks in 4 time windows

	$TW0$	$TW1$	$TW2$	$TW3$
T1	1	1	1	0.4729
T2	1	1	0.3451	0.1746
T3	1	1	1	0.4729
T4	1	0.5164	0.2365	0.1353
T5	1	1	0.2936	0.1948
T6	1	1	1	0.3889
N1	1	1	0.4729	0.2931
N2	1	0.5360	0.2365	0.0146
N3	1	0.2936	0.1948	0.0109
N4	1	1	1	0.3501
N5	1	1	0.4976	0.2365

ups for services. On the other hand, it requires more resources and in consequence, has a higher cost. Therefore, there is a trade-off between the degree of redundancy and cost.

In cases that tasks have a certain level of redundancy, it means that there are multiple methods or paths to implement them. For each method, we can use Equations 5.4 and 5.5 to evaluate m and c. Given more than one backup methods, we need to aggregate methods first. We use Figure 5.7 as an example to illustrate the aggregate methodology. In this example, Task 1 requires three nodes ($p1$ ($N1$, $N2$, $N3$) or $p2$ ($N4$, $N5$, $N6$)) to implement it. It also distributes two copies ($p1$ and $p2$) of the task over six nodes. For each single method $p1$ or $p2$, we have shown how to calculate trust for tasks in Equations 5.4 and 5.5. Given each method's trust, we will aggregate them together and evaluate trust metrics for the replicated task.

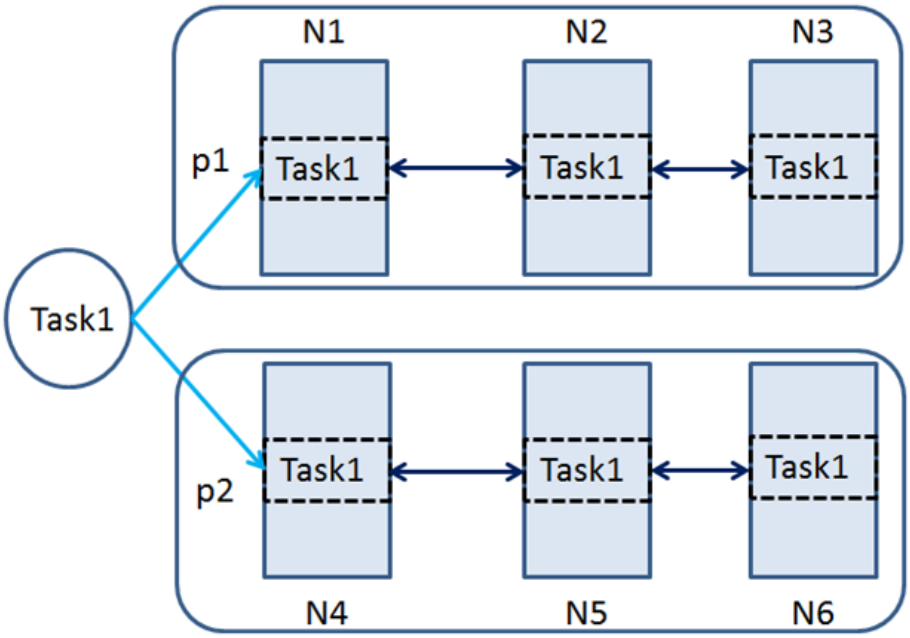


Figure 5.7. An example of redundancy

In the above example, there exist more than one implementation for the task (also called redundancy), we follow the redundancy theory (Equation 5.6) to aggregate them, as shown in Equations 5.7 and 5.8. Here, we assume that implementations are independent from each other. Basically, Equation 5.6 calculates the probability that at least one of two independent events happens. For example, if we have $(m_1, c_1) = (0.8, 0.8)$, and $(m_2, c_2) = (0.9, 0.9)$, then $(m, c) = (0.98, 0.9717)$. We can see that by adding more redundancy, we can increase tasks' m, c, and the trust-reliability assessment results. However, it requires more resources.

$$P(EA \cup EB) = P(EA) + P(EB) - P(EA \cap EB) \tag{5.6}$$

$$m = m_1 + m_2 - m_1 * m_2 \tag{5.7}$$

$$c = 1 - 2 * \sqrt{(\frac{(1 - m_2) * (1 - c_1)}{2})^2 + (\frac{(1 - m_1) * (1 - c_2)}{2})^2} \tag{5.8}$$

5.5.3 Resource Configuration

In the cloud computing scenario, given a set of devices or resources, the service providers or vendors need to assign right resources to applications. Suppose that the vendor has a set of candidate devices which can provide functional usage for an application; however, these resources might have different trust-reliability assessment results. At the same time, applications might also have different requirements.

By using Equations 5.4 and 5.5 and Algorithm 1, we are able to calculate and assess trust-reliability for each implementation. And if there exists redundancy, we use Equations 5.7, 5.8 and trust-reliability assessment algorithm together to evaluate trust for the application. Therefore, we can do trust-reliability assessment for each possible assembles of resources.

Note that, our trust-reliability assessment results are dynamic based on real-time monitored traffic. This information can also be used to dynamically configure the resources for the applications. In summary, by using our trust management framework, we can guide vendors for resource configuration.

5.6 Chapter Summary

In this chapter, we adapted and applied our measurement theory based trust management framework for cloud computing platforms. It consists of two metrics: trustworthiness and confidence. It begins from flow measurements. We derived trust of nodes based on all the tasks running on them and all the flows they send and/or receive. Similarly, for tasks, their trust depends on the flows and the nodes which implement the tasks. We provided a way for cloud vendors to estimate nodes and tasks' trust.

We used an example of attack to illustrate the usage of our trust management framework. We showed that although tasks themselves are not malicious initially, they can be affected and be compromised by other tasks if we are not aware of nodes' trust.

To diminish the damage, redundancy is an important feature for cloud computing platforms. We showed that by adding more copies or paths for tasks, it can increase their trust-reliability assessment results. Also, we provided a potential way for the administrators to dynamically allocate resource to tasks. For example, when the trust-reliability of a task decreases below a threshold, the administrator can allocate some additional paths for the task. While if the trust-reliability of a task is very high, the administrators might decrease the degree of its redundancy.

In summary, our trust management framework is able to provide guidance information for the administrators or even cloud customers to make decisions, e.g. migrating tasks from suspect nodes to trustworthy nodes, dynamically allocating resource, and managing the trade-off between the degree of redundancy and cost of resource.

6 CONCLUSIONS

In this work, we first provided a survey of existing trust management frameworks. We investigated how trust is defined by researchers from different disciplines and how can it be represented in the field of computer science. We also presented different trust inference schemes. Many of them have two important formulas: transitivity and aggregation formulas. Furthermore, we reviewed some potential trust attacks in trust management frameworks. We described four types of behaviors in these attacks. We analyzed existing frameworks vulnerabilities to the attacks. If they are robust to the attacks, we listed which defense mechanisms they use.

Then, we developed a measurement theory based trust management framework that aims to provide an intuitive way to represent and manage cognitive trust. For cognitive trust, we introduced two trust metrics: trustworthiness and confidence. We showed with experiments on two real online social communities data sets the validity of our framework, as well as its enormous potential usage in various social network applications. Our results showed that different applications or data sets have different trust inference patterns. Therefore, our framework is significantly important because it serves as an underlying fundamental for other schemes which focus on specific trust inference formulas.

We applied our trust management framework in two applications: stock market analysis and cloud computing scenarios. In the first application, we used the abnormal stock returns as ground truth for our trust management framework. Our experimental results showed that the users power/reputation, built by the inter trust among them, using our trust management system, can help in making better analysis of abnormal stock returns. Compared with treating all the authors equally or simply weighting authors by the number of their followers, we could see that our trust network based

power/reputation mechanism could amplify the correlation between a specific firms Twitter sentiment valence and the firms stock abnormal returns.

In the second application, we applied our trust management framework for cloud computing scenarios in which we aimed to help detecting and preventing tasks and computing nodes from being attacked. We provided a way for cloud vendors to estimate computing nodes and tasks trust. We used a simulated example of attack to illustrate the usage of our trust management framework. We showed that although tasks themselves are not malicious initially, they can be affected and be compromised by other tasks if we are not aware of nodes trust. To diminish the damage, redundancy is another important feature for cloud computing platforms. We showed that by adding more copies or paths for tasks, it can increase their trust. Also, we proposed a trust-reliability assessment algorithm which takes both trustworthiness and confidence into account. It could provide guidance information for the administrator or even customers to make decisions, e.g., migrating tasks from suspect nodes to trustworthy nodes, dynamically allocating a resource, and managing the trade-off between the degree of redundancy and cost of the resource.

REFERENCES

REFERENCES

[1] Marsh Stephen. *Formalising Trust as a Computational Concept*. PhD thesis, University of Stirling, Scotland, 1994.

[2] Patricia Victor, Chris Cornelis, and Martine De Cock. *Trust Networks for Recommender Systems*. Atlantis Publishing Corporation, 1st edition, 2011.

[3] Wenjun Jiang, Jie Wu, and Guojun Wang. On selecting recommenders for trust evaluation in online social networks. *ACM Transactions on Internet Technology*, 15(4):14:1–14:21, November 2015.

[4] Jennifer Ann Golbeck. *Computing and Applying Trust in Web-based Social Networks*. PhD thesis, University of Maryland at College Park, MD, USA, 2005.

[5] Matthew Richardson, Rakesh Agrawal, and Pedro Domingos. Trust management for the semantic web. In *International Semantic Web Conference*, pages 351–368. Springer, 2003.

[6] Audun Jøsang, Elizabeth Gray, and Michael Kinateder. Simplification and analysis of transitive trust networks. *Web Intelligence and Agent Systems*, 4(2):139–161, 2006.

[7] Christiano Castelfranchi and Rino Falcone. *Trust Theory: A Socio-cognitive and Computational Model*, volume 18. John Wiley & Sons, New York, 2010.

[8] Bernardo A Huberman, Daniel M Romero, and Fang Wu. Social networks that matter: Twitter under the microscope. *Available at SSRN 1313405*, 2008.

[9] Qusai Shambour and Jie Lu. A trust-semantic fusion-based recommendation approach for e-business applications. *Decision Support Systems*, 54(1):768–780, 2012.

[10] Hui Fang, Jie Zhang, Murat Şensoy, and Nadia Magnenat-Thalmann. Reputation mechanism for e-commerce in virtual reality environments. *Electronic Commerce Research and Applications*, 13(6):409–422, 2014.

[11] Audun Jøsang, Roslan Ismail, and Colin Boyd. A survey of trust and reputation systems for online service provision. *Decision Support Systems*, 43(2):618–644, 2007.

[12] Paolo Massa and Paolo Avesani. Controversial users demand local trust metrics: An experimental study on epinions.com community. In *Proceedings of the 20th National Conference on Artificial Intelligence*, volume 1, pages 121–126. AAAI Press, 2005.

[13] Sepandar D Kamvar, Mario T Schlosser, and Hector Garcia-Molina. The eigentrust algorithm for reputation management in p2p networks. In *Proceedings of the 12th International Conference on World Wide Web*, pages 640–651. ACM, 2003.

[14] Xiaoyong Li, Feng Zhou, and Xudong Yang. Scalable feedback aggregating (sfa) overlay for large-scale p2p trust management. *IEEE Transactions on Parallel and Distributed Systems*, 23(10):1944–1957, 2012.

[15] Mozhgan Tavakolifard. *On Some Challenges for Online Trust and Reputation Systems*. PhD thesis, Norwegian University of Science and Technology, Trondheim, Norway, 2012.

[16] Ping Zhang and Arjan Durresi. Trust management framework for social networks. In *Proceedings of the 2012 IEEE International Conference on Communications*, pages 1042–1047, 2012.

[17] Wanita Sherchan, Surya Nepal, and Cecile Paris. A survey of trust in social networks. *ACM Computing Surveys*, 45(4):47, 2013.

[18] Jennifer Golbeck. Trust on the world wide web: A survey. *Foundations and Trends in Web Science*, 1(2):131–197, 2006.

[19] Kang Chen, Guoxin Liu, Haiying Shen, and Fang Qi. Sociallink: Utilizing social network and transaction links for effective trust management in p2p file sharing systems. In *Proceedings of IEEE International Conference on Peer-to-Peer Computing*, pages 1–10, September 2015.

[20] Paolo Massa and Paolo Avesani. Trust-aware recommender systems. In *Proceedings of the 1st ACM Conference on Recommender Systems*, pages 17–24. ACM, 2007.

[21] Suvash Sedhain, Scott Sanner, Darius Braziunas, Lexing Xie, and Jordan Christensen. Social collaborative filtering for cold-start recommendations. In *Proceedings of the 8th ACM Conference on Recommender Systems*, pages 345–348. ACM, 2014.

[22] Guibing Guo, Jie Zhang, and Daniel Thalmann. Merging trust in collaborative filtering to alleviate data sparsity and cold start. *Knowledge-Based Systems*, 57:57–68, 2014.

[23] Audun Jøsang, Elizabeth Gray, and Michael Kinateder. Analysing topologies of transitive trust. In *Proceedings of the 1st International Workshop on Formal Aspects in Security & Trust*, pages 9–22. Pisa, Italy, 2003.

[24] Paolo Massa and Paolo Avesani. Trust metrics on controversial users: Balancing between tyranny of the majority. *International Journal on Semantic Web and Information Systems*, 3(1):39–64, 2007.

[25] Ugur Kuter and Jennifer Golbeck. Sunny: A new algorithm for trust inference in social networks using probabilistic confidence models. In *Proceedings of the 22nd National Conference on Artificial Intelligence*, volume 2, pages 1377–1382. AAAI Press, 2007.

[26] Yonghong Wang, Chung-Wei Hang, and Munindar P. Singh. A probabilistic approach for maintaining trust based on evidence. *Journal of Artificial Intelligence Research*, 40(1):221–267, January 2011.

[27] Yuan Yao, Hanghang Tong, Xifeng Yan, Feng Xu, and Jian Lu. Matri: A multi-aspect and transitive trust inference model. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 1467–1476, 2013.

[28] Yuan Yao, Hanghang Tong, Feng Xu, and Jian Lu. Subgraph extraction for trust inference in social networks. In *Encyclopedia of Social Network Analysis and Mining*, pages 2084–2098. Springer, 2014.

[29] Daire O'Doherty, Salim Jouili, and Peter Van Roy. Towards trust inference from bipartite social networks. In *Proceedings of the 2nd ACM SIGMOD Workshop on Databases and Social Networks*, pages 13–18, 2012.

[30] Hui Fang, Guibing Guo, and Jie Zhang. Multi-faceted trust and distrust prediction for recommender systems. *Decision Support Systems*, 71:37 – 47, 2015.

[31] Antesar M. Shabut, Keshav P. Dahal, Sanat Kumar Bista, and Irfan U. Awan. Recommendation based trust model with an effective defence scheme for manets. *IEEE Transactions on Mobile Computing*, 14(10):2101–2115, 2015.

[32] Jordi Sabater and Carles Sierra. Review on computational trust and reputation models. *Artificial Intelligence Review*, 24(1):33–60, 2005.

[33] Sonja Grabner-Kräuter and Ewald A Kaluscha. Empirical research in on-line trust: A review and critical assessment. *International Journal of Human-Computer Studies*, 58(6):783–812, 2003.

[34] Sebastian Ries, Jussi Kangasharju, and Max Mühlhäuser. A classification of trust systems. In *Proceedings of the International Conference On the Move to Meaningful Internet Systems*, pages 894–903, 2006.

[35] Sini Ruohomaa and Lea Kutvonen. Trust management survey. In *Proceedings of the 3rd International Conference on Trust Management*, pages 77–92, 2005.

[36] Stefan Spitz and York Tüchelmann. A survey of security issues in trust and reputation systems for e-commerce. In *Autonomic and Trusted Computing*, pages 203–214. Springer, 2011.

[37] Kevin Hoffman, David Zage, and Cristina Nita-Rotaru. A survey of attack and defense techniques for reputation systems. *ACM Computing Survey*, 42(1):1:1–1:31, December 2009.

[38] Bryan W Husted. The ethical limits of trust in business relations. *Business Ethics Quarterly*, pages 233–248, 1998.

[39] Jin-Hee Cho, Kevin Chan, and Sibel Adali. A survey on trust modeling. *ACM Computing Survey*, 48(2):28:1–28:40, October 2015.

[40] Yan Sun, Wei Yu, Zhu Han, and K.J.R. Liu. Trust modeling and evaluation in ad hoc networks. In *Proceedings of IEEE Global Telecommunications Conference*, volume 3, pages 1862–1867, 2005.

[41] Shenyun Che, Renjian Feng, Xuan Liang, and Xiao Wang. A lightweight trust management based on bayesian and entropy for wireless sensor networks. *Security and Communication Networks*, 8(2):168–175, 2015.

[42] Cai-Nicolas Ziegler and Jennifer Golbeck. Investigating interactions of trust and interest similarity. *Decision Support Systems*, 43(2):460–475, March 2007.

[43] Nianhua Yang. A similarity based trust and reputation management framework for vanets. *International Journal of Future Generation Communication and Networking*, 6(2):25–34, 2013.

[44] Carmen Fernandez-Gago, Isaac Agudo, and Javier Lopez. Building trust from context similarity measures. *Computer Standards & Interfaces*, 36(4):792 – 800, 2014.

[45] D Harrison McKnight and Norman L Chervany. The meanings of trust. Technical report, University of Minnesota, 1996.

[46] Alfarez Abdul-Rahman and Stephen Hailes. Supporting trust in virtual communities. In *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*, page 9 pp. vol.1, 2000.

[47] Bharat Bhargava, Leszek Lilien, Arnon Rosenthal, Marianne Winslett, M Sloman, TS Dillon, E Chang, FK Hussain, W Nejdl, D Olmedilla, et al. The pudding of trust. *IEEE Intelligent Systems*, 19(5):74–88, 2004.

[48] Surya Nepal, Wanita Sherchan, and Athman Bouguettaya. A behaviour-based trust model for service web. In *Proceedings of IEEE International Conference on Service-Oriented Computing and Applications*, pages 1–4. IEEE Computer Society, 2010.

[49] Yanchao Zhang and Yuguang Fang. A fine-grained reputation system for reliable service selection in peer-to-peer networks. *IEEE Transactions on Parallel and Distributed Systems*, 18(8):1134–1145, 2007.

[50] Audun Jøsang, Claudia Keser, and Theo Dimitrakos. Can we manage trust? In *Trust Management*, pages 93–107. Springer, 2005.

[51] Paolo Massa. A survey of trust use and modeling in real online systems. *Trust E-services: Technologies, Practices and Challenges*, pages 51–83, 2007.

[52] Jin-Hee Cho, A. Swami, and Ing-Ray Chen. A survey on trust management for mobile ad hoc networks. *IEEE Communications Surveys Tutorials*, 13(4):562–583, April 2011.

[53] Dongxia Wang, T. Muller, Yang Liu, and Jie Zhang. Towards robust and effective trust management for security: A survey. In *Proceedings of IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications*, pages 511–518, September 2014.

[54] Hua Ma and Zhigang Hu. Cloud service recommendation based on trust measurement using ternary interval numbers. In *Proceedings of International Conference on Smart Computing*, pages 21–24, November 2014.

[55] Hassan Shakeri and Abbas Ghaemi Bafghi. A layer model of a confidence-aware trust management system. *International Journal of Information Science and Intelligent System*, 3(1):73–90, January 2014.

[56] R. Guha, Ravi Kumar, Prabhakar Raghavan, and Andrew Tomkins. Propagation of trust and distrust. In *Proceedings of the 13th International Conference on World Wide Web*, pages 403–412, New York, NY, USA, 2004.

[57] Jennifer Golbeck and James Hendler. Inferring binary trust relationships in web-based social networks. *ACM Transactions on Internet Technology*, 6(4):497–529, 2006.

[58] Jure Leskovec, Daniel Huttenlocher, and Jon Kleinberg. Predicting positive and negative links in online social networks. In *Proceedings of the 19th International Conference on World Wide Web*, pages 641–650, New York, NY, USA, 2010.

[59] Richong Zhang and Yongyi Mao. Trust prediction via belief propagation. *ACM Transactions on Information Systems*, 32(3):15, 2014.

[60] Guangwei Zhang, Jianchu Kang, and Rui He. Towards a trust model with uncertainty for e-commerce systems. In *IEEE International Conference on e-Business Engineering*, pages 200 –207, October 2005.

[61] Paolo Massa and Paolo Avesani. Trust-aware collaborative filtering for recommender systems. *On the Move to Meaningful Internet Systems*, 2004.

[62] Yan Lindsay Sun, Wei Yu, Zhu Han, and K.J.R. Liu. Information theoretic framework of trust modeling and evaluation for ad hoc networks. *IEEE Journal on Selected Areas in Communications*, 24(2):305 – 317, February 2006.

[63] Mohammad Gias Uddin, Mohammad Zulkernine, and Sheikh Iqbal Ahamed. Cat: A context-aware trust model for open and dynamic systems. In *Proceedings of the ACM Symposium on Applied Computing*, pages 2024–2029, New York, NY, USA, 2008.

[64] Justin Zhan and Xing Fang. A novel trust computing system for social networks. In *Proceedings of IEEE 3rd International Conference on Privacy, Security, Risk and Trust and IEEE 3rd International Conference on Social Computing*, pages 1284 –1289, October 2011.

[65] Linke Guo, Chi Zhang, and Yuguang Fang. A trust-based privacy-preserving friend recommendation scheme for online social networks. *IEEE Transactions on Dependable and Secure Computing*, 12(4):413–427, July 2015.

[66] Stephen Marsh and Mark R Dibben. Trust, untrust, distrust and mistrust–an exploration of the dark (er) side. In *Trust Management*, pages 17–33. Springer, 2005.

[67] Hui Fang, Jie Zhang, and Nadia Magnenat Thalmann. A trust model stemmed from the diffusion theory for opinion evaluation. In *Proceedings of the International Conference on Autonomous Agents and Multi-agent Systems*, pages 805–812, Richland, SC, 2013.

[68] Surya Nepal, Wanita Sherchan, and Cecile Paris. Strust: A trust model for social networks. In *Proceedings of IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications*, pages 841–846, November 2011.

[69] Li Xiong and Ling Liu. Peertrust: Supporting reputation-based trust for peer-to-peer electronic communities. *IEEE Transactions on Knowledge and Data Engineering*, 16(7):843–857, 2004.

[70] Haibin Zhang, Yan Wang, Xiuzhen Zhang, and Ee-Peng Lim. Reputationpro: The efficient approaches to contextual transaction trust computation in e-commerce environments. *ACM Transactions on the Web*, 9(1):2:1–2:49, January 2015.

[71] Xiaoming Zheng, Yan Wang, Mehmet A Orgun, Guanfeng Liu, and Haibin Zhang. Social context-aware trust prediction in social networks. In *Service-Oriented Computing*, pages 527–534. Springer, 2014.

[72] Yonghong Wang and Munindar P. Singh. Formal trust model for multiagent systems. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pages 1551–1556, San Francisco, CA, USA, 2007.

[73] Guojun Wang and Jie Wu. Multi-dimensional evidence-based trust management with multi-trusted paths. *Future Generation Computer Systems*, 27(5):529–538, 2011.

[74] Guangchi Liu, Qing Yang, Honggang Wang, Xiaodong Lin, and Mike P Wittie. Assessment of multi-hop interpersonal trust in social networks by three-valued subjective logic. In *IEEE Conference on Computer Communications*, pages 1698–1706. IEEE, 2014.

[75] Audun Jøsang. A logic for uncertain probabilities. *International Journal of Uncertainty Fuzziness Knowledge-Based System*, 9(3):279–311, June 2001.

[76] Sebastian Ries. Certain trust: A trust model for users and agents. In *Proceedings of the ACM Symposium on Applied Computing*, pages 1599–1604, New York, NY, USA, 2007.

[77] Thomas DuBois, Jennifer Golbeck, and Aravind Srinivasan. Predicting trust and distrust in social networks. In *Proceedings of IEEE 3rd International Conference on Privacy, Security, Risk and Trust and IEEE 3rd International Conference on Social Computing*, pages 418–424, October 2011.

[78] Yefeng Ruan, Lina Alfantoukh, Anna Fang, and Arjan Durresi. Exploring trust propagation behaviors in online communities. In *Proceedings of the 17th International Conference on Network-Based Information Systems*, pages 361–367, September 2014.

[79] Yefeng Ruan, Lina Alfantoukh, and Arjan Durresi. Exploring stock market using twitter trust network. In *Proceedings of IEEE 29th International Conference on Advanced Information Networking and Applications*, pages 428–433, March 2015.

[80] Sibel Adali, Robert Escriva, Mark K Goldberg, Mykola Hayvanovych, Malik Magdon-Ismail, Boleslaw K Szymanski, William A Wallace, and Gregory Williams. Measuring behavioral trust in social networks. In *Proceedings of IEEE International Conference on Intelligence and Security Informatics*, pages 150–152. IEEE, 2010.

[81] Hyung Jun Ahn. A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem. *Information Sciences*, 178(1):37–51, January 2008.

[82] Touhid Bhuiyan. *Trust for Intelligent Recommendation*. Springer, 2013.

[83] Ing-Ray Chen, Jia Guo, and Fenye Bao. Trust management for soa-based IoT and its application to service composition. *IEEE Transactions on Services Computing*, 9(3):482–495, 2014.

[84] Jennifer Golbeck. Trust and nuanced profile similarity in online social networks. *ACM Transactions on the Web*, 3(4):12:1–12:33, September 2009.

[85] Guanfeng Liu, Yan Wang, Mehmet A Orgun, et al. Trust transitivity in complex social networks. In *Proceedings of the 25th Conference on Artificial Intelligence*, volume 11, pages 1222–1229, 2011.

[86] Diego Gambetta. Can we trust trust? In *Trust: Making and Breaking Cooperative Relations*, pages 213–237. Basil Blackwell, 1988.

[87] Audun Jøsang and Roslan Ismail. The beta reputation system. In *Proceedings of the 15th Bled Electronic Commerce Conference*, 2002.

[88] SJ Knapskog. A metric for trusted systems. In *Proceedings of the 21st National Security Conference*, pages 16–29. Citeseer, 1998.

[89] Mozhgan Tavakolifard, Kevin C. Almeroth, and Jon Atle Gulla. Does social contact matter?: Modelling the hidden web of trust underlying twitter. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 981–988, New York, NY, USA, 2013.

[90] David Goldberg, David Nichols, Brian M. Oki, and Douglas Terry. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12):61–70, December 1992.

[91] Steven Tadelis. The economics of reputation and feedback systems in e-commerce marketplaces. *IEEE Internet Computing*, 20(1):12–19, January 2016.

[92] Runfang Zhou and Kai Hwang. Powertrust: A robust and scalable reputation system for trusted peer-to-peer computing. *IEEE Transactions on Parallel and Distributed Systems*, 18(4):460–473, April 2007.

[93] Takashi Yajima, Asaki Matsumoto, and Hiroshi Shigeno. Ptrust: Provisional value based trust for reputation aggregation in peer-to-peer networks. In *1st International Symposium on Access Spaces*, pages 180 –185, June 2011.

[94] Weilian Xue, Yaqiong Liu, Keqiu Li, Zhongxian Chi, Geyong Min, and Wenyu Qu. Dhtrust: A robust and distributed reputation system for trusted peer-to-peer networks. *Concurrency and Computation: Practice and Experience*, 24(10):1037–1051, 2012.

[95] Xinxin Fan, Mingchu Li, Jianhua Ma, Yizhi Ren, Hui Zhao, and Zhiyuan Su. Behavior-based reputation management in p2p file-sharing networks. *Journal of Computer and System Sciences*, 78(6):1737–1750, 2012.

[96] Zeqian Shen and Neel Sundaresan. Reprank: Reputation in a peer-to-peer online system. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 163–164, New York, NY, USA, 2013.

[97] Zheng Yan, Yu Chen, and Yue Shen. Percontrep: A practical reputation system for pervasive content services. *The Journal of Supercomputing*, 70(3):1051–1074, 2014.

[98] Hasnae Rahimi and Hanan EL Bakkali. Ciosos: Combined idiomatic-ontology based sentiment orientation system for trust reputation in e-commerce. In *International Joint Conference*, pages 189–200. Springer, 2015.

[99] Kang Chen, Haiying Shen, K. Sapra, and Guoxin Liu. A social network based reputation system for cooperative p2p file sharing. *IEEE Transactions on Parallel and Distributed Systems*, 26(8):2140–2153, August 2015.

[100] Chunchun Wu, Tie Luo, Fan Wu, and Guihai Chen. An endorsement-based reputation system for trustworthy crowdsourcing. In *Proceedings of IEEE Conference on Computer Communications Workshops*, pages 89–90, April 2015.

[101] Irina Perfilieva and Jiří Močkoř. *Mathematical Principles of Fuzzy Logic*. Springer Science & Business Media, 1999.

[102] Rino Falcone, Giovanni Pezzulo, and Cristiano Castelfranchi. A fuzzy approach to a belief-based trust computation. In *Trust, Reputation, and Security: Theories and Practice*, pages 73–86. Springer, 2003.

[103] Vibhor Kant and Kamal K. Bharadwaj. Fuzzy computational models of trust and distrust for enhanced recommendations. *International Journal of Intelligent Systems*, 28(4):332–365, 2013.

[104] Kawser Wazed Nafi, Tonny Shekha Kar, Md. Amjad Hossain, and M.M.A. Hashem. A fuzzy logic based certain trust model for e-commerce. In *Proceedings of International Conference on Informatics, Electronics Vision*, pages 1–6, May 2013.

[105] Abdullah Aref and Thomas Tran. Using fuzzy logic and q-learning for trust modeling in multi-agent systems. In *Proceedings of Federated Conference on Computer Science and Information Systems*, pages 59–66, September 2014.

[106] Xin Liu, Anwitaman Datta, and Ee-Peng Lim. *Computational Trust Models and Machine Learning*. CRC Press, 2014.

[107] Fei Hao, Geyong Min, Man Lin, Changqing Luo, and L.T. Yang. Mobifuzzytrust: An efficient fuzzy trust inference mechanism in mobile social networks. *IEEE Transactions on Parallel and Distributed Systems*, 25(11):2944–2955, November 2014.

[108] Audun Jøsang, Guibing Guo, Maria Silvia Pini, Francesco Santini, and Yue Xu. Combining recommender and reputation systems to produce better online advice. In *Modeling Decisions for Artificial Intelligence*, pages 126–138. Springer, 2013.

[109] Sarvapali D. Ramchurn, Nicholas R. Jennings, and et al. A computational trust model for multi-agent interactions based on confidence and reputation. In *Proceedings of 6th International Workshop of Deception, Fraud and Trust in Agent Societies*, pages 69–75, 2003.

[110] Yonghong Wang and Munindar P. Singh. Trust representation and aggregation in a distributed agent system. In *Proceedings of the 21st National Conference on Artificial Intelligence*, pages 1425–1430. AAAI Press, 2006.

[111] Audun Jøsang, Tanja Ažderska, and Stephen Marsh. Trust transitivity and conditional belief reasoning. In *Trust Management VI*, pages 68–83. Springer, 2012.

[112] Chung-Wei Hang, Yonghong Wang, and Munindar P. Singh. Operators for propagating trust and their evaluation in social networks. In *Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems*, pages 1025–1032, 2009.

[113] Guanfeng Liu, Yan Wang, and Mehmet Orgun. Trust inference in complex trust-oriented social networks. In *International Conference on Computational Science and Engineering*, volume 4, pages 996–1001. IEEE, 2009.

[114] Erich Peter Klement, Radko Mesiar, and Endre Pap. *Triangular Norms*, volume 8. Springer Science & Business Media, 2013.

[115] Nele Verbiest, Chris Cornelis, Patricia Victor, and Enrique Herrera-Viedma. Trust and distrust aggregation enhanced with path length incorporation. *Fuzzy Sets and Systems*, 202:61 – 74, 2012.

[116] Huanyu Zhao and Xiaolin Li. Vectortrust: Trust vector aggregation scheme for trust management in peer-to-peer networks. *The Journal of Supercomputing*, 64(3):805–829, 2013.

[117] Bert Huang, Angelika Kimmig, Lise Getoor, and Jennifer Golbeck. A flexible framework for probabilistic models of social trust. In *Social Computing, Behavioral-Cultural Modeling and Prediction*, pages 265–273. Springer, 2013.

[118] Priyanka Agrawal, Vikas K Garg, and Ramasuri Narayanam. Link label prediction in signed social networks. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence*, pages 2591–2597. AAAI Press, 2013.

[119] Jihang Ye, Hong Cheng, Zhe Zhu, and Minghua Chen. Predicting positive and negative links in signed social networks by transfer learning. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 1477–1488, 2013.

[120] Jiliang Tang, Shiyu Chang, Charu Aggarwal, and Huan Liu. Negative link prediction in social media. *ArXiv Preprint ArXiv:1412.2723*, 2014.

[121] Young Ae Kim and Hee Seok Song. Strategies for predicting local trust based on trust propagation in social networks. *Knowledge-Based Systems*, 24(8):1360–1371, 2011.

[122] Jiliang Tang, Huiji Gao, Xia Hu, and Huan Liu. Exploiting homophily effect for trust prediction. In *Proceedings of the 6th ACM International Conference on Web Search and Data Mining*, pages 53–62. ACM, 2013.

[123] Rossouw Von Solms and Johan Van Niekerk. From information security to cyber security. *Computers & Security*, 38:97–102, 2013.

[124] N Paulauskas and E Garsva. Computer system attack classification. *Elektronika ir Elektrotechnika*, 66(2):84–87, 2015.

[125] David Basin, Cas Cremers, Kunihiko Miyazaki, Sasa Radomirovic, and Dai Watanabe. Improving the security of cryptographic protocol standards. *IEEE Security & Privacy*, 13(3):24–31, 2015.

[126] Reid Kerr and Robin Cohen. Smart cheaters do prosper: Defeating trust and reputation systems. In *Proceedings of the 8th International Conference on Autonomous Agents and Multi-agent Systems*, pages 993–1000, 2009.

[127] Omar Abdel Wahab, Jamal Bentahar, Hadi Otrok, and Azzam Mourad. A survey on trust and reputation models for web services: Single, composite, and communities. *Decision Support Systems*, 74:121–134, 2015.

[128] Yan Sun, Zhu Han, and K.J.R. Liu. Defense of trust management vulnerabilities in distributed networks. *IEEE Communications Magazine*, 46(2):112–119, 2008.

[129] Athirai Aravazhi Irissappane, Siwei Jiang, and Jie Zhang. Towards a comprehensive testbed to evaluate the robustness of reputation systems against unfair rating attack. In *Conference on User Modeling, Adaptation and Personalization Workshops*, volume 12, 2012.

[130] B Clifford Neuman and Theodore Ts'o. Kerberos: An authentication service for computer networks. *IEEE Communications Magazine*, 32(9):33–38, 1994.

[131] Iasonas Polakis, Panagiotis Ilia, Federico Maggi, Marco Lancini, Georgios Kontaxis, Stefano Zanero, Sotiris Ioannidis, and Angelos D Keromytis. Faces in the distorting mirror: Revisiting photo-based social authentication. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, pages 501–512. ACM, 2014.

[132] Lars Rasmusson and Sverker Jansson. Simulated social control for secure internet commerce. In *Proceedings of the 1996 Workshop on New Security Paradigms*, pages 18–25, New York, NY, USA, 1996. ACM.

[133] Audun Jøsang. Robustness of trust and reputation systems: Does it matter? In *Trust Management VI*, pages 253–262. Springer, 2012.

[134] John Douceur. The sybil attack. In *Peer-to-Peer Systems*, volume 2429 of *Lecture Notes in Computer Science*, pages 251–260. Springer, 2002.

[135] Hasari Tosun and John W. Sheppard. Incorporating evidence into trust propagation models using markov random fields. In *2011 IEEE International Conference on Pervasive Computing and Communications Workshops*, pages 263 –269, March 2011.

[136] Yan Sun and Yuhong Liu. Security of online reputation systems: The evolution of attacks and defenses. *IEEE Signal Processing Magazine*, 29(2):87 –97, March 2012.

[137] Yan Lindsay Sun, Zhu Han, Wei Yu, and KJ Ray Liu. A trust evaluation framework in distributed networks: Vulnerability analysis and defense against attacks. In *IEEE Conference on Computer Communications*, volume 2006, pages 1–13, 2006.

[138] Kevin Lai, Michal Feldman, Ion Stoica, and John Chuang. Incentives for cooperation in peer-to-peer networks. In *Workshop on Economics of Peer-to-Peer Systems*, pages 1243–1248, 2003.

[139] Paul Resnick et al. The social cost of cheap pseudonyms. *Journal of Economics & Management Strategy*, 10(2):173–199, 2001.

[140] Claudiu Duma, Nahid Shahmehri, and Germano Caronni. Dynamic trust metrics for peer-to-peer systems. In *Proceedings of the 16th International Workshop on Database and Expert Systems Applications*, pages 776–781, August 2005.

[141] Nitin Kumar Saini, Amit Chaturvedi, and Ramesh Chand Yadav. Identifying collusion attacks in p2p trust and reputation systems. *International Journal of Computer Applications*, 2:36–41, 2014.

[142] Paul Resnick, Ko Kuwabara, Richard Zeckhauser, and Eric Friedman. Reputation systems. *Communications of the ACM*, 43(12):45–48, December 2000.

[143] Yefeng Ruan, Arjan Durresi, and Lina Alfantoukh. Trust management framework for internet of things. In *Proceedings of IEEE 30th International Conference on Advanced Information Networking and Applications*, pages 1013–1019, March 2016.

[144] Ahmet Burak Can and Bharat Bhargava. Sort: A self-organizing trust model for peer-to-peer systems. *IEEE Transactions on Dependable and Secure Computing*, 10(1):14–27, January 2013.

[145] Jiliang Tang, Xia Hu, Yi Chang, and Huan Liu. Predictability of distrust with interaction data. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 181–190, 2014.

[146] Haifeng Yu, Michael Kaminsky, Phillip B. Gibbons, and Abraham Flaxman. Sybilguard: Defending against sybil attacks via social networks. In *Proceedings of the 2006 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, pages 267–278, 2006.

[147] Tad Hogg, Dennis M Wilkinson, Gabor Szabo, and Michael J Brzozowski. Multiple relationship types in online communities and social networks. In *Association for the Advancement of Artificial Intelligence Spring Symposium: Social Information Processing*, pages 30–35, 2008.

[148] Jin Huang, Feiping Nie, Heng Huang, and Yi-Cheng Tu. Trust prediction via aggregating heterogeneous social networks. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, pages 1774–1778, 2012.

[149] Tong Zhao, Chunping Li, Mengya Li, Qiang Ding, and Li Li. Social recommendation incorporating topic mining and social trust analysis. In *Proceedings of the 22nd ACM International Conference on Information and Knowledge Management*, pages 1643–1648, 2013.

[150] Michael Sirivianos, Kyungbaek Kim, and Xiaowei Yang. Socialfilter: Introducing social trust to collaborative spam mitigation. In *Proceedings of the IEEE Conference on Computer Communications*, pages 2300–2308, April 2011.

[151] Cai-Nicolas Ziegler and Georg Lausen. Propagation models for trust and distrust in social networks. *Information Systems Frontiers*, 7:337–358, 2005.

[152] Audun Jøsang. An algebra for assessing trust in certification chains. In *The Network and Distributed System Security Symposium*, volume 99, page 6th, 1999.

[153] Glenn Shafer et al. *A Mathematical Theory of Evidence*, volume 1. Princeton University Press, Princeton, 1976.

[154] Alistair G Sutcliffe, Di Wang, and Robin IM Dunbar. Modelling the role of trust in social relationships. *ACM Transactions on Internet Technology*, 15(4):16, 2015.

[155] Ifan Hughes and Thomas Hase. *Measurements and Their Uncertainties: A Practical Guide to Modern Error Analysis*. Oxford University Press, London, 2010.

[156] Kannan Govindan and Prasant Mohapatra. Trust computations and trust dynamics in mobile adhoc networks: A survey. *IEEE Communications Surveys & Tutorials*, 14(2):279–298, 2012.

[157] Yefeng Ruan and Arjan Durresi. A survey of trust management systems for online social communities - trust modeling, trust inference and attacks. *Knowledge-Based Systems*, 106:150–163, 2016.

[158] Chung-Wei Hang, Zhe Zhang, and Munindar P Singh. Shin: Generalized trust propagation with limited evidence. *IEEE Computer*, (3):78–85, 2013.

[159] George Vogiatzis, Ian MacGillivray, and Maria Chli. A probabilistic model for trust and reputation. In *Proceedings of the 9th International Conference on Autonomous Agents and Multi-agent Systems*, volume 1, pages 225–232, 2010.

[160] WT Luke Teacy, Jigar Patel, Nicholas R Jennings, and Michael Luck. Travos: Trust and reputation in the context of inaccurate information sources. *Autonomous Agents and Multi-Agent Systems*, 12(2):183–198, 2006.

[161] Christopher J Hazard and Munindar P Singh. Macau: A basis for evaluating reputation systems. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence*, 2013.

[162] Anthony A Clifford. *Multivariate Error Analysis*. John Wiley & Sons, 1973.

[163] Manfred Drosg. *Dealing with Uncertainties: A Guide to Error Analysis*. Springer, 2009.

[164] Millett Granger Morgan and Mitchell Small. *Uncertainty: A Guide to Dealing with Uncertainty in Quantitative Risk and Policy Analysis*. Cambridge University Press, Cambridge, 1992.

[165] Douglas C Montgomery and George C Runger. *Applied Statistics and Probability for Engineers*. John Wiley & Sons, New York, 2010.

[166] Grzegorz Wierzowiecki and Adam Wierzbicki. Efficient and correct trust propagation using closelook. In *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, volume 1, pages 676–681, 2010.

[167] Rensis Likert. A technique for the measurement of attitudes. *Archives of Psychology*, 1932.

[168] Anatoly E Fridman. *The Quality of Measurements: A Metrological Reference*. Springer Science & Business Media, 2011.

[169] Stanley Wasserman. *Social Network Analysis: Methods and Applications*, volume 8. Cambridge University Press, Cambridge, 1994.

[170] Paul W Holland and Samuel Leinhardt. Holland and leinhardt reply: Some evidence on the transitivity of positive interpersonal sentiment, 1972.

[171] Georgios E. Theodorakopoulos. *Robust Network Trust Establishment for Collaborative Applications and Protocols*. PhD thesis, University of Maryland at College Park, MD, USA, 2007.

[172] Herman JC Berendsen. *A Student's Guide to Data and Error Analysis*, volume 1. Cambridge University Press, Cambridge, 2011.

[173] Georgios E. Theodorakopoulos and John S. Baras. On trust models and trust evaluation metrics for ad hoc networks. *IEEE Journal on Selected Areas in Communications*, 24(2):318–328, February 2006.

[174] Ueli Maurer. Modelling a public-key infrastructure. In *Computer Security ESORICS 96*, volume 1146 of *Lecture Notes in Computer Science*, pages 325–350. Springer, Heidelberg, 1996.

[175] Seungjoon Lee, R. Sherwood, and B. Bhattacharjee. Cooperative peer groups in nice. In *INFOCOM 22nd Annual Joint Conference of the IEEE Computer and Communications*, volume 2, pages 1272–1282 vol.2, March 2003.

[176] Paolo Massa and Paolo Avesani. Trust metrics in recommender systems. In *Computing with Social Trust*, HumanComputer Interaction Series, pages 259–285. Springer, London, 2009.

[177] Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. Earthquake shakes twitter users: Real-time event detection by social sensors. In *Proceedings of the 19th International Conference on World Wide Web*, pages 851–860. ACM, 2010.

[178] Johan Bollen, Huina Mao, and Xiaojun Zeng. Twitter mood predicts the stock market. *Journal of Computational Science*, 2(1):1–8, 2011.

[179] Hao Wang, Dogan Can, Abe Kazemzadeh, François Bar, and Shrikanth Narayanan. A system for real-time twitter sentiment analysis of 2012 US presidential election cycle. In *Proceedings of the ACL 2012 System Demonstrations*, pages 115–120. Association for Computational Linguistics, 2012.

[180] Abdullah Almaatouq, Ahmad Alabdulkareem, Mariam Nouh, Erez Shmueli, Mansour Alsaleh, Vivek K. Singh, Abdulrahman Alarifi, Anas Alfaris, and Alex (Sandy) Pentland. Twitter: Who gets caught? observed trends in social micro-blogging spam. In *Proceedings of the ACM Conference on Web Science*, pages 33–41, 2014.

[181] Sibel Adali, Fred Sisenda, and Malik Magdon-Ismail. Actions speak as loud as words: Predicting relationships from social behavior data. In *Proceedings of the 21st International Conference on World Wide Web*, pages 689–698. ACM, 2012.

[182] Courtenay Honey and Susan C Herring. Beyond microblogging: Conversation and collaboration via twitter. In *Proceedings of the 42nd Hawaii International Conference on System Sciences*, pages 1–10. IEEE, 2009.

[183] Anubhav Kale, Amit Karandikar, Pranam Kolari, Akshay Java, Tim Finin, and Anupam Joshi. Modeling trust and influence in the blogosphere using link polarity. In *Proceedings of the International Conference on Weblogs and Social Media*, 2007.

[184] Bing Liu. Sentiment analysis and subjectivity. *Handbook of Natural Language Processing*, 2:627–666, 2010.

[185] Mike Thelwall, Kevan Buckley, Georgios Paltoglou, Di Cai, and Arvid Kappas. Sentiment strength detection in short informal text. *Journal of the American Society for Information Science and Technology*, 61(12):2544–2558, 2010.

[186] Yelp. Yelp competetion dataset, Accessed in 2014.

[187] Thomas F Coleman and Jorge J Moré. Estimation of sparse jacobian matrices and graph coloring blems. *SIAM Journal on Numerical Analysis*, 20(1):187–209, 1983.

[188] Andranik Tumasjan, Timm Oliver Sprenger, Philipp G Sandner, and Isabell M Welpe. Predicting elections with twitter: What 140 characters reveal about political sentiment. *Proceedings of the International Conference on Weblogs and Social Media*, 10(1):178–185, 2010.

[189] Bharath Sriram, Dave Fuhry, Engin Demir, Hakan Ferhatosmanoglu, and Murat Demirbas. Short text classification in Twitter to improve information filtering. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 841–842. ACM, 2010.

[190] Efthymios Kouloumpis, Theresa Wilson, and Johanna D Moore. Twitter sentiment analysis: The good the bad and the omg! In *Proceedings of the International Conference on Weblogs and Social Media*, volume 11, page 164, 2011.

[191] Balakrishnan Gokulakrishnan, Pavalanathan Priyanthan, Thiruchittampalam Ragavan, Nadarajah Prasath, and AShehan Perera. Opinion mining and sentiment analysis on a twitter data stream. In *International Conference on Advances in ICT for Emerging Regions*, pages 182–188. IEEE, 2012.

[192] Farhan Hassan Khan, Saba Bashir, and Usman Qamar. Tom: Twitter opinion mining framework using hybrid classification scheme. *Decision Support Systems*, 57:245–257, 2014.

[193] Paul C Tetlock. Giving content to investor sentiment: The role of media in the stock market. *The Journal of Finance*, 62(3):1139–1168, 2007.

[194] Lily Fang and Joel Peress. Media coverage and the cross-section of stock returns. *The Journal of Finance*, 64(5):2023–2052, 2009.

[195] Hong Keel Sul, Alan R. Dennis, and Lingyao Ivy Yuan. Trading on twitter: The financial information content of emotion in social media. In *Proceedings of the 47th Hawaii International Conference on System Sciences*, pages 806–815, January 2014.

[196] Chun-Hao Chen and Chih-Hung Yu. A series-based group stock portfolio optimization approach using the grouping genetic algorithm with symbolic aggregate approximations. *Knowledge-Based Systems*, 125:146–163, 2017.

[197] Yuchen Pan, Zhi Xiao, Xianning Wang, and Daoli Yang. A multiple support vector machine approach to stock index forecasting with mixed frequency sampling. *Knowledge-Based Systems*, 122:90–102, 2017.

[198] Alfred Cowles 3rd. Can stock market forecasters forecast? *Econometrica: Journal of the Econometric Society*, pages 309–324, 1933.

[199] Eugene F Fama. Random walks in stock market prices. *Financial Analysts Journal*, 51(1):75–80, 1995.

[200] Peter Klibanoff, Owen Lamont, and Thierry A Wizman. Investor reaction to salient news in closed-end country funds. *The Journal of Finance*, 53(2):673–699, 1998.

[201] Xueming Luo, Jie Zhang, and Wenjing Duan. Social media and firm equity value. *Information Systems Research*, 24(1):146–163, 2013.

[202] Hailiang Chen, Prabuddha De, Yu Hu, and Byoung-Hyoun Hwang. Wisdom of crowds: The value of stock opinions transmitted through social media. *The Review of Financial Studies*, 27(5):1367–1403, 2014.

[203] Qing Li, Yan Chen, Jun Wang, Yuanzhu Chen, and Hsinchun Chen. Web media and stock markets : A survey and future directions from a big data perspective. *IEEE Transactions on Knowledge and Data Engineering*, 30(2):381–399, 2017.

[204] John R Nofsinger. Social mood and financial economics. *The Journal of Behavioral Finance*, 6(3):144–160, 2005.

[205] Chong Oh and Olivia Sheng. Investigating predictive power of stock micro blog sentiment in forecasting future stock price directional movement. In *International Conference on Information Systems*, pages 1–19, 2011.

[206] Shuai Zhao, Yunhai Tong, Xinhai Liu, and Shaohua Tan. Correlating twitter with the stock market through non-gaussian svar. In *Proceedings of the 8th International Conference on Advanced Computational Intelligence*, pages 257–264, February 2016.

[207] Nuno Oliveira, Paulo Cortez, and Nelson Areal. The impact of microblogging data for stock market prediction: Using twitter to predict returns, volatility, trading volume and survey sentiment indices. *Expert Systems with Applications*, 73:125–144, 2017.

[208] Lian Fen Lee, Amy P Hutton, and Susan Shu. The role of social media in the capital market: Evidence from consumer product recalls. *Journal of Accounting Research*, 53(2):367–404, 2015.

[209] Hong Kee Sul, Alan R Dennis, and Lingyao Ivy Yuan. Trading on Twitter: Using social media sentiment to predict stock returns. *Decision Sciences*, 48(3):454–488, 2017.

[210] Yefeng Ruan, Ping Zhang, Lina Alfantoukh, and Arjan Durresi. Measurement theory-based trust management framework for online social communities. *ACM Transactions on Internet Technology*, 17(2):16:1–16:24, March 2017.

[211] Timm O Sprenger, Andranik Tumasjan, Philipp G Sandner, and Isabell M Welpe. Tweets and trades: The information content of stock microblogs. *European Financial Management*, 20(5):926–957, 2014.

[212] Dale W Jorgenson and Khuong Vu. Information technology and the world economy. *The Scandinavian Journal of Economics*, 107(4):631–650, 2005.

[213] Tim Loughran and Bill McDonald. When is a liability not a liability? textual analysis, dictionaries, and 10-ks. *The Journal of Finance*, 66(1):35–65, 2011.

[214] Jasmina Smailović, Miha Grčar, Nada Lavrač, and Martin Žnidaršič. Predictive sentiment analysis of tweets: A stock market application. In *Human-Computer Interaction and Knowledge Discovery in Complex, Unstructured, Big Data*, pages 77–88. Springer, 2013.

[215] Gabriele Ranco, Darko Aleksovski, Guido Caldarelli, Miha Grčar, and Igor Mozetič. The effects of twitter sentiment on stock price returns. *PloS One*, 10(9):e0138441, 2015.

[216] Eric Gilbert and Karrie Karahalios. Widespread worry and the stock market. In *Proceedings of the International Conference on Weblogs and Social Media*, pages 59–65, 2010.

[217] Xue Zhang, Hauke Fuehres, and Peter A Gloor. Predicting stock market indicators through Twitter "I hope it is not as bad as I fear". *Procedia-Social and Behavioral Sciences*, 26:55–62, 2011.

[218] Paul C Tetlock, Maytal Saar-Tsechansky, and Sofus Macskassy. More than words: Quantifying language to measure firms' fundamentals. *The Journal of Finance*, 63(3):1437–1467, 2008.

[219] Nicholas Evangelopoulos, Michael J Magro, and Anna Sidorova. The dual micro/macro informing role of social network sites: Can Twitter macro messages help predict stock prices? *Informing Science*, 15, 2012.

[220] Clive WJ Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica: Journal of the Econometric Society*, pages 424–438, 1969.

[221] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.

[222] Margaret M Bradley and Peter J Lang. Measuring emotion: The self-assessment manikin and the semantic differential. *Journal of Behavior Therapy and Experimental Psychiatry*, 25(1):49–59, 1994.

[223] Ian H Witten, Eibe Frank, Mark A Hall, and Christopher J Pal. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 2016.

[224] Ilya Zheludev, Robert Smith, and Tomaso Aste. When can social media lead financial markets? *Scientific Reports*, 4:4213, 2014.

[225] Mike Thelwall, Kevan Buckley, and Georgios Paltoglou. Sentiment strength detection for the social web. *Journal of the Association for Information Science and Technology*, 63(1):163–173, 2012.

[226] Stephen J Brown and Jerold B Warner. Using daily stock returns: The case of event studies. *Journal of Financial Economics*, 14(1):3–31, 1985.

[227] Antti Ilmanen. *Expected Returns: An Investor's Guide to Harvesting Market Rewards*. John Wiley & Sons, 2011.

[228] John Y Campbell, Andrew Wen-Chuan Lo, and Archie Craig MacKinlay. *The Econometrics of Financial Markets*. Princeton University Press, Princeton, 1997.

[229] Payam Refaeilzadeh, Lei Tang, and Huan Liu. Cross-validation. In *Encyclopedia of Database Systems*, pages 532–538. Springer, 2009.

[230] Lada A Adamic and Bernardo A Huberman. Power-law distribution of the world wide web. *Science*, 287(5461):2115–2115, 2000.

[231] Francis Galton. Regression towards mediocrity in hereditary stature. *The Journal of the Anthropological Institute of Great Britain and Ireland*, 15:246–263, 1886.

[232] Ekkehart Boehmer, Jim Masumeci, and Annette B Poulsen. Event-study methodology under conditions of event-induced variance. *Journal of Financial Economics*, 30(2):253–272, 1991.

[233] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, and Matei Zaharia. A view of cloud computing. *Communications of the ACM*, 53(4):50–58, April 2010.

[234] Hoang T Dinh, Chonho Lee, Dusit Niyato, and Ping Wang. A survey of mobile cloud computing: Architecture, applications, and approaches. *Wireless Communications and Mobile Computing*, 13(18):1587–1611, 2013.

[235] Ronald L. Krutz and Russell Dean Vines. *Cloud Security: A Comprehensive Guide to Secure Cloud Computing*. Wiley Publishing, 2010.

[236] Chunming Rong, Son T Nguyen, and Martin Gilje Jaatun. Beyond lightning: A survey on security challenges in cloud computing. *Computers & Electrical Engineering*, 39(1):47–54, 2013.

[237] Siani Pearson. Privacy, security and trust in cloud computing. In *Privacy and Security for Cloud Computing*, pages 3–42. Springer, 2013.

[238] Mohammed A. AlZain, Eric Pardede, Ben Soh, and James A. Thom. Cloud computing security: From single to multi-clouds. In *Proceedings of the 45th Hawaii International Conference on System Science*, pages 5490–5499, January 2012.

[239] Sui Song, Li Ling, and CN Manikopoulo. Flow-based statistical aggregation schemes for network anomaly detection. In *IEEE International Conference on Networking, Sensing and Control*, pages 786–791. IEEE, 2006.

[240] Azeem Sarwar and Muhammad Naeem Khan. A review of trust aspects in cloud computing security. *International Journal of Cloud Computing and Services Science*, 2(2):116, 2013.

[241] Ayad Barsoum and Anwar Hasan. Enabling dynamic data and indirect mutual trust for cloud computing storage systems. *IEEE Transactions on Parallel and Distributed Systems*, 24(12):2375–2385, December 2013.

[242] Khaled M. Khan and Qutaibah Malluhi. Establishing trust in cloud computing. *IT Professional*, 12(5):20–27, September 2010.

[243] Nan Feng and Minqiang Li. An information systems security risk assessment model under uncertain environment. *Applied Soft Computing*, 11(7):4332–4340, 2011.

[244] Shaonan Wang, Radu State, Mohamed Ourdane, and Thomas Engel. Riskrank: Security risk ranking for ip flow records. In *International Conference on Network and Service Management*, pages 56–63, October 2010.

[245] Mohsen Rezvani, Verica Sekulic, Aleksandar Ignjatovic, Elisa Bertino, and Sanjay Jha. Interdependent security risk analysis of hosts and flows. *IEEE Transactions on Information Forensics and Security*, 10(11):2325–2339, November 2015.

[246] Yefeng Ruan and Arjan Durresi. A trust management framework for cloud computing platforms. In *Proceedings of the 31st International Conference on Advanced Information Networking and Applications*, pages 1146–1153, March 2017.

[247] Chih-Fong Tsai, Yu-Feng Hsu, Chia-Ying Lin, and Wei-Yang Lin. Intrusion detection by machine learning: A review. *Expert Systems with Applications*, 36(10):11994–12000, 2009.

[248] Chirag Modi, Dhiren Patel, Bhavesh Borisaniya, Hiren Patel, Avi Patel, and Muttukrishnan Rajarajan. A survey of intrusion detection techniques in cloud. *Journal of Network and Computer Applications*, 36(1):42–57, 2013.

[249] Fairuz Amalina Narudin, Ali Feizollah, Nor Badrul Anuar, and Abdullah Gani. Evaluation of machine learning classifiers for mobile malware detection. *Soft Computing*, 20(1):343–357, 2016.

[250] Hongxin Hu, Gail-Joon Ahn, and Ketan Kulkarni. Detecting and resolving firewall policy anomalies. *IEEE Transactions on Dependable and Secure Computing*, 9(3):318–331, 2012.

VITA

VITA

Yefeng Ruan was born in Zhejiang, China. In 2009, he graduated from Wuhan University of Technology, Hubei, China, with his bachelor's degree in Electrical, Electronics and Communications Engineering. In the same year, he was admitted to the Master of Engineering program at Zhejiang University, Zhejiang, China. After he graduated from Zhejiang University, he was admitted to the Computer Science Ph.D. program at Purdue University in 2012.

Yefeng's area of expertise is in trust management framework, which included social network analysis, cyber security of cloud platforms, Internet of things/cyber physical systems, financial analysis, recommender systems, and so on. His research focused on security from trust's point of view.