

TRUST MANAGEMENT FRAMEWORK FOR FOG COMPUTING



By

Mahnoor Hamza

A thesis submitted to the faculty of Information Security Department,
Military College of Signals, National University of Sciences and Technology,
Islamabad, Pakistan, in partial fulfillment of the requirements for the degree of MS in
Military College of Signals, NUST

May 2021

Declaration

I hereby declare that no portion of work presented in this thesis has been submitted in support of another award or qualification either at this institution or elsewhere

DEDICATION

Dedicated to my Loving Parents

And

To Hands that ever raised in Prayer for me

Acknowledgement

First and foremost, I wish to express my gratitude to The Almighty for blessing me with His boundless support in this very intense academic year. I would love to express my warm appreciation to my parents for educating me the worth of persistence and determination. I wish to thank my husband Ibrahim for his unhindered support throughout this process and my sister QuratulAin for relentlessly providing me with her valuable advice time after time.

Last but not the least, I wish to extend my special thanks to my supervisor Asst. Prof Mian Muhammad Waseem Iqbal for his considerate guidance and in contributing much in helping me shape and reshape this piece of work.

ABSTRACT

Fog computing is the new way forward to prevent network traffic outburst the exhausted IoT devices. It has moved the computation away from the central cloud to various logical points along its path, what used to be a direct connection of an IoT device to the central cloud now passes through a series of fog nodes, processing data along the way before it reaches the cloud. Few of the key highlights of deploying Fog include efficient use of bandwidth, increased reliability and reduced latency. The ease of deployment and flexibility of fog has attracted many, but various security and privacy issues hinder its vast deployment by major platforms. One such open issue if regarding the trust establishment, trust is the level of confidence that an object will behave in an acceptable manner.

This thesis explores the requirements for establishing trust in fog environment, and through extensive literature review determines the similarities between SIoT and fog, as well as the trust requirement necessary for it. To mitigate the security and privacy issues of fog, we propose a two-way trust management scheme based on Bayes model, which allows both the service requestor and the service provider to validate the authenticity of each other before connecting. It is capable of effectively stopping a node from connecting with a malicious node. It is evaluated by simulating the system in Netlogo: an agent-based network simulator. The system shows resilience against trust-based network attacks and extensive evaluation shows that it has high accuracy and fastest convergence.

TABLE OF CONTENTS

DEDICATION	iv
ABSTRACT	vi
LIST OF FIGURES	x
LIST OF TABLES	xi
ACRONYMS	xii
1 INTRODUCTION	1
1.1 Overview	1
1.2 Motivation	2
1.3 Scope and Objectives	2
1.4 Contributions	3
1.5 Thesis Outline	3
2 PRELIMINARIES	5
2.1 What is Internet of Things (IoT)?	5
2.2 IoT as a Social Network (SIoT)	5
2.3 Current Trends in IoT and SIoT	6
2.4 An Overview of Fog Computing	7
2.4.1 Characteristics and Definition of Fog	7
2.4.2 Fog Reference Architecture	8
2.4.3 Applications of Fog	10
2.4.4 Research Direction	11
2.5 Motivation	12
3 LITERATURE REVIEW	13
3.1 Trust in the light of IT technology	13
3.2 Trust Management in SIoT	14
3.2.1 Trust Dimensions	16
3.3 Trust Requirements	16

3.4	A Survey of Trust models in IT environment	18
4	PROPOSED TRUST MANAGEMENT FRAMEWORK	24
4.1	Social Qualitative Trust Management Framework	24
4.2	Trust Composition using Bayes Model	27
4.2.1	Discounting Operation	28
4.3	Mathematical Description of Proposed Scheme	30
4.3.1	Trust Metrics	30
4.3.2	Mathematical Model	32
4.4	Algorithm of the Proposed Model	33
4.5	Attacks on Trust Management systems and the Resilience of SQT to these attacks	34
5	PERFORMANCE AND COMPARATIVE ANALYSIS OF PROPOSED SCHEME	36
5.1	Simulation Setup	36
5.2	Evaluation and Performance of SQT framework	36
5.3	Comparative Analysis	41
6	CONCLUSION AND FUTURE DIRECTION	43
	BIBLIOGRAPHY	43

LIST OF FIGURES

2.1	A multi-layer hierarchical Fog model with Cloud at the top and IoT devices at the bottom. The intelligence of the system decreases from top (Cloud) to bottom (IoT devices). Each intelligence group present at a higher level is more capable of performing complex tasks than the preceding layer.	9
3.1	An outlook of how devices in a network may work. SR determines the level of trust of SP by gathering recommendations from its neighbors.	15
4.1	Simplified single layer fog environment with each fog node connected to an IoT device. The fog nodes can communicate with their neighbors to share trust values.	25
4.2	Nodes A and B are the trustor and trustee nodes, to establish trust they need to validate each other by taking input from the recommender. The recommendations are weighed according to the relation of the nodes with the recommenders, this concept is popularly known as Discounting operation. . .	29
5.1	Initial setup of the simulation. The highlighted nodes are 1-hop neighbors, both blue and red highlighted nodes are calculating trust of one another as shown in Command Center	37
5.2	Trust value of a randomly selected good fog node in a network of 100 and 500	38
5.3	Trust value of a randomly selected bad fog node in a network of 100 and 500. The trust value remains below the threshold which makes it hard for a bad node to carry out attacks.	39
5.4	The performance of the proposed framework in on off selective forwarding attack.	40
5.5	Comparison of SQT system model with two existing SIoT models, Kowshalya and TMCoI-SIoT in the presence of on off selective forwarding attack.	42

LIST OF TABLES

2.1	Deployment models for fog computing	10
3.1	Categories of Trust computation in SIoT	16
3.2	Comparison and Analysis of Literature Review	20
3.3	Analysis of Trust Models in Fog	23
4.1	Trust based network attacks and resilience of our proposed solution to them	35
5.1	Comparison of SIoT trust model with the proposed SQT model	41

ACRONYMS

Internet of Things	IoT
Quality of Service	QoS
Service-oriented Internet of Things	SIoT
Mobile Ad-Hoc Networks	MANETs
Internet of Everything	IoE
Service Provider	SP
Service Requestor	SR
Self Promotion Attack	SPA
Bad Mouthing Attack	BMA
Ballot Stuffing Attack	BSA
Opportunistic Service Attack	OSA
On Off Attack	OOA

INTRODUCTION

1.1 Overview

The Internet of Things was introduced as early as 2008, since its inception it has grown tremendously, as a result of which, there are more objects connected to the internet than people. IoT is a network of devices, such as automobiles, homes, mobile phones and many more, that can sense and collect data from their surroundings and transmit it over some network. These devices produce data in abundance but lack the ability to process and store it [1], which implies that external means are necessary to compute and store this data to stabilize the resource constraint IoT devices [2]. Cloud computing introduced the concept of virtual handling of data, it offers virtually unlimited storage and processing resources which helps resolve the issues faced by resource poor devices. According to Cisco Global Cloud Index the amount of data collectively produced by people, things and machines will cross 847 zettabytes by 2021 [3].

Global trends are moving towards a more localized approach to enhance productivity over reduced cost, resulting in a shift from the centralized cloud to decentralized edge devices. The concept of offloading has gained much popularity since its inception, as it helps a device offload some of its tasks to a more computationally powerful device resulting in a smooth operation. Likewise, IoT devices follow the same footsteps and offload the computation intensive tasks to other systems in order to enhance their performance and bandwidth, as well as to mitigate the associated latency issues. The point in a network where the end-users connect to the core network is called the network edge. The concept of processing data on edge devices has alleviated the dependence on central nodes for regular tasks, rather it promotes processing of data near to its origin.

Fog Computing is a multi-layer architecture providing cloud-like services to the distributed network of devices at the edge of the network [4]. It came as an ally to the traditional cloud computing paradigm, enhancing its scope and supporting applications that

involve short response time, mobility support and confidentiality. Providing all this while bringing processing capabilities near the network edge improving latency and power consumption cost, Fog has attracted a lot of attention and made a huge impact on the market. Fog is a network of devices called fog nodes capable of performing tasks having some processing power, any device with adequate processing power can be a fog node and the end users use them for offloading tasks.

Trust in simplest terms, is defined as the confidence an object will behave in a manner defined by the QoS / security policies. In a digital environment, trust plays a huge role in helping out first time collaborators. Depending upon the trust levels of individual nodes as well as the network policies for minimum trust threshold a device can be deemed secure or insecure. A trust management system helps establish trust between nodes of a network for smooth functioning.

1.2 Motivation

Fog is an emerging architecture capable of filling various loopholes between cloud and IoT devices including storage limitations, computational cost, control and communication issues. The dynamic Fog network faces many security and privacy challenges that must be addressed for its development. This thesis will highlight one of the many solutions to this problem by proposing a trust management framework for Fog. Establishment of trust among two fog nodes is necessary for seamless collaboration, otherwise the nodes will hesitate to collaborate due to lack of confidence. This research will focus on trust among the fog nodes only .

Limited literature is available on establishing a trust management framework in Fog, whereas extensive studies have been carried out for establishing trust among IoT devices and cloud platforms. Various security loopholes present in Fog may give rise to network attacks, such as, man in the middle attack, ballot stuffing attack, on off attack etc. In order to protect user data and privacy there must be some sort of intrusion detection system present, such as a trust management framework that identify rogue nodes in a network.

1.3 Scope and Objectives

The following research outlines the need for an efficient trust management system for Fog. Through this research we recognized the need for both entities to trust one another before

connecting, so we propose a two way trust management system to enhance the reliability of the network. The scope of the research is to make a two way cost effective trust management framework in order to minimize the security and privacy issues in order to preserve the development of Fog.

1.4 Contributions

Our research contributions are as follows:

- A generic two-way trust management system is proposed which considers both qualitative and social trust metrics for trust calculation using the beta reputation function.
- Bayes trust estimates the current trust status of a node and predict its future behavior, it helps to evade “on off” attack.
- The accuracy and convergence of the proposed solution is tested by an agent based simulation software. The evaluation also incorporates the effects of varying good and bad nodes present in the network.
- We validate the proposed SQT management model through simulations and experimental results.

1.5 Thesis Outline

The structure of the thesis is as follows:

- Chapter 1: This chapter presents an overview of the thesis report, research objectives along with our contributions.
- Chapter 2: In this chapter, we explore IoT domain in depth and its familiarity with Fog. It also covers background study of Fog computing paradigm and a use case to understand the importance of trust in Fog.
- Chapter 3: Trust and various trust management systems are briefly explained along with trust requirements and possible network attacks.
- Chapter 4: The mathematical model of the research is presented along with the proposed technique.

- Chapter 5: This chapter presents the evaluation and validation of the proposed technique.
- Chapter 6: This chapter concludes the thesis and presents future direction.

PRELIMINARIES

This section will explore the domains of IoT, SIoT and Fog in detail while also highlighting the correlation between the aforementioned domains.

2.1 What is Internet of Things (IoT)?

To understand Fog, it is imperative to understand the concept of the ever-increasing IoT devices. Cisco defines IoT as an ecosystem of connected devices including people, places, objects and things. A device can be anything that has a sensor and is connected with a network. IoT cannot be confined to a single entity, it is a web of connected devices, sensing and collecting data by the minute. This data is useful to tons of different applications that are in place to make the system/ environment more automated and efficient by reducing cost, waste and loss. It empowers computers to make independent decisions based on the information gathered by themselves, it is equivalent to giving an electronic box the sense of sight, hearing and smell [5].

The purpose of IoT devices is simply to help generate lots and lots of data, to improve the accuracy of services offered by the applications associated with IoT. When these devices are connected over a large network such as the internet, they will sense and collect more information paving way towards improved and smarter services.

2.2 IoT as a Social Network (SIoT)

Internet of Things (IoT) and social networks colluded to give rise to Social Internet of Things (SIoT), a social network made up of things making decisions based on collected data. The purpose of SIoT is to promote social links in a network of devices collaborating to form social relations like humans do. The goal is help objects communicate with each other with minimal human interference. It will give rise to a more efficient and secure network of devices working harmoniously [6].

The social networks have proven to achieve greater results than any individual system, hence it will help IoT obtain even better results. The world is moving rapidly towards the

concept of Internet of Everything (IoE), which implies a network where everything will be a part of it, communicating and collaborating among themselves. In order to achieve harmonious functioning of a network in which every object is producing abundant data, it is imperative to give some autonomy to the system. It is humanly impossible to keep track of each device, and make decisions for it based on the gathered data. That is where SIoT steps in, it will make the social network a more independent entity, capable of making decisions autonomously. Selecting trustworthy objects to provide and receive services with least human intervention.

2.3 Current Trends in IoT and SIoT

IoT and SIoT technology has shaped the way our future may look, from a switch to whole cities, they have made everything in between smart. Be it smart gadgets, smart cars, smart homes or smart streets devices have now become far more capable of making decisions based on their experiences and collaborations. These devices are mostly sensors, sensing and collecting data from their surroundings which helps a system, be it smart home or smart city work in harmony. Currently, the data collected by millions of IoT devices is being processed and filtered in Cloud. The intricacy of this operation is expressed in the workflow described below:

$$collection \longrightarrow storage \longrightarrow analysis \longrightarrow action \longrightarrow inference \quad (2.1)$$

Cloud offers virtually unlimited storage capacity and performance capability which plays really well with devices such as IoT, that collect data in abundance but have limited resources of their own to process and store it. Technology as we know it is going through huge transformations, each sector is progressing independently so much so that some are ahead of others in the race. Such as storage and processing capacity has increased ten folds in the last decade but the network capacity to move big data without latency is still just a concept [7]. Hence, the question arises: How will the ever growing data produced by IoT will reach Cloud platform for storage and processing? The network bandwidth limits the applications of IoT and gives rise to many latency and quality of service (QoS) problems.

An easy solution to the above problem may be solved by using Fog as an intermediary layer which performs data filtering and processing near to where the data is being produced

i-e edge of the network. And sends to Cloud only that data which is absolute required. It will also prevent the system from a single point of failure, due to the distributed nature of Fog.

2.4 An Overview of Fog Computing

This section will describe the concept, working and architecture of Fog computing in great depth contrary to an overview given in the previous chapter. Section 2.4.1 describes some of its key characteristics and a definition, section 2.4.2 discusses fog reference architecture in great depth. Section 2.4.3, 2.4.4, 2.5 describe some applications of fog in modern day architecture, followed by the research direction taken up to complete this research and the motivation behind it.

2.4.1 Characteristics and Definition of Fog

Fog is an emerging multi-layer architecture stretching from the centralized cloud to the logical extreme of a network enabling processing and analytics to be carried out closest to the source of the generated data. It fills the storage, computation, control and communication loopholes between the centralized cloud and the IoT devices present at the edge of the network.

Cisco defines Fog as a *horizontal, system level architecture* bringing storage ,networking and control closer to the users, it further elaborates by defining a fog node as a *mini-cloud* closer to the edge of the network [8].

We live in the era of IoT where there will be more than one trillion devices up and running producing even greater amounts of data by the year 2025 [9]. Which means even more data will be there to send on the network than before. Fog computing presents an efficient solution to this problem by mitigating the dependency of IoT on the central node (cloud) and providing a distributed platform to deal with the constraints regarding heterogeneity and mobility of these resource constraint devices [10]. Furthermore, it minimizes the risk associated with data leakage due to long distance transmission and outsourcing features cloud provides, by processing and filtering the data near the edge and only sending whats needed onto the cloud [11].

In the traditional cloud architecture, the end user is always restricted to be a data consumer, the only possible actions it can perform include web browsing, scanning images, watching

videos etc. However, the role of the end user has evolved to generate/produce data, now they can also shift their roles between a consumer and a producer. This transition in roles can only be managed by the help of fog computing [10, 11]. As suggested by Bonomi et al. in [12] fog is the appropriate platform to secure the future of upcoming IoT services and applications but it has yet many challenges to overcome before its worldwide adaptation.

Fog computing faces many security and privacy issues due to its flexibility and distributed nature [13]. It comprises of different fog nodes that facilitates the fog clients by offloading and providing services at a close proximity. The wide acceptance of fog skyrocketed after OpenFog introduced a generalized system architecture and guidelines for its accurate and easy modelling [14]. It is imperative to consider fog the essential extension of cloud that might just save the network from overflowing traffic flowing from numerous IoT applications to cloud servers. It showcases minimal cost of deployment as it can make use of the nodes already in the path of the traffic. All these and many more features make fog a platform to look forward to.

2.4.2 Fog Reference Architecture

OpenFog proposed a generalized structure of fog in 2017 [14], which laid the grounds for all future research to be carried out. As briefly discussed in Section 2.4.1, fog is a multi-layer technology having hierarchical structure, with each layer capable of performing computational tasks according to its intelligence group. The placement of fog between cloud and IoT is such that, cloud holds the top place in a pyramid structure and IoT fills the bottom most level. The layers in between present different levels of fog devices categorized according to their intelligence level and computational capabilities to handle more complex tasks. The general outlook of a Fog infrastructure is shown in Figure 2.1

It is evident that higher hierarchy nodes have a greater overview of the rest of the network, hence they can make more informed decisions as they get more filtered and transformed data as compared to the previous levels. It is only obvious that Cloud is be at the top hierarchy of this pyramid as it has the capability to turn data in knowledge. Nodes present at the same level can communicate among themselves and should be able to carry out offloading and handing over tasks depending upon their workload as they are assumed to have the same intelligence level. It is imperative to discuss different deployment models for fog

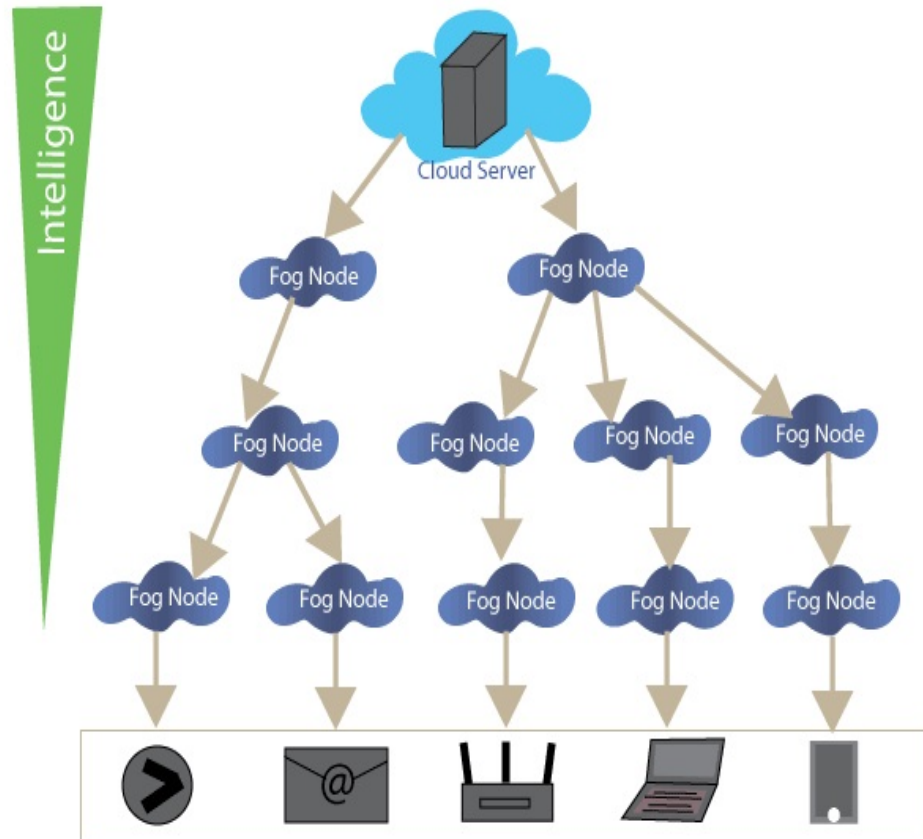


Figure 2.1: A multi-layer hierarchical Fog model with Cloud at the top and IoT devices at the bottom. The intelligence of the system decreases from top (Cloud) to bottom (IoT devices). Each intelligence group present at a higher level is more capable of performing complex tasks than the preceding layer.

while discussing its reference architecture, as of yet there are no formal deployment models defined. Hence various deployment models of fog can be used to gain perspective on the structure of Fog as shown in Table 5.1.

Table 2.1: Deployment models for fog computing

Deployment Model	Description
Public Fog Model	Much like public cloud, public fog provides services for the general public where the people can rent the necessary services they require. Public infrastructure in case of cloud is owned by a single entity, but it might not be the case for fog. Various organizations need to collaborate to create this deployment model.
Private Fog Model	Private cloud provides services to the high-end security companies, these services are costly as compared to the public fog. A single entity can own private fog models and provide services to other parties as needed.
Hybrid Fog Model	As the name suggests, hybrid fog is the result of the collusion between public and private fog. This model is appropriate for applications with diversified security levels, sensitive info can be passed on to private fog and non-sensitive info to public fog.
Community Fog Model	This model reduces the cost of deployment by proposing a community scheme where two or more companies can collaborate to achieve cheaper services as compared to private fog but more security as compared to public fog.

2.4.3 Applications of Fog

Fog shows a promising future with hundreds of IoT applications running smoothly across integrated smart systems. It will change the face of IoT applications such as e-health, smart cities, automated traffic control etc. and make them more accurate and trustworthy for users

to relay upon [15,16]. It has been a hot topic in the recent years due to its the ease of adoption and flexibility which has attracted a lot of tech giants to invest in it.

The Linux foundation has launched a edge computing platform to minimize the gap between IoT devices and cloud called EdgeX Foundry [17]. One of its key features is that it can aggregate data from various IoT devices with heterogeneous underlying technologies, it standardizes the data and makes collaboration possible between different protocol IoT networks.

Similarly, Amazon also provides AWS Greengrass [18] services to the IoT devices. Its main focus is to filter out all the unnecessary data in queue to reach cloud and sends only that data which needs deep processing or storage.

Microsoft has also joined the trend and launched Azure IoT Edge [19] as its solution to edge computing, extending cloud-like services to the edge of the network improving network reliability and bandwidth.

2.4.4 Research Direction

Fog offers numerous benefits due to its diverse nature, such as scalability, enhanced privacy, mitigating dependability on a central entity, reducing the probability of single point of failure, reduces bandwidth and storage cost, results in a more context and location aware system and makes it possible to analyze data in transit. However, as of yet it is a vast domain ready to be researched upon to minimize the lot of privacy and security issues that are in the way of its world wide adoption. The flexibility and diversified nature of fog make it more susceptible to various network attacks such as man-in-the-middle attack, injection of malicious nodes in the network etc. [13].

This research exploits the lack of a secure and efficient solution to protect fog domain from losing valuable data though minor breaches in the network. Trust management has been proved to be an authentic and cost effective way to improve the security and privacy of various domains such as IoT, SIoT and Cloud. There is limited work available on trust management schemes in Fog, hence we propose a two-way trust management scheme for fog computing that will validate the identity of the node providing the service as well as of the node requesting the service.

2.5 Motivation

Let us consider smart cities as an example to highlight the need and use of Fog in the future of internet. The concept of smart cities is simply to provide services to its residents to better build the city infrastructure. It promises optimum utilization of resources and balances it with providing its residents with unparalleled quality of life. From smart grid stations to smart traffic management systems, smart city offers various over the top applications in order to provide optimum services.

The issues with smart city deployment is the excessive cost, as each application needs an independent network for communication called *silos*, which is expensive and not an optimum solution as it leads to fragmentation of services. There is no standardized solution that enables smart cities to fully take advantage of all its applications. But fog computing may provide a better solution, by laying out a single network infrastructure for all the applications to communicate over. By providing public fog services to the cities, third party entities will not need to build and maintain their own infrastructure which will help reduce the cost by many folds.

Fog networks are generally large-scale networks comprising of various network objects also called fog nodes (i-e, any device with sufficient processing power and memory), these nodes are bound to communicate with each other for various transactions, increasing the probability of attacks. The proposed two way trust management scheme should ensure that both the nodes have established a trusted connection prior to the transaction.

LITERATURE REVIEW

The *level of confidence that a device will behave in a satisfactory manner* is known as trust. This chapter will provide an insight to the importance of trust in IT technology, its key features and management models. Section 3.1 will explore the importance of trust in IT technologies followed by trust in SIIoT and its similarities with fog environment, discussed in great depth in Section 3.2

3.1 Trust in the light of IT technology

In an IT environment, multiple devices from various places connect to each other, share data and resources. Much like humans need to develop trust before sharing among themselves to minimize the chances of sharing important information with a fraudulent party, devices in an IT environment must trust one another to minimize the chances of breach. Trust establishment in such an environment is more than just a security concern, a trusted system is considered to be more reliable, integrous, dependable and capable of performing a given task. An honest collaboration always ends in the highest productivity.

Trust is a subjective term, it is defined as the manner in which an object act which is deemed satisfactory for you. In an IT environment, a set of policies and protocols will determine the satisfactory manner an object should behave to earn the title of *trustworthy*. Secondly, trust needs to be updated frequently, much as humans change their nature with time, trustworthy devices can also become untrustworthy over time. Hence, a device must not only rely on past observations rather it must incorporate a method to predict future behavior of other devices to keep the trustworthiness up-to-date.

As discussed above, trust is subjective in order to establish trust one must first quantify the level of assurance you have on a device that it will behave according to the set of rules defined. This process depends upon the type of application in question, for example, a health application will require a higher degree of protocols whereas a smart street lamp will require low degree of protocol. In other words, trust in IoT depends upon the deployment model,

environment, level of security required, and type of application used.

In distributed systems such as a fog network, the task of establishing trust becomes even more strenuous, as a fog network consists of devices from diverse backgrounds that may have never communicated with each other before. Trust in this case is difficult to develop as hostile entities might be present in the network waiting to exploit any moment of weakness. However, if established it becomes pertinent in distributed systems.

There are two major elements of a trust: a trustor and a trustee. Trustor being the entity that requires a service and trustee is the entity that will provide a service. Trust gives the trustor enough confidence that the trustee will provide the best service without compromise and behave according to the protocols defined. A trust management system is where it all comes down to, this system provides means to collect, calculate and propagate trust values throughout the network. It consists of a comprehensive scheme that lets an object decide for itself whether the other object is trustworthy for communication or not.

3.2 Trust Management in SIoT

The most important part of any trust management system is the trust computation block, it determines the requirements of assurance necessary to establish trust and tells how to quantify them. This block deals with raw values, trust requirements and final trust values as a result of trust calculations. We will discuss trust management in SIoT in great depth due to its similarity with fog computing, SIoT liberates the devices to request and receive services from one another. Hence, this section will present a thorough study of trust management in SIoT to form a trust management scheme for fog.

Service- oriented IoT (SIoT) is a combination of IoT with social networks, where the objects of the network form social relations like humans do. The objects frequently communicate and collaborate among themselves based on a level of assurance that the other object is trustworthy. SIoT can be best understood as a peer to peer network where a service requester (SR) may ask for a service from a service provider (SP). In an honest network the SR will have full confidence that the SP is trustworthy and will provide legitimate services.

Figure 3.1 presents an example of how trust assessment is carried out in SIoT. It shows a scenario where the service requester(SR) has no experience with the service provider(SP) prior to this instance, in such a case SR will ask its neighbors for their recommendations of

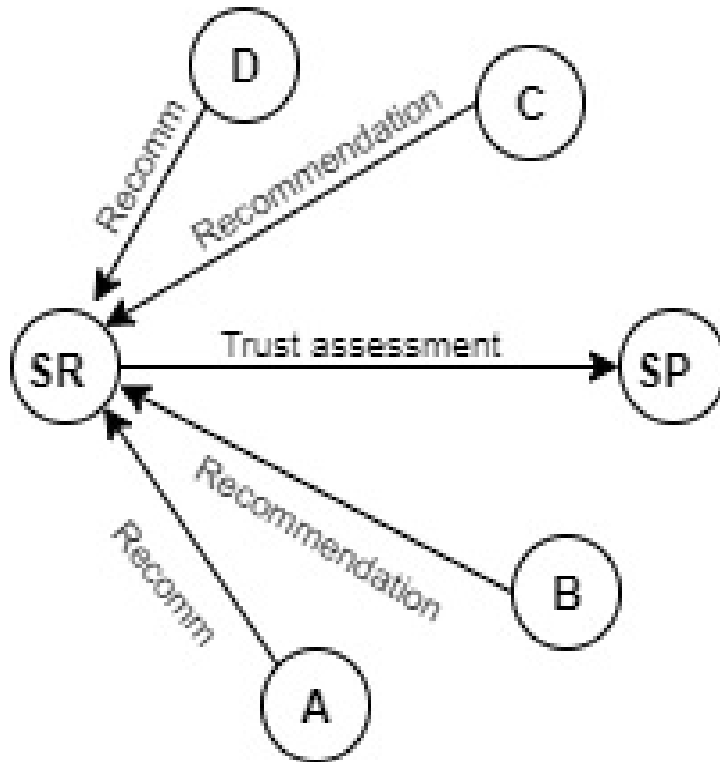


Figure 3.1: An outlook of how devices in a network may work. SR determines the level of trust of SP by gathering recommendations from its neighbors.

their experiences with the SP. The recommenders A,B,C and D will send their trust assessment of SP to the SR, it is assumed that SR will have established some level of trust with its recommenders through previous interactions. SR will combine it's own experience with the recommenders, their level of trust and the recommendations for SP to form its opinion and connect with it.

Trust helps objects make independent decisions without human intervention. It provides a safe and honest environment for network objects to request and avail services from each other. There are two most common scenarios; first, an object that has previous experience with the other object and second, where an object has no prior experience. In the former case, the object can easily formulate trust value giving its own experience the highest weightage, whereas the later case has been discussed and explained in figure 3.1. One of the main requirements of a distributed fog network is to help the fog nodes make independent decisions and collaborations among themselves based on some level of trust assurance that their connection will not suffer with respect to reliability and availability.

Trust management in SIIoT is very similar to trust management in fog. The goal of trust is for the SR to know that the SP is safe to connect and communicate with in terms of

reliability and availability and vice versa. Hence, all the devices that are part of the fog network must have a certain level of trust among them for successful collaboration and communication. Fog requires trust to be established at both ends of the communication i.e. the service requester and the service provider both should only provide and avail services if they trust one another [10]. We will explore the trust dimensions and trust requirements to build a premise for our proposed trust management framework for fog computing.

3.2.1 Trust Dimensions

The authors in [20] surveyed a large number of trust management models in SIoT and categorized trust computation into five trust dimensions. The categories are as follows: Composition, propagation, update, formation, aggregation. This section describes these trust dimensions in more depth as given by Table .

Table 3.1: Categories of Trust computation in SIoT

Trust Dimensions	Description
Trust Composition	This category determines the trust indicators to be used for trust computation. It's further subdivided into quality of service (QoS) trust and social trust.
Trust Propagation	The trust propagation category determines whether the system will use a distributed or a centralised approach, each has their own pros and cons.
Trust Update	This dimension deals with the frequency with which the trust values will be updated. A social network is highly unpredictable, even a highly trusted node can turn.
Trust Formation	The formation dimension as the name suggests, determines how many trust indicators will be combined and in what manner and formulation.
Trust Aggregation	The aggregation dimension determines how recommendations from the neighboring nodes will be combined with the personal experiences.

3.3 Trust Requirements

We have carefully studied the trust requirements in Mobile Ad-hoc Networks (MANETs) [21] and established that they can be used for defining trust requirements for fog computing

as well. These requirements are as follows:

- Fog networks are dynamic in nature, they continue to change with old nodes breaking off and new nodes connecting to the network. This scenario suggests that trust should be **dynamic** in nature.
- Fog network is a highly diversified network, compatible with different applications on the same platform. This implies that trust value for service A provided by node B will not hold for service B provided by it. Hence trust needs to be **subjective**.
- Each object is ruled by its own set of policies, so if node A trusts node B, and node B trusts node C then it is not necessary that node A will trust node C as well. Which implies that trust may or may not be **transitive**.
- Similarly, trust is also **asymmetric**. If node A trusts node B then node B may or may not trust node A.
- Trust in fog should be **context dependent** as a single fog node may offer services to a variety of applications. It is necessary to understand that if a node is trustworthy for a traffic congestion app it may not be trustworthy for a health app.

The discussion under this section can be summarized to a single observation that trust in fog computing is necessary to ensure a secure and safe fog computing environment. The highlighted advantages of introducing trust in fog are as follows:

- It will improve reliability, allowing the fog nodes to predict the future behavior of each other. So, a client may be able to select a fog node in its vicinity that will provide the best service.
- It will allow monitoring of the distributed fog network, malicious objects and rogue nodes will be detected and avoided.
- It will strengthen the basis for collaboration, fog nodes will be able to offload to other fog nodes with the necessary processing power.

3.4 A Survey of Trust models in IT environment

Trust has been around for the longest time as a basic human instinct that needs to be present before relying on fellow human beings. It has been studied extensively in almost all fields including economics, philosophy, sociology and now it has found its way in the IT environment. Trust is a blessing in disguise for an environment with risk, uncertainties and collaborations with unknown entities such as the fog network [22]. Fog is a relatively new and growing computing paradigm and only a handful of work is available on trust management in Fog but due to its similarities with SIIoT [23] the related researches have been thoroughly studied and reviewed to build a trust management system suitable for a fog environment. Trust in Fog is more than just secure transfer of data, it encompasses the canons of integrity, consistency, truthfulness and reliability of a party on its service provider. The existing mechanisms help the node make a wiser decision for secure data transfer depending upon various trust metrics and QoS parameters. In this section, each technique will be discussed briefly.

A very popular trust composition technique is Bayesian inference, which considers trust to be a random variable and using it creates a probability distribution model. The parameters of this probability distribution are updated according to new recorded observations. This technique was first used by Josang et al. [24] to create a reputation system, it models trust as a random variable $\epsilon [0,1]$ and follows the beta distribution curve; each set of positive and negative experience is mapped to the probability distribution parameters (α, β) and the final trust is computed by taking its average. The same model was used by Ganeriwal et al. [25] to propose a reputation system in a wireless sensor network, this model takes the negative and positive experiences as input, using it formulates a sensor node reputation score. The system has two major blocks; watchdog and reputation block. Watchdog is responsible for monitoring and branding the nodes into cooperative and uncooperative based on probability values, whereas the reputation block is responsible for combining the recommendations from neighboring nodes using beta reputation function. The following model protects against bad mouthing and ballot stuffing attacks and addresses scalability; a chain of trusted nodes is used to calculate the trust value of a distant node. But the model neither incorporate the direct observations of the trustor node nor the position of trustee in the network. Another system proposed in [26] uses subjective logic to build an opinion model, where trust is

combined on the basis of confidence the trustor has on the recommendation of its neighbors. The major drawback of this model is the overhead as it consistently needs opinions to form inputs. Authors in [27] combine fuzzy logic and Bayesian belief theory to develop the trust model using various performance metrics such centrality, mobility etc. Chen et al. [28] also used a fuzzy membership function to measure the uncertainty of the trust model.

The trust management system for social objects is entirely based upon their social relations and the interactions between them, the authors in [29] make use of a similar approach by combining various social metrics such as direct observations, centrality, community of interest (CoI), Cooperativeness etc. as the trust indicators for trust composition. Using social trust metrics makes this scheme help detect and protect the network better against on off selective forwarding attacks. A centralized system is proposed in [30] where a guarantor is hired to measure the level of assurance of the SP by the help of a reputation function. This model neglects the direct observations altogether only focuses on the reputation of the object, it's not suitable for low latency applications. Martinez-Zulia et al. [31] combine the QoS and social trust parameters as the components of trust composition, and weigh the recommendations of the raters using credibility for indirect trust aggregation. Similarly, Chen et al. [32] weighs the recommendations of raters using similarity index the SR has for its raters.

Belief theory is another popular trust composition technique, Yu et al. [33] adopts it as the basis for computing trust in autonomous systems (ASs). This model has two elements to it; the response collection of the proposition (theory) and the aggregation of the responses based on subjective probabilities. In another one of his researches Josang et al. [34] describes that a node's estimation of trust by another node depends upon 4 main factors (b-belief, d-disbelief, u-uncertainty and a-absence of evidence), where the sum of these factors is unity. In a groundbreaking research Ries et al. [35] laid the foundation for evaluation of propositional logic terms under uncertainty, compliant with standard probabilistic approach and subjective logic. Based on this approach Ries [36] proposed a context dependent model; CertainTrust that helped agents select trustworthy partners in a risky engagement. Wang et al. [37] applied logit regression to determine the relation between cumulative evidence gathered by the nodes considering environmental context variables including energy-sensitivity, capability-limitation, and cost-awareness.

Table 3.2: Comparison and Analysis of Literature Review

Research Paper	Trust Metrics	Trust Model	Main Contributions
Service-oriented Internet of Things (SIoT)			
[32]	1. QoS 2. Social Trust	CoI-based Trust Model	A scalable, capacity limited feedback-based trust model that uniquely combines the direct and indirect observations with minimum convergence time
[31]	Social trust parameters	Behavior based model	Builds a reliable SIoT network by incorporating its trust metrics and updating trust with minimum overhead
[30]	Reputation	Guarantor based trust model	A reputation model involving a 3rd part guarantor
[39]	1. Direct Trust 2. Indirect Trust	Communities of interest based trust model	A hybrid trust model that integrates social behavior of objects with their communities to model trust
[40]	1. Direct Trust 2. Indirect Trust	Context based model	Combines social relations of objects with context and capacity of the objects
Wireless Sensor Networks and Mobile ad-hoc networks (MANETs)			
[25]	Positive and negative feedback	Feedback based reputation model	A reputation system in a WSN, takes negative and positive experiences as input and generates the reputation scores of each node.
[28]	1. QoS 2. Social Trust	Behavior based trust model	Calculates trust based on positive and negative experiences and the uncertainty using the fuzzy membership function
[37]	1. Energy sensitivity 2. Capability limitation 3. Cost awareness	Behavior based trust model	Applies logit regression to learn the relation between cumulative evidence gathered by a node toward another node including the corresponding environmental context variables.

Bao et al. [38] uses the ownership as an input to his trust management framework, which is a decentralized CoI based model where nodes divide among inter- and intra-communities. A similar model [39] requires each community to have an administrator duly elected by the nodes of that community. The admin is the central entity responsible for managing service requests, calculating the trust values and for seamless running of the network. A context-

based trust management model was proposed by the same author in [40], the design catered scalability and accuracy in its system grounded on the context of the network. Each node has multiple trust values depending upon the type of service and context, the model works well for dynamic networks, but the overhead may be an issue in larger networks. Truong et al. published a series of researches in [41, 42], in which he proposed a model based on experience, reputation and knowledge. The following model mimics the human cognitive process which fits perfectly in a SIoT environment. A few models in social internet of things (SIoT), wireless sensor network (WSN) and mobile ad-hoc networks MANETs are briefly explained in table 3.2

Cloud computing has some very well-established trust management models, but they cannot be directly applied to fog computing due to various reasons such existence of a distributed network in fog and lack of mobility support in cloud. One of the major issues is the absence of a central authority to verify and monitor the attributes defined to measure the trust of a service provided by the fog node. In a fog network, the fog clients and the fog nodes are vendor specific and dynamic in nature due to which reputation-based trust model cannot be implemented in fog. For the implementation of SLA based trust model in fog a licensed third party is required to monitor the and validate the SLAs for the fog clients. Trust in cloud computing is a unidirectional requirement due to its in-place security mechanisms that ensure trust between the cloud users and the cloud service providers, whereas it is a two-way requirement in Fog due to its open and flexible nature. This two-way requirement of trust in fog network makes it a difficult task to design and implement a trust model [10].

The existing literature on trust computational models in fog environment is limited, few research papers available have been thoroughly studied and analyzed including [43, 44, 45, 46, 47, 48]. S.A. Soleymani et al. in [43] propose a model that combines experience and plausibility using fuzzy logic to compute trust. The proposed model authenticates, calculates and chooses the most trustworthy node using a set of modules. Wang in a series of research articles [44, 45, 46] use regression analysis and fitting function to model a trustworthy communication, the model creates a trustworthy connection by mapping trust values with communication variable in a sensor cloud system using fog based approach. The author extends his research in [45] and propose a hierarchical model as opposed to the linear model proposed in [44]. In both the researches the author uses fog layer to calculate the trust func-

tion, store its value and execute tasks based on the value of trust. Authors in [47] present a lightweight scheme that used feed-backs from multiple sources to identify trustworthy IoT edge devices. All the above-mentioned researches use fog as a supplementary layer to reduce the computational cost and to enhance the storage capability of other networks. None of the aforementioned models primarily focus on fog and its structure to create a trust model. Our trust model's primary focus is to ensure the trustworthy transfer of data in peer-to-peer communication between fog nodes. Rahman et. al. [48] propose a broker-based trust evaluation framework based on fuzzy logic. This framework uses only QoS parameters and do not incorporate the social relationship of the nodes. The existing models in fog are listed in a comprehensive manner in table 3.3.

The motivation behind creating a trust management model is to complement the rapidly growing fog network, by introducing trust in fog we are in fact moving towards a more reliant and secure platform. Our work is different than the said models as it primarily focuses on fog computing paradigm while uniquely combining the observations, reputation and other trust metrics necessary in a fog network, it also presents a two-way trust system, where both the fog nodes involved in the transaction establish trust before connecting. Our model is independent of any third-party involvement which is one of the requirements for the fog network, our reputation system is based on Bayesian inference with discounting factor explained in more detail in the next section [24,25]

Table 3.3: Analysis of Trust Models in Fog

Research Paper	Trust Metrics	Trust Model	Main Contributions
Fog Networking Environment			
[43]	1. Experience 2. Authentication	Fuzzy trust-based model	The model uses a set of modules to authenticate, calculate and choose the most trustworthy node
[44, 45, 46]	1. QoS 2. Position 3. Unique identifier	Multiple linear regression model	A model for trustworthy communication using regression analysis and fitting function that relates trust value with the communication variable
[47]	Multi source feedback	Multisource feedback model	A lightweight scheme that uses feedbacks from multiple sources to identify trustworthy IoT edge devices.
[48]	QoS parameters	Broker based model	A broker-based trust evaluation framework based on fuzzy logic

PROPOSED TRUST MANAGEMENT FRAMEWORK

Trust is used as an assessment criterion to determine the security level of a respective object, but the situation of trust in fog infrastructure is rather complex due to various reasons [49]. The acceptability and deployment of fog has amplified the need for a secure and efficient method for data transfer and reliable service provider selection. One of such methods is by establishing trust between the network entities [14]. A great deal of work has been carried out in cloud computing paradigm mitigating the security and privacy challenges prevailing in the said infrastructure. However, none of these models can be readily applied to fog computing due to distinct dissimilarities in both computing paradigms. This section will explain the proposed trust management framework, its mathematical model, trust indicators and algorithm.

4.1 Social Qualitative Trust Management Framework

There are three shareholders in a fog environment; the IoT devices (generates data), the fog nodes (transfers and filters the data in transit) and cloud servers (processes and stores the data). This research focuses on establishing trust between the fog nodes for secure transfer of data. For simplicity, we will consider a single layer fog architecture without compromising on any of its key features to build our trust management system. Each fog node is connected to a set of IoT devices based on its position and the type of services it offers. A fog node can always communicate with its neighboring fog nodes to exchange trust values. A simplified single layer fog architecture is used to build a collaborative trust management system model as shown in figure 4.1.

The proposed solution establishes trust among fog nodes so that they can collaborate for offloading, data sharing and other services. Ideally, for two fog nodes to request a connection they must be present in the same area, or within each other's geographical range. As discussed in the previous sections, trust is a two-way requirement in fog environment as both the fog nodes; the SP and the SR, must validate each other for security before sharing any

TRUST MANAGEMENT SYSTEM

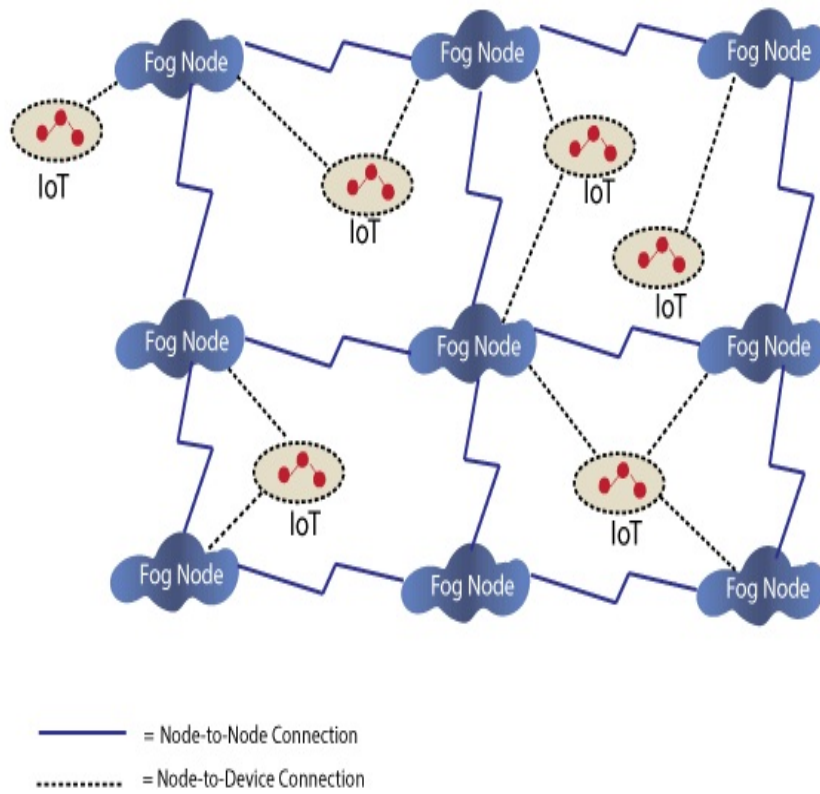


Figure 4.1: Simplified single layer fog environment with each fog node connected to an IoT device. The fog nodes can communicate with their neighbors to share trust values.

data. This will prevent a rogue fog node from entering and hiding in a trusted network. In our proposed scheme the SR node calculates the trust value of the SP node by asking its neighbors for recommendations and combining them with its experience, this approach makes it easier to prevent a connection between an honest and a rogue node. Moreover, Each node will maintain a rank which decreases every time a node fails to provide a decent service; malicious nodes are removed from the network when their rank falls below the set threshold. The malicious nodes are blacklisted and broadcasted to the network to avoid any discrepancies in the future.

When a fog node requests a service from its neighboring nodes for connection, the available SP node will first validate the authenticity of the requestor before proceeding to accept or decline. The SP node will follow the same procedure as the SR node did to calculate trust, it asks its neighbors for recommendations, combines it with its experience and formulate the trust using the beta reputation system [24]. A predetermined threshold determines whether the connection request will be accepted or declined. In a fog environment it is necessary for both the parties involved in a transaction to validate one another, as any device can become a fog node increases the probability of malicious nodes entering the network. Therefore, the SR node will also validate the SP node to ensure authentic service before the actual data exchange. To reduce the overhead of the trust management system, each node will exchange the trust values with its neighbors only without effecting the efficiency of the system [50].

The step by step procedure from the connection request to data exchange is as follows:

- The SR node will ping all of its neighbors with a service request.
- Depending upon availability, one of its neighbors will respond as a potential SP. However, it must first ensure the authenticity of the request for protection against rogue SRs.
- The SP will ping its neighbors for recommendations of SR, and combine them with its own experience to obtain a trust value.
- If the Trust value for SR is below the threshold, its request will be declined immediately. The rank of the SR will decrease (if rank of a fraudulent node becomes zero it is then exiled from the network).

- The SP will update and store the trust value of any malicious node detected for future reference.
- Once SP has authenticated the request, SR will ensure the validity of the SP before connecting to avoid malicious payloads rather than the legit service.
- SR will validate SP on the same lines as discussed above.
- After validation of SP and SR at both ends, a trusted connection is established, and real communication can take place.

A good TMS does not require the trust value to be propagated throughout the whole network, each node can broadcast its experiences to its neighbors instead for better efficiency.

4.2 Trust Composition using Bayes Model

The trust requirements defined in Mobile Ad-Hoc Networks [25] explained in section 3 can be used to derive the trust requirements for a fog environment which view trust as dynamic, subjective, intransitive and asymmetric. Trust computation must yield a dynamic value of trust in conjunction with the trust requirements and the design dimensions [24]. The proposed SQT framework takes on a distributed approach, it uses social and network parameters for trust computation, and follows a multi-trust and event-driven trust update.

By virtue of its nature, trust is asymmetric and subjective which implies that each node in a network experiences it differently based on its limited data set, the best way to calculate trust in an architecture such as Fog is by using Bayes Model [25]. In this model, trust is a random variable determined by the parameters of probability distribution curve, and updates to it are based on the new observations. The advantage of using Bayes model in a fog environment is that it allows a node to predict the future behavior of the others based on its current observations. The heart of Bayes model is the beta probability density function, which gives it a sound mathematical base for feedback accumulation.

Consider a process that may have two parameters y, y' as its possible outcomes such that γ represents the number of times y is observed and $\hat{\gamma}$ represents the number of times y' is observed then to observe the future behavior of the process the parameters of the beta distribution function can be set as:

$$\alpha = \gamma + 1 \quad (4.1)$$

$$\beta = \hat{\gamma} + 1 \quad (4.2)$$

Where $\gamma, \hat{\gamma} \geq 0$. Hence, the probability expectation of the beta distribution is given by:

$$E(p) = \alpha / (\alpha + \beta) \quad (4.3)$$

Now, consider four nodes (A,B,C & D), where node A & node B are trustor and trustee respectively, and node C & node D are the recommenders as shown in Figure 4.2. Now, by using the equation 4.3 the reputation function is expressed as:

$$E(\phi(p|\gamma_{AB}, \hat{\gamma}_{AB})) = \frac{\gamma_{AB} + 1}{\gamma_{AB} + \hat{\gamma}_{AB} + 2} \quad (4.4)$$

Equation 4.4 gives the subjective reputation of node B from node A's perspective, it is not possible to calculate the objective value of reputation as each node has different experiences with B.

4.2.1 Discounting Operation

The node calculating trust assigns different weights to the recommendations it receive depending upon its own relation with the recommender nodes, this process is called discounting. If a node does not trust one of the recommender nodes then it can completely ignore its recommendation, similarly it may give more weight to the recommendations of a trusted node as compared to a node with low trust value. This technique helps evade trust-based attacks such as bad mouthing and ballot stuffing attacks. As shown in figure 4.2, let us suppose node A is the trustor gathering reputation of node B on the recommendation of node C, then the beta distribution given by α_{AB} and β_{AB} will be as follows:

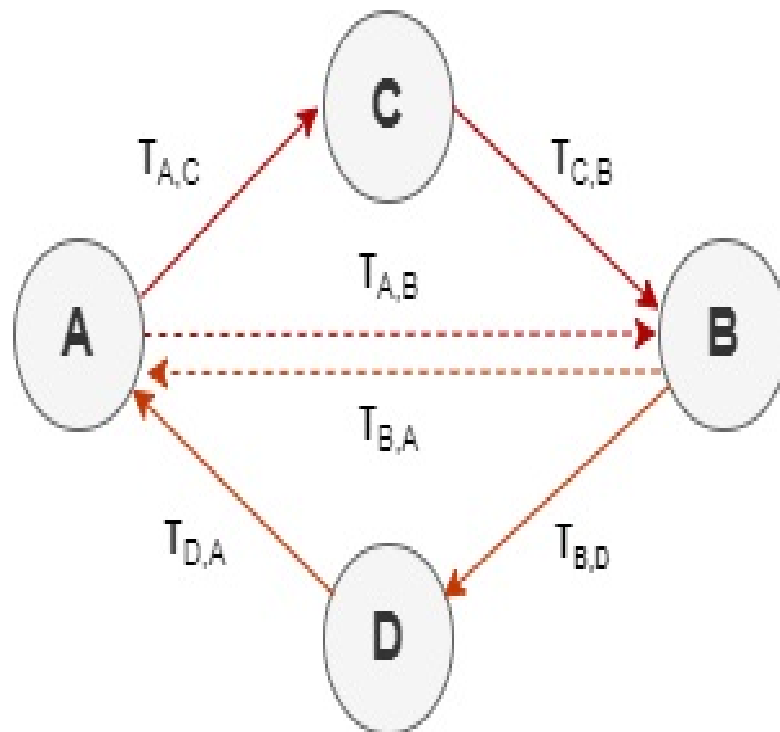


Figure 4.2: Nodes A and B are the trustor and trustee nodes, to establish trust they need to validate each other by taking input from the recommender. The recommendations are weighed according to the relation of the nodes with the recommenders, this concept is popularly known as Discounting operation.

$$R_{AB} = \frac{\alpha_{AB}}{\alpha_{AB} + \beta_{AB}} \quad (4.5)$$

$R_{AB} = R_{AC} \otimes R_{CB}$, where \otimes is called the discounting operator. The α_{AB} and β_{AB} parameters in equation 4.4 are updated as follows:

$$\alpha_{AB} = \alpha_{AB}^{prev} + \frac{2\alpha_{AC}\alpha_{CB}}{[(\beta_{AC} + 2)(\alpha_{CB} + \beta_{CB} + 2)] + 2\alpha_{AC}} \quad (4.6)$$

$$\beta_{AB} = \beta_{AB}^{prev} + \frac{2\alpha_{AC}\beta_{CB}}{[(\beta_{AC} + 2)(\alpha_{CB} + \beta_{CB} + 2)] + 2\alpha_{AC}} \quad (4.7)$$

Equations 4.6 and 4.7 represent how the α_{AB} and β_{AB} parameters depend upon the node's trust on the recommender. Discounting helps achieve an unbiased reputation of the trustee node which helps in evading many trust-based attacks.

4.3 Mathematical Description of Proposed Scheme

4.3.1 Trust Metrics

A trust indicator is an essential part of trust composition as it determines the properties based on which the value of trust is calculated. Trust indicators are shortlisted depending upon the requirements of the system, there is no hard and fast rule to include or leave out a certain type of indicators for trust composition. Generally, more than one parameter is required to build an effective trust management system. Our model considers various social and qualitative trust metrics for its trust calculation; the social metrics, include direct observations made by the trustor as well as the recommendations by the neighboring nodes. This section will briefly discuss the trust metrics of our proposed SQT management system.

4.3.1.1 Direct Trust

The experience of the trustor after a successful transaction with the trustee determines the direct trust. In a trust management system, it is imperative for a node to have the ability to calculate individualistic trust for an unbiased decision. Direct trust holds maximum weightage in our proposed system to decrease the effect of various false recommendation attacks.

D_{AB} denotes the trust of node B as calculated by node A for transaction k. The relevance of transaction k is given by transaction factor $tf_{AB}^k \in [0,1]$ between the two nodes. The feedback of node B given by node A is represented by $f_{AB}^k \in [0,1]$ then the formula for n transactions is given by [29]:

$$D_{AB} = \frac{\sum_{k=1}^n tf_{AB}^k f_{AB}^k}{\sum_{k=1}^n f_{AB}^k} \quad (4.8)$$

4.3.1.2 Reputation Function

The reputation metric is important due to various reasons, firstly it helps the trustor determine the image of the trustee, secondly it plays a pivotal role when there has been no prior transaction between them. In this scenario the trustor greatly depends on the reputation of the trustee in the network. The SQT system model uses Bayesian inference to combine feedback from the recommenders. The simplicity and flexibility of the Bayesian formulation qualifies it as the best approach for this model [24]. The reputation of node B as perceived by node A is given by equation 4.5.

The reputation function is completed by the discounting step that predicts the future behavior of the node B as seen by node A, in this case, based on its past behavior. This protects against the network feedback attacks such as bad mouthing and ballot stuffing attacks. The discounting function is given by equations 4.6 and 4.7 as discussed in section ??

4.3.1.3 Degree Centrality

Degree centrality represents the number of direct connections a node has in a network, higher degree centrality means the node has great importance within its network. The reputation function will be influenced by the degree centrality value of a node, if node B has a higher degree centrality then its recommendation would be higher and vice versa. To minimize the effect of centrality on the reputation function we calculate C_{AB} :

$$C_{AB} = X_{AB} \cap X_A \quad (4.9)$$

Where, X_{AB} and X_A represent the mutual friends of node A and B and the friends of node A respectively.

4.3.1.4 Service Score

A reward and penalty metric is added to make the SQT system to provide an extra layer of protection against malicious nodes.

$$S_B = \begin{cases} 1 \times wt_s \text{ reward} \\ -1 \times wt_s \text{ penalty} \end{cases} \quad (4.10)$$

Where wt_s represents the weight of the service.

4.3.2 Mathematical Model

4.3.2.1 Bayes Trust

The Bayes model [24] defines trust in terms of collective desirable and undesirable behavior of the trustee as observed by other nodes, it is given by the following equation:

$$ET_B = \frac{\alpha_B + 1}{\alpha_B + \beta_B + 2} \quad (4.11)$$

Where α_B and β_B denote the desirable and undesirable behavior of the trustee (node B in this scenario) respectively. Bayes trust showcases the predicted behavior of the trustee, in other words it is the expected behavior of the node, this value might not always be the same as the computed trust as the object which is malicious might become innocent in the future and vice versa.

4.3.2.2 Calculated Trust

The calculated trust is determined by combining all the trust metrics defined by equations 4.5 4.8 4.9 4.10 in the following sequence:

$$T_{AB} = \delta D_{AB} + \sigma R + \omega C_{AB} + \theta S_B \quad (4.12)$$

where δ, σ, ω and θ are the weights assigned for normalizing the data.

4.3.2.3 Final Trust

The final trust of the node B is calculated by:

$$T_{final} = (ET_B * T_{AB}) - m_e \quad (4.13)$$

m_e is the marginal error, hence it is taken out from the final equation. The final trust value in our proposed model is given by the product of Bayes trust with the calculated trust. ET_B is the predicted future behavior of the node in light of its calculated trust, for example if the calculated trust $T_{AB} = 0.8$ then the expected behavior of the node is desirable $\therefore ET_B \geq 0.5$.

4.4 Algorithm of the Proposed Model

The detailed working of our proposed model is described in Algorithm (1). The *threshold* for our proposed model is 0.5, but it can be higher for more critical applications.

Node A initiates the communication with node B by sending a service request. The node B computes the trust of node A by running the algorithm, the trust metrics are assigned appropriate weights for calculation of T_{BA} . Lastly, it is combined with the Bayes trust to get the value of final trust. If $T_{BA} \geq threshold_1$ then it will allow the communication to proceed. After getting the connection approval node A will compute the trust of node B, it will follow the same steps; assign weights to the trust metrics, calculate T_{AB} and combine it with Bayes trust to get the final trust. Now, if $T_{final} \geq threshold$ then a trusted connection is established between the two nodes. Our model requires both the SR and SP to establish trust before creating a connection, if either one of nodes fail to establish trust then a trusted connection will not be created and node A will look for other SPs in its neighborhood.

Algorithm 1 Trust Computation Algorithm

Input: [N1,N2,N3,N4,...][$\delta 1, \sigma 1, \omega 1, \theta 1, \delta 2, \sigma 2, \omega 2, \theta 2$]**Output:** Trusted connection

```
for  $i$  between 1 and  $n$  do
   $Nb$  = neighborhood size of  $i$ 
  for  $j$  between 1 and  $Nb$  do
    if  $Ni$  has capacity & rank  $\geq 1$  then
      /*Trust metrics calculation
       $D = \frac{\sum_{k=1}^n t f^k f^k}{\sum_{k=1}^n f^k}$ 
       $R = \frac{\alpha}{\alpha + \beta}$ 
       $C = X_{AB} \cap X_A$ 
       $S = \begin{cases} 1 \times wt_s \text{ reward} \\ -1 \times wt_s \text{ penalty} \end{cases}$ 
      /* calculated trust of trustor node (node A)
       $T_{BA} = \delta 2 D_{BA} + \sigma 2 R_{BA} + \omega 2 C_{BA} + \theta 2 S_B$ 
      for  $k$  between 1 and  $Nb$  do
        /*discounting step
         $T_{BA} = T_{BC} \otimes T_{CA}$ 
      end
      /* Bayes trust of trustor node (node A)
       $ET_B = \frac{\alpha_A + 1}{\alpha_A + \beta_A + 2}$ 
      if  $T_{BA} \geq \text{threshold}$  then
        /*calculated trust of trustee node (node B)
         $T_{AB} = \delta 1 D_{AB} + \sigma 1 R_{AB} + \omega 1 C_{AB} + \theta S_B$ 
        for  $k$  between 1 and  $Nb$  do
          /*discounting step
           $T_{AB} = T_{AC} \otimes T_{CB}$ 
        end
        /*Bayes trust of trustee node (node B)
         $ET_B = \frac{\alpha_B + 1}{\alpha_B + \beta_B + 2}$ 
         $T_{final} = (ET_B * T_{AB}) - Me$ 
        if  $T_{final} \geq \text{threshold}$  then
          | Trusted connection is established
        end
      end
    end
  end
end
```

4.5 Attacks on Trust Management systems and the Resilience of SQT to these attacks

The aim of a trust management system is to build a trustworthy network with high trust threshold, in such a system the malicious objects tend to perform efficiently to get a good rank in a network only to create mischief later. A few most common types of these attacks are given below along with a sound justification on how our proposed model is resilient towards them.

Table 4.1: Trust based network attacks and resilience of our proposed solution to them

Attacks	Description	Justification
Self Promotion Attack (SPA)	The Self-promotion attack as the name suggests is when a malicious node gives false good recommendation about itself when requested for service and performs poorly once selected.	The proposed SQT management system is resilient to this attack as it does not allow any node the power to self-recommend.
Bad Mouthing Attack (BMA)	This is a type of collusion attack where several nodes team up to spread false recommendations about an innocent node.	This attack is addressed by our feedback aggregation method, it considers the level of trust between the trustor and the recommender weighs the recommendation accordingly.
Ballot Stuffing Attack (BSA)	This is also a type of collusion attack, much like BMA, where malicious nodes collude to propagate false recommendations to build the reputation of another malicious node which might damage the network if selected.	The justification for BMA also applies to this attack.
Opportunistic Service Attack (OSA)	It is an attack where a malicious node deliberately performs good services to regain its reputation once it feels that its reputation is decreasing, only to provide poor services.	This kind of attack is dealt with by observing the behavior of the nodes and eliminating those with inconsistent trust values over a certain amount of time.
On-Off Attack (OOA)	This is an attack where a node frequently shifts between good and bad behavior to avoid being labelled.	The same solution to OSA can be applied to this as well.

PERFORMANCE AND COMPARATIVE ANALYSIS OF PROPOSED SCHEME

The proposed system is tried and tested in a simulation environment. The details of simulation parameters, evaluation and performance, and the comparative analysis of the proposed framework is discussed in this section.

5.1 Simulation Setup

A simulation is created in Netlogo v 6.2.0 to evaluate the proposed trust management system for Fog computing. Using Netlogo, we have created a large network (500 nodes), a medium size network (100 nodes) and a small network (50 nodes). The list of simulation parameters and their respective values are given in table ??.

The simulation setup is shown in figure 5.1, the left window pane consists of all the controls and sliders that determine the number of nodes, number of links each node will create, conduct change of a node from good to bad, node analysis and a csv output file. The right window pane shows the experiment world, in this case a network of 50 nodes can be observed. The function of highlighted nodes are as follows, the red and blue nodes are SP and SR respectively, and the green node is their common neighbor. The simulations are carried out for 50, 100 and 500 nodes respectively. In our setup any node can be a service requestor or service provider, each node will take recommendations from its 1-hop neighbors only to build reputation rating of the trustee node. The trust update is event-driven, simulation runs for 25 iterations during which time the trust values are calculated and updated accordingly.

5.2 Evaluation and Performance of SQT framework

This section will discuss the evaluation and performance of the SQT framework, it will consider two basic scenarios. The experiment was conducted to analyze the behavior of a good node as well as a bad node in the network. The simulation is carried out for 25 iterations for 100 and 500 nodes respectively.

Figure 5.2 shows the performance of a random good node in the proposed framework

Setup

Red is the Selected Node, Blue wants to exchange packets with Red and Green is their Common neighbour

no-of-nodes 50 create-links 4

GC_to_BC? apply_conduct_change_after_iteration 15

Run

single_node_analysis (for 25 iteration)

Run After 3 nodes are selected. Else call Step until three nodes are selected.

Output

file_name
test-good-to-bad-node-record.csv

write-to-file

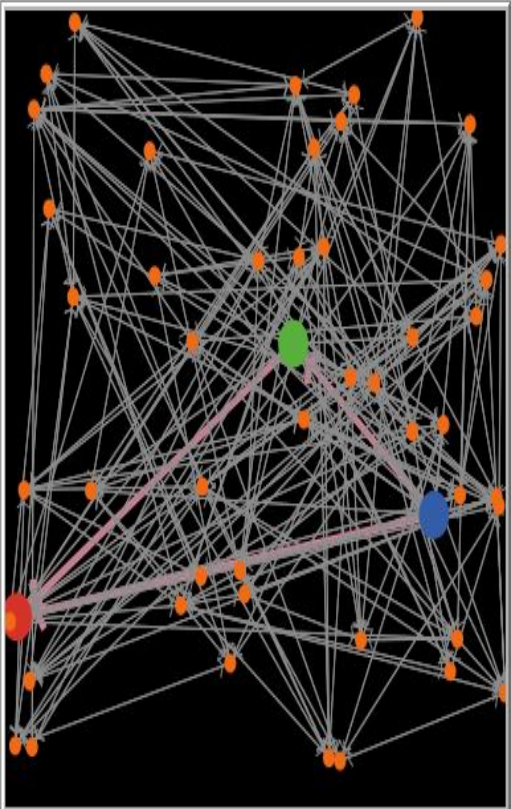


Figure 5.1: Initial setup of the simulation. The highlighted nodes are 1-hop neighbors, both blue and red highlighted nodes are calculating trust of one another as shown in Command Center

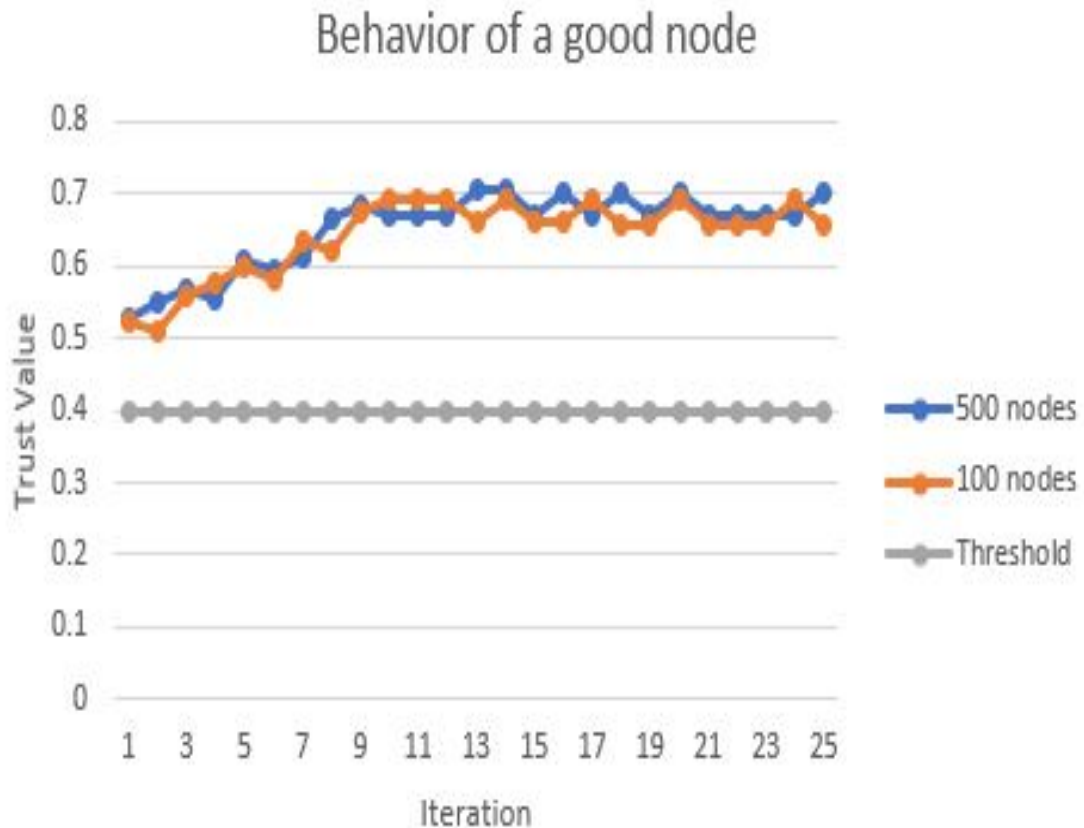


Figure 5.2: Trust value of a randomly selected good fog node in a network of 100 and 500 under normal circumstances, it exhibits the accuracy and convergence of the trust value in networks with 100 and 500 nodes. The trust value converges quickly with more accuracy when more weight is given to the direct observations. Whereas trust value converges later in the computation cycle when more weight is assigned to indirect observations.

The SQT trust model is simulated in Netlogo v 6, the code of which is available on github (https://github.com/MH9196/FogTrustModel/blob/main/directed_graph_network_with_single_node.netlogo). The performance analysis of a randomly selected bad node is shown in figure 5.3. The algorithm is designed to penalize a node on bad service, this makes it twice as hard for the node to recover its reputation. Each node maintains a rank which decreases whenever it behaves undesirably, if the rank of a node becomes zero it is eventually kicked out of the network.

The resilience of our proposed is tested against the on off selective forwarding attack. The results in figure 5.4 clearly depict the accuracy of malicious node detection. It can be seen that as soon as the node changed its behavior, the system detected it and the trust value went

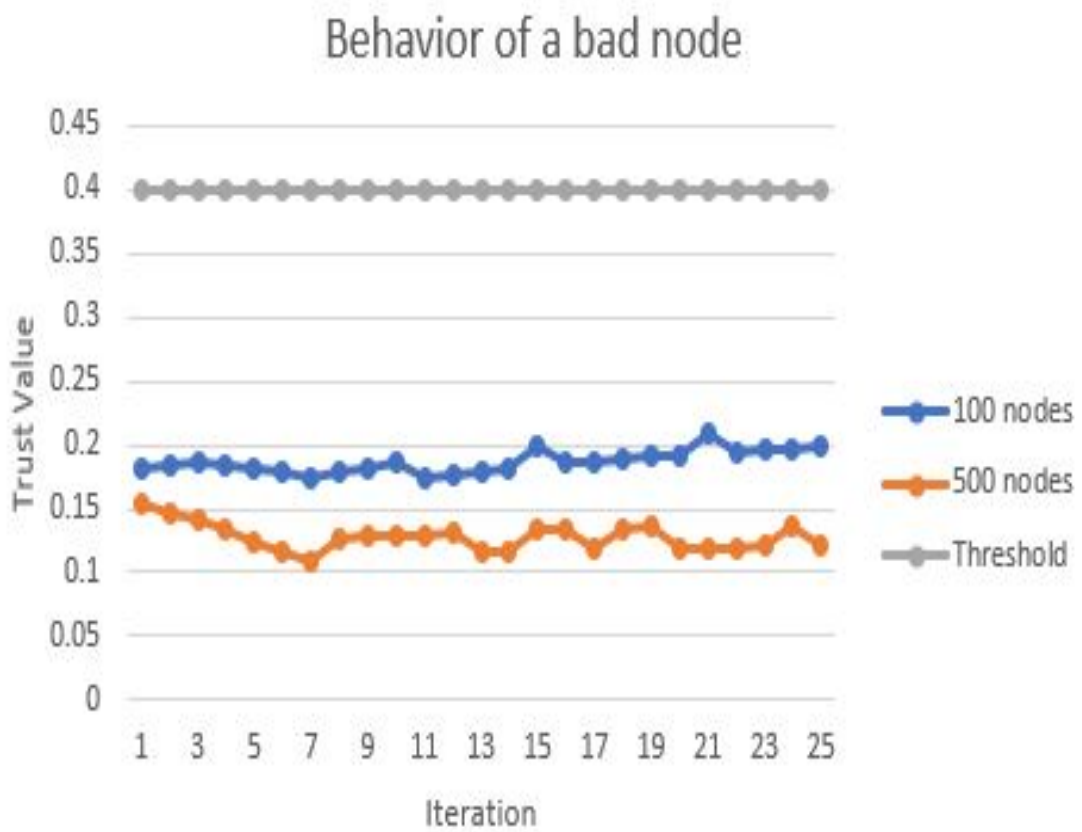


Figure 5.3: Trust value of a randomly selected bad fog node in a network of 100 and 500. The trust value remains below the threshold which makes it hard for a bad node to carry out attacks.

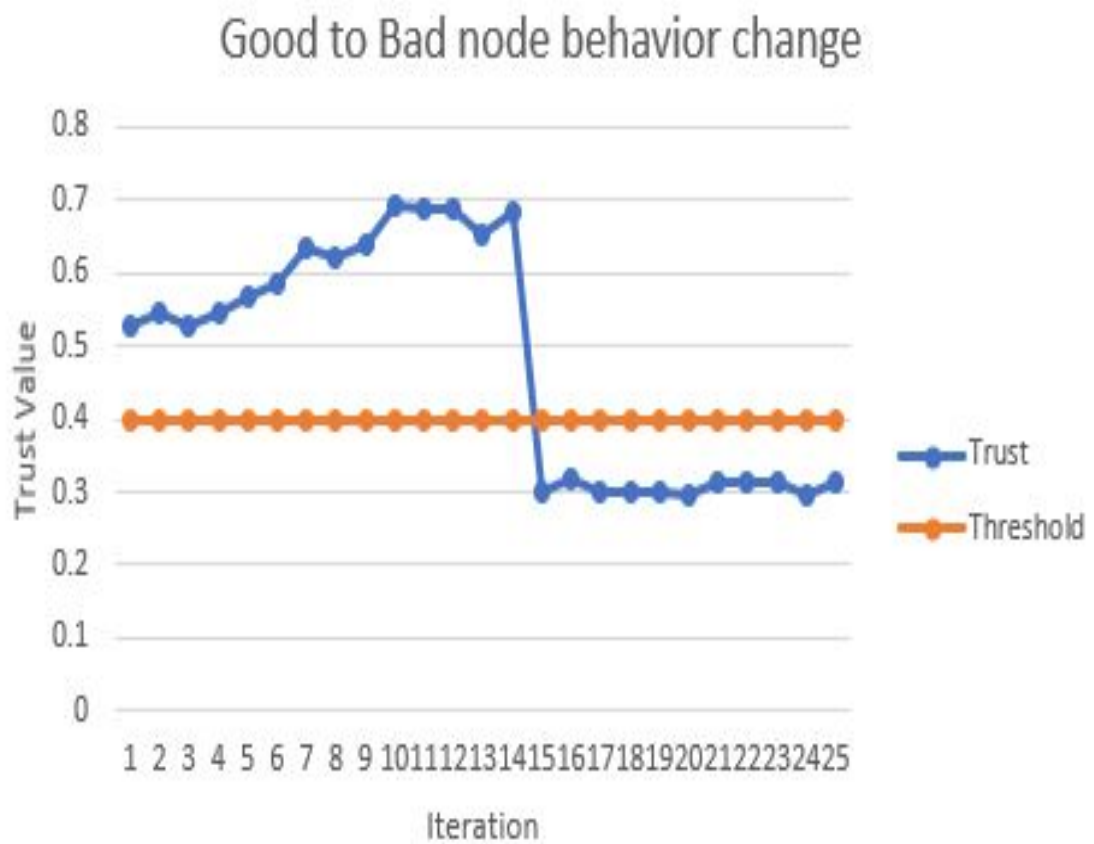


Figure 5.4: The performance of the proposed framework in on off selective forwarding attack.

below the set threshold. It can be observed that the trust value below threshold do not show a considerable increase with each upcoming iteration.

5.3 Comparative Analysis

Due to limited work available on the subject, we carried out the comparative analysis with the model introduced by A.M. Kowshalya et al. [29] and TMCoi-SIoT [39] with the proposed SQT model against on off attack. All the models can detect on off attack, but the proposed SQT model can not only identify but make it hard for the malicious node to recover its reputation as shown in figure 5.5. An in depth comparison of the models is given in table 5.1. The proposed SQT model is a two-way trust approach, it requires both communicating nodes to validate each other before connecting. Moreover, SQT is also resilient against other attacks such as:

1. Self-promotion attack(SPA) as it does not allow any node to self-recommend
2. Bad-mouthing attack(BMA) as it only considers recommendations from trusted neighbors
3. Ballot-stuffing attack(BSA) due to weighted recommendations
4. Opportunistic service attack(OSA) as it eliminates nodes with inconsistent behavior over time

Table 5.1: Comparison of SIoT trust model with the proposed SQT model

Contributuion	Kowshalya	TMCoi-SIoT	SQT
Two-way trust	x	x	✓
Distributed approach	x	x	✓
Resilient against on off attack	✓	✓	✓
Resilient against other trust-based attacks (SPA, BSA, BMA, OSA)	x	x	✓
Low Computation Cost	✓	✓	✓

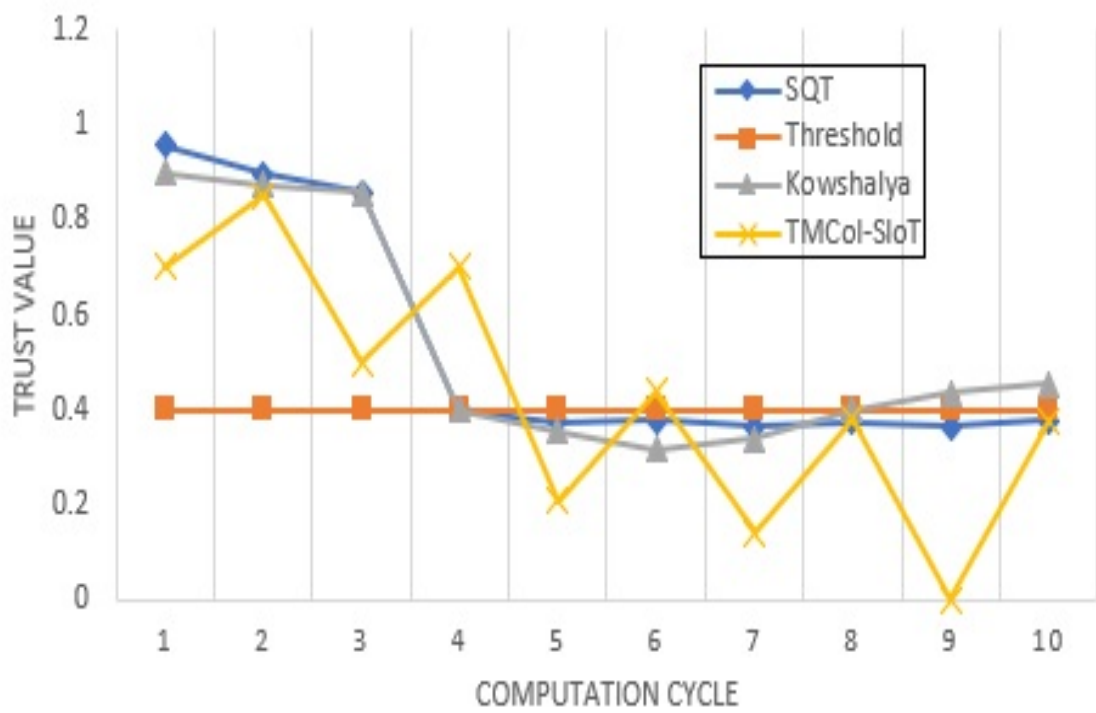


Figure 5.5: Comparison of SQT system model with two existing SIoT models, Kowshalya and TMCoi-SIoT in the presence of on off selective forwarding attack.

CONCLUSION AND FUTURE DIRECTION

It has been made evident that one of the major security and privacy issues faced by fog today is the lack of a trust management framework. The existing trust management models cannot be implemented to fog but extensive literature review has led to the deduction that fog network is much more like SIoT and MANETs, hence, the basis for building SQT management framework was derived from them. In this paper, a peer-to-peer trust management system is proposed that enable the fog nodes to develop a level of trust before connecting with others. The system prevents fog nodes from establishing communication with untrustworthy nodes. Bayes model forms the mathematical basis of our proposed solution, it can calculate the trust value of a node and predict its future behavior. The extensive evaluation shows that the system is resilient to trust-based network attacks, it converges quickly and expels the malicious nodes over a period of time.

BIBLIOGRAPHY

- [1] M. Chiang and T. Zhang, “Fog and IoT: An overview of research opportunities,” *IEEE Internet things J.*, vol. 3, no. 6, pp. 854–864, 2016.
- [2] M. Yannuzzi, R. Milito, R. Serral-Gracià, D. Montero, and M. Nemirovsky, “Key ingredients in an IoT recipe: Fog Computing, Cloud computing, and more Fog Computing,” in *2014 IEEE 19th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, 2014, pp. 325–329.
- [3] Thomas Barnett, Jr, “Cisco Global Cloud Index 2015–2020”, Accessed: 6-1-2020, from https://www.cisco.com/c/dam/m/en_us/services-provider/ciscoknowledgenetwork/files/622_11_15-16-Cisco_GCI_CKN_2015-2020_AMER_EMEAR_NOV2016.pdf
- [4] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, “Edge computing: Vision and challenges,” *IEEE internet things J.*, vol. 3, no. 5, pp. 637–646, 2016.
- [5] Lopez Research LLC, “An Introduction to Internet of Things (IoT) November 2013”, Accessed: 31-3-2021, from https://www.cisco.com/c/dam/en_us/solutions/trends/iot/introduction_to_IoT_november.pdf
- [6] L. Atzori, A. Iera, G. Morabito, and M. Nitti, “The social internet of things (sIoT)—when social networks meet the internet of things: Concept, architecture and network characterization,” *Comput. networks*, vol. 56, no. 16, pp. 3594–3608, 2012.
- [7] A. Rayes and S. Salam, “Internet of things from hype to reality,” Springer, 2017.
- [8] E. Marín-Tordera, X. Masip-Bruin, J. García-Almiñana, A. Jukan, G.-J. Ren, and J. Zhu, “Do we all really know what a fog node is? Current trends towards an open definition,” *Comput. Commun.*, vol. 109, pp. 117–130, 2017.
- [9] D. Alsen, M. Patel, and J. Shangkuan, “The future of connectivity: enabling the Internet of Things,” McKinsey Company. Available online at <https://www.mckinsey.com/featured-insights/internet-of-things/our-insights/the-future-of-connectivity-enabling-the-internet-of-things>, checked on, vol. 6, no. 20, p. 2019, 2017.
- [10] M. Mukherjee et al., “Security and privacy in fog computing: Challenges,” *IEEE Access*, vol. 5, pp. 19293–19304, 2017.
- [11] N. Mäkitalo, A. Ometov, J. Kannisto, S. Andreev, Y. Koucheryavy, and T. Mikkonen, “Safe, secure executions at the network edge: coordinating cloud, edge, and fog computing,” *IEEE Softw.*, vol. 35, no. 1, pp. 30–37, 2017.
- [12] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, “Fog computing and its role in the internet of things,” in *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*, 2012, pp. 13–16.
- [13] A. Alrawais, A. Alhothaily, C. Hu, and X. Cheng, “Fog computing for the internet of things: Security and privacy issues,” *IEEE Internet Comput.*, vol. 21, no. 2, pp. 34–42, 2017.

- [14] O. C. A. W. Group, "OpenFog reference architecture for fog computing," OPFRA001, vol. 20817, p. 162, 2017.
- [15] E. Alemneh, S.-M. Senouci, and P. Brunet, "PV-Alert: A fog-based architecture for safeguarding vulnerable road users," in 2017 Global Information Infrastructure and Networking Symposium (GIIS), 2017, pp. 9–15.
- [16] A. M. Rahmani et al., "Exploiting smart e-Health gateways at the edge of healthcare Internet-of-Things: A fog computing approach," *Futur. Gener. Comput. Syst.*, vol. 78, pp. 641–658, 2018.
- [17] EdgeX Foundry Retrieved April 16, 2021, from website: <https://www.edgexfoundry.org/>
- [18] AWS greengrass Retrieved April 16, 2021, from website: <https://aws.amazon.com/greengrass/>
- [19] Azure IoT Edge— Microsoft Retrieved 16 April 2021, from website: <https://azure.microsoft.com/en-us/services/iot-edge/>
- [20] J. Guo, R. Chen, and J. J. P. Tsai, "A survey of trust computation models for service management in internet of things systems," *Comput. Commun.*, vol. 97, pp. 1–14, 2017.
- [21] J.-H. Cho, A. Swami, and R. Chen, "A survey on trust management for mobile ad hoc networks," *IEEE Commun. Surv. tutorials*, vol. 13, no. 4, pp. 562–583, 2010
- [22] D. H. McKnight and N. L. Chervany, "What trust means in e-commerce customer relationships: An interdisciplinary conceptual typology," *Int. J. Electron. Commer.*, vol. 6, no. 2, pp. 35–59, 2001.
- [23] T. S. Dybedokken, "Trust management in fog computing." NTNU, 2017.
- [24] A. Josang and R. Ismail, "The beta reputation system," in Proceedings of the 15th bled electronic commerce conference, 2002, vol. 5, pp. 2502–2511
- [25] S. Ganeriwal, L. K. Balzano, and M. B. Srivastava, "Reputation-based framework for high integrity sensor networks," *ACM Trans. Sens. Networks*, vol. 4, no. 3, pp. 1–37, 2008.
- [26] A. Jøsang and S. J. Knapskog, "A metric for trusted systems," 1998.
- [27] M. Panda and A. Abraham, "Development of a reliable trust management model in social internet of things," *Int. J. Trust Manag. Comput. Commun.*, vol. 2, no. 3, pp. 229–258, 2014.
- [28] D. Chen, G. Chang, D. Sun, J. Li, J. Jia, and X. Wang, "TRM-IoT: A trust management model based on fuzzy reputation for internet of things," *Comput. Sci. Inf. Syst.*, vol. 8, no. 4, pp. 1207–1228, 2011.
- [29] A. M. Kowshalya and M. L. Valarmathi, "Trust management for reliable decision making among social objects in the Social Internet of Things," *IET Networks*, vol. 6, no. 4, pp. 75–80, 2017.

- [30] H. Xiao, N. Sidhu, and B. Christianson, “Guarantor and reputation based trust model for social internet of things,” in 2015 international wireless communications and mobile computing conference (IWCMC), 2015, pp. 600–605.
- [31] P. Martinez-Julia and A. F. Skarmeta, “Beyond the separation of identifier and locator: Building an identity-based overlay network architecture for the Future Internet,” *Comput. Networks*, vol. 57, no. 10, pp. 2280–2300, 2013.
- [32] R. Chen, J. Guo, and F. Bao, “Trust management for SOA-based IoT and its application to service composition,” *IEEE Trans. Serv. Comput.*, vol. 9, no. 3, pp. 482–495, 2014.
- [33] B. Yu and M. P. Singh, “An evidential model of distributed reputation management,” in *Proceedings of the first international joint conference on Autonomous Agents and Multiagent Systems: Part 1*, 2002, pp. 294–301.
- [34] A. Jøsang, R. Ismail, and C. Boyd, “A survey of trust and reputation systems for online service provision,” *Decis. Support Syst.*, vol. 43, no. 2, pp. 618–644, 2007.
- [35] S. Ries, S. M. Habib, M. Mühlhäuser, and V. Varadharajan, “Certainlogic: A logic for modeling trust and uncertainty,” in *International conference on trust and trustworthy computing*, 2011, pp. 254–261.
- [36] S. Ries, “Certain trust: a trust model for users and agents,” in *Proceedings of the 2007 ACM symposium on Applied computing*, 2007, pp. 1599–1604.
- [37] Y. Wang, Y.-C. Lu, I.-R. Chen, J.-H. Cho, A. Swami, and C.-T. Lu, “Logittrust: A logit regression-based trust model for mobile ad hoc networks,” in *6th ASE International Conference on Privacy, Security, Risk and Trust*, Boston, MA, 2014, pp. 1–10.
- [38] F. Bao, R. Chen, and J. Guo, “Scalable, adaptive and survivable trust management for community of interest based internet of things systems,” in *2013 IEEE eleventh international symposium on autonomous decentralized systems (ISADS)*, 2013, pp. 1–7.
- [39] O. Ben Abderrahim, M. H. Elhdhili, and L. Saidane, “TMCoi-SIoT: A trust management system based on communities of interest for the social Internet of Things,” in *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*, 2017, pp. 747–752.
- [40] O. Ben Abderrahim, M. H. Elhdhili, and L. Saidane, “CTMS-SIoT: A context-based trust management system for the social Internet of Things,” in *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*, 2017, pp. 1903–1908.
- [41] N. B. Truong, T.-W. Um, and G. M. Lee, “A reputation and knowledge based trust service platform for trustworthy social internet of things,” *Innov. clouds, internet networks (ICIN)*, Paris, Fr., 2016.
- [42] N. B. Truong, T.-W. Um, B. Zhou, and G. M. Lee, “From personal experience to global reputation for trust evaluation in the social internet of things,” in *GLOBECOM 2017-2017 IEEE Global Communications Conference*, 2017, pp. 1–7.

- [43] S. A. Soleymani et al., “A secure trust model based on fuzzy logic in vehicular ad hoc networks with fog computing,” *IEEE Access*, vol. 5, pp. 15619–15629, 2017.
- [44] T. Wang et al., “Fog-based evaluation approach for trustworthy communication in sensor-cloud system,” *IEEE Commun. Lett.*, vol. 21, no. 11, pp. 2532–2535, 2017.
- [45] T. Wang, G. Zhang, M. Z. A. Bhuiyan, A. Liu, W. Jia, and M. Xie, “A novel trust mechanism based on fog computing in sensor–cloud system,” *Futur. Gener. Comput. Syst.*, 2018.
- [46] T. Wang et al., “When sensor-cloud meets mobile edge computing,” *Sensors*, vol. 19, no. 23, p. 5324, 2019.
- [47] J. Yuan and X. Li, “A reliable and lightweight trust computing mechanism for IoT edge devices based on multi-source feedback information fusion,” *Ieee Access*, vol. 6, pp. 23626–23638, 2018.
- [48] F. H. Rahman, T.-W. Au, S. H. S. Newaz, W. S. Suhaili, and G. M. Lee, “Find my trustworthy fogs: A fuzzy-based trust evaluation framework,” *Futur. Gener. Comput. Syst.*, vol. 109, pp. 562–572, 2020.
- [49] Z. Hao, E. Novak, S. Yi, and Q. Li, “Challenges and software architecture for fog computing,” *IEEE Internet Comput.*, vol. 21, no. 2, pp. 44–53, 2017.
- [50] P. B. Velloso, R. P. Laufer, D. de O. Cunha, O. C. M. B. Duarte, and G. Pujolle, “Trust management in mobile ad hoc networks using a scalable maturity-based model,” *IEEE Trans. Netw. Serv. Manag.*, vol. 7, no. 3, pp. 172–185, 2010.