Adaptive Trust Calculation in Fog Computing

By

00000328179

Alishba Nawaz

Supervisor: Dr. Mian Muhammad Waseem Iqbal

A thesis submitted to the faculty of Computer Software Engineering 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 Software Engineering.

April 2023

THESIS ACCEPTANCE CERTIFICATE

Certified that final copy of MS/MPhil thesis written by Ms. **Alishba Nawaz,** Registration No. **00000328179**, of **Military College of Signals** has been vetted by undersigned, found complete in all respect as per NUST Statutes/Regulations, is free of plagiarism, errors and mistakes and is accepted as partial, fulfillment for award of MS/MPhil degree. It is further certified that necessary amendments as pointed out by GEC members of the student have been also incorporated in the said thesis.

Signature: _____

Supervisor: Assoc Prof Dr. Waseem Iqbal

Date: _____

Signature (HoD): _____

Date: _____

Signature (Dean/Principal): _____

Date: _____

Abstract

Fog is well suited for situations where a huge number of decentralized devices must communicate, provide live analysis of data, and perform storage jobs because of its inherent decentralized nature and capacity to process data in transit, i.e., ability to draw conclusions in real-time. Fog computing offers the dependability that time-sensitive smart healthcare systems require because of its ability to operate near the end user and independence from centralized architecture. Because healthcare data is so vital, there is a need for stronger security and privacy solutions for fog computing, where trust is crucial The goal of this research is to provide a context-based adaptive trust solution for the smart healthcare environment using Bayesian technique and similarity measures against bad mouthing and ballot stuffing since context dependent trust solution for fogs is still an open research topic. To assess our findings, the proposed trust model has been simulated in Contiki and Cooja. In contrast to static weighting, adaptive weights assigned to direct and indirect trust using entropy values assure the least amount of trust bias, and calculations of context similarity remove recommender nodes with malevolent intent utilizing server, coworker, and service similarity. Due to its minimal trust computation overhead and linear complexity O(n), this model is effective.

Dedication

I dedicate this thesis to myself, for achieving what once seemed impossible.

ACKNOWLEDGEMENTS

I am grateful to God Almighty for bestowing upon me the strength and zeal to complete this thesis, and I am grateful to Him for His mercy and generosity, without which I could not have completed this research. I'd want to thank everyone in my family, especially my father, Malik Muhammad Nawaz, for his unwavering support and for being my rock throughout these arduous scholastic years.

Finally, I would like to express my gratitude to my supervisor, Assoc. Prof. Dr. Mian Muhammad Waseem Iqbal, for his persistent guidance and great assistance in making my thesis a reality.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

Internet of Things	IoT
Mobile Ad-Hoc Networks	MANET
Quality of Protection	QoP
Quality of Service	QoS
Social Internet of Things	SIoT
Trust Management Systems	TMS
Service Provider	SP
Service Requester	SR
Trust Manager	TM
Self-Promotion Attack	SPA
Bad Mouthing Attack	BMA
Opportunistic Service Attack	OSA
Ballot Stuffing Attack	BSA
Service-oriented IoT Systems	SoA-IoT
Community of Interest	COI
Packet Delivery Ratio	PDR
Communication Overhead	CO
Co-Work	Cox
Server Similarity	SS

Chapter 1

Introduction

1.1 Overview

The bulk of data generated by IoT devices and sensors is expanding dramatically, putting a strain on the nodes' ability to gather, keep, evaluate, and execute it.

World trends are shifting away from centralized clouds and towards decentralized edge devices in an effort to increase productivity over cost reduction. Since its creation, the idea of offloading has grown in popularity since it enables a device to transfer some of its responsibilities to a device with greater computational capability, resulting in a more streamlined functioning. In a similar vein, IoT devices transfer computationally demanding activities to other systems in order to improve their performance and bandwidth as well as to reduce the resulting latency problems. The network edge is the location on a network where end users connect to the main network. At the same time, data is being processed. The idea of processing data on edge devices has reduced the need for central nodes to perform routine operations and instead encourages processing of data close to its source. Platform of cloud was established to manage these details, but it introduced worries about bandwidth and latency, particularly for time-critical systems. Fog computing, which involves moving processing power far from the central cloud system and closer to the end user, has arisen in recent years as a solution to this problem. [7]. An edge network closer "mini-cloud" is what is meant by a fog node. [10]. Anything near the end user can serve as a fog node, such as routers or switches or servers. However, before it can be widely adopted, it must overcome numerous problems, the most important of which are security and privacy. There are numerous techniques for establishing trust between service provider and requestor nodes as well as for secure data transport and authentication.[5][12][3].

The form that is most basic, trust is the belief that a given item will behave in accordance with the QoS and security regulations. In a digital setting, trust is essential for assisting new colleagues. A device can be classified as secure or insecure depending on the trust levels of individual nodes and the network policies for the minimum trust threshold. A trust management system aids in building trust amongst network nodes to ensure efficient operation. The degree of assurance that an IoT device will act in a specific way is defined

as trust [15]. Each device in the network of fog needs to comprise a specific trust level for them to engage and communicate with one another. [3]. Context is crucial in fog computing and trust calculation; for instance, one fog node might be relied upon to complete a task for a fog client but useless for another. Smart healthcare is one of the expanding use cases for fog computing [1]. Fog computing has been and can be used in fields such as remote monitoring, automated patient surveillance, and smart healthcare devices [14][9]. More security and confidentiality preventive measures are required as a result of the connectedness of devices as a result of the ongoing surveillance and the delicacy of patient data.

1.2 Motivation:

Fog computing has the reliability that time-sensitive smart medical systems require because to its capability to operate near to the user and its independence from centralized structure. Managing such a substantial volume of data and sensitive patient data poses questions about security, privacy, and trust. In the literature, there aren't many models for fog applications in healthcare, proposals for system architecture, or encryption [2] [11], but there is minimal [20] to no research on trust, particularly context-based dynamic trust solutions. We have discovered a sizable research gap, especially when determining the security level of a node or device from which you are dispensing or receiving services necessitates trust as a key evaluation factor. This is due to the dynamic nature of trust and the importance of the data generated by smart healthcare systems. In order to address these assaults, we also aim to propose a adaptive context-based trust system for time-critical smart medical systems.

1.3 Scope and Objectives

The primary goals of this research are to investigate existing trust schemes in the Internet of Things (IoT), SIoT (Social Internet of Things), and Fogs, as well as to identify how SIoT trust solutions can be applied in fogs and to examine existing fog-based solutions in healthcare. The primary goal of this project will be to provide a context aware trust management solution for fog computing that can be applied for time-sensitive smart healthcare systems while minimizing the risk of attacks and increase security and privacy.

1.4 Contributions

The research aims to achieve the following goals:

- To study the current trust mechanism in IOT.
- Using a Bayesian methodology and similarity metrics, For fog-based IoT systems, we suggested a context-aware trust solution.
- To propose trust mechanism to overcome ballot stuffing and bad mouthing in medical IOT system.
- Comparative analysis of proposed and existing schemes in the field.
- Utilizing entropy theory, an adaptive system for allocating weights to computations of both direct and indirect trust.
- The simulation of our model be done on Contiki, Cooja and showed the difference between employing static and adaptive weighting techniques.

1.5 Thesis Outline

This thesis is divided into six chapters:

- Chapter 1: This chapter contains introduction, scope, objectives and the contributions we have made in this thesis research.

- Chapter 2: In this chapter, we briefly discuss IoT, SIoT and also fog computing in detail.

- Chapter 3: Deals with the topic's chosen literature review, previous work, and comparisons.

- Chapter 4: This chapter discusses the proposed model in detail, including mathematical modelling.

- Chapter 5: This chapter delves into the practical simulation, analysis, and assessment of our proposed model utilizing a healthcare use case.

- Chapter 6: This chapter concludes the research and makes recommendations for further work.

Chapter 2

PRELIMINARIES

This chapter will provide an overview of IoT, SIoT, and Fog computing, as well as its applications and architecture.

2.1 Internet of Things:

The Internet of Things (IoT) refers to a massive network of interconnected objects. Items can be anything having sensors, software, or any other technology that connects to and exchanges data with other objects via a network [13]. In other words, Sensors, the data they gather, and the processing, storing, and analysis of that data make up the Internet of Things (IoT). It permits us to access diverse applications from wherever, at any time. This network of networked things generates a vast amount of data, which enables IoT-related apps to make informed decisions and improve their services. IoT applications include smart automobiles, smart homes, smart healthcare and smart cities

2.2 Social Internet of Things

Many researchers have been drawn to the integration of social ties with IoT systems in recent years. IoT objects emulate human social behavior in SIOT. The goal of incorporating social relations in IoT is to simplify the process of establishing interactions among items and produce better results through collective effort. A social network of items working together to complete a job outperforms an individual working alone.

As the world moves toward the Internet of Everything, which will entail everything (people, processes, things, data) collaborating and interacting, as well as producing a vast quantity of data, it is vital to give the system some autonomy, which is where SIOT comes in. Decisions can be made by SIOT systems without the need for human intervention. Each

device/object can develop relationships with others and collaborate based on the rules established by device owners.

2.3 Current Issues in IoT/SIOT

IoT devices and social interactions are producing exponentially more data, placing a strain on nodes' capacity to simultaneously collect, analyze, store, and process data. According to International Data Corporation (IDC), data created by connected IoT devices would total 73.1ZB by 2025 [14]. Cloud computing was developed to manage the ever-increasing data produced by trillions of IoT devices. Because the cloud provides the required storage and processing capabilities to resource restricted IoT devices, cloud and IoT complement each other in terms of capabilities. Data is collected, processed, analyzed, and stored in the cloud. However, it has bandwidth, dependability, and latency difficulties, which are especially problematic for time-critical systems that require real-time processing or quick results. These issues limit IoT uses. To address these concerns, Cisco developed the notion of fog computing [6], It involves moving processing power away from the centralized cloud infrastructure and towards the end user. Because fogs, unlike clouds, are distributed, this solves the bandwidth and latency difficulties while also increasing reliability and reducing reliance on a single central cloud system.

2.4 Overview of Fog Computing

In contrast to the summary provided in the previous chapter, this section will go into considerable detail into the concept, operation, and architecture of fog computing. Part 2.4.1 provides a definition and some of its major properties; section 2.4.2 goes into considerable detail about the fog reference architecture. The uses of fog in contemporary architecture are discussed in Sections 2.4.3, 2.4.4, and 2.5, which are followed by the research strategy used to finish this study and its rationale.

2.4.1 Fog Computing

So far, we have relied on the cloud for IoT data management, but due to the rapid expansion of devices of IoT and henceforth data, it is difficult to rely on a single entity for processing of this massive volume of data. The fog computing paradigm was developed for this reason. Fog is a cloud extension that moves storage, communication, compute, and other networking functions to the network's edge. Data in fog computing is examined and kept by fog nodes located near the source of the data. Cloud-related issues such as latency and bandwidth are addressed with fog computing. Because data is processed closer to edge devices, the time required to respond to a service request is reduced, which solves the cloud's time-sensitive request processing limitation [16], less bandwidth is required because not all data is sent to the cloud, and there is no single point failure due to the distributed nature of fog nodes. A "mini-cloud" that is nearer to the edge devices is a fog node. [10].

The Open Fog Consortium was created in November 2015 to encourage the use of fog computing in a variety of industries. Among its original members are major technology academic institutions and companies 5 like Dell, Cisco, Microsoft, Dell, Princeton University, and Hitachi, among many others [17]. It was amalgamated with the Industrial Internet Consortium in January 2019, and is now known as the Industry IoT Consortium [18]. The computing of fog paradigm is promising for the future of IoT, but it has several difficulties before widespread adoption, including privacy, security, architecture definition, and so on. Nonetheless, it is a platform to watch.

2.4.2 Fog Architecture:

According to a survey [19], several fog computing architectures exist in the literature depending on the application demand and deployment circumstances. The majority of these architectures are extensions of the basic three-tiered fog service design. The Open Fog Consortium [25] has tried to design a generic fog architecture as a starting point for developing an industry standard for fog computing. Fog is organized in a multi-layered hierarchical structure. As shown in Fig. 2.1, Fog's layers lie between the cloud and the IoT layer, operating as an intermediary body between them. The number of layers may vary depending on the application.

Computation can be done at any level, but computational intelligence reduces as the level drops. Higher-level nodes will have a better view of the network.

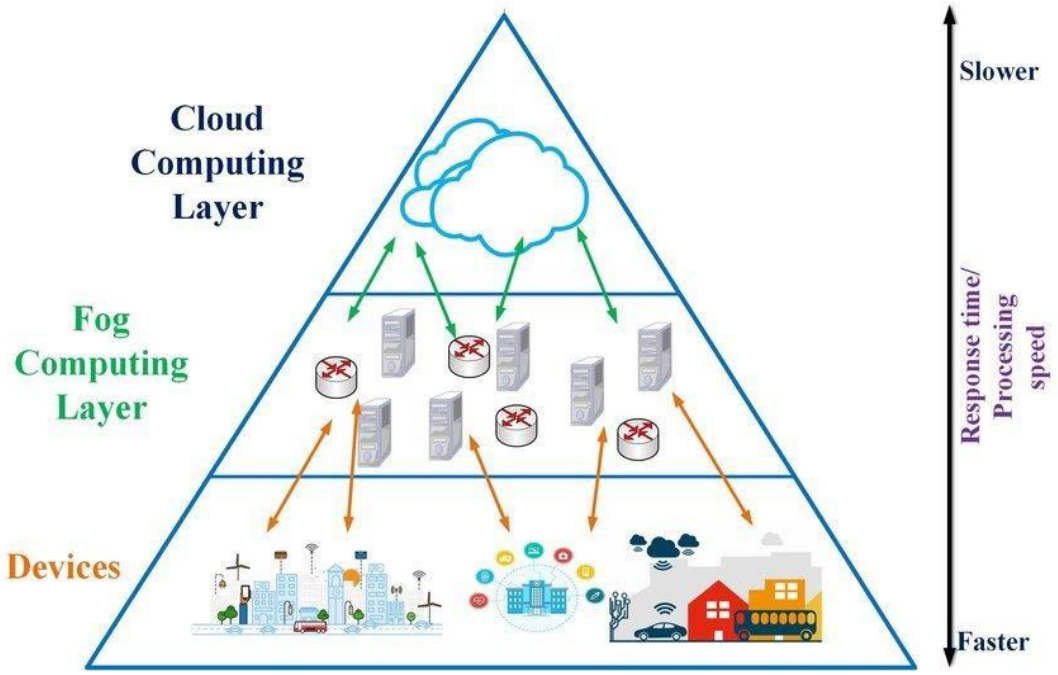

Figure 2.1 Fog Architecture from faster processing with

quick response time towards slower processing

The Internet of Things (IoT) layer is the lower layer, which is where data is created and sent to edge nodes for additional processing. Data processing and computation take place at the fog layer, which is composed of several decentralized nodes. Nodes located at the same layer can share resources, perform fault tolerance, and offload data depending on the workload and are presumed to have similar intelligence levels. The presence of a fog layer between the Internet of Things (IoT) and the cloud reduces service delays and boosts Quality of Service (QoS) by speeding up response times. With the highest computational intelligence, cloud resides at the top of the hierarchy and handles services that call for complicated calculations or data that fog cannot manage, such as the processing of large data.

Table 2.1 Fog Deployment Model

Deployment Model	Description
Public Fog Model	Similar to public cloud, public fog offers services to the public where users can rent the essential services they need. A single organization owns the public infrastructure in the case of clouds, but this shouldn't be the case in the case of fog. Collaboration between several organizations is required to develop this deployment approach.
Private Fog Model	Private cloud services are more expensive than public cloud services and are offered to high-end security organizations. Private fog models may be owned by a single entity, which may also offer services to third parties as required.
Community Fog Model	This model lowers the cost of deployment by recommending a community-based system wherein two or more businesses can work together to provide services at a lower cost than private fog but with greater security than public fog.

Hybrid Fog Model	Hybrid fog, as its name implies, is the outcome of the union of public and private fog. Sensitive information can be transmitted to private fog and non-sensitive information to public fog under this paradigm, making it suitable for applications with varying levels of security.

2.4.3 Uses and Advantages of Fog Computing

Fog is well suited for situations where communication is required between a large number of decentralized devices, carry out processing, and store data due to its innate decentralization and ability to process data while it is being transmitted, i.e., its potential to make decisions in real-time. Fog has a broad range of applications because of this, including time-sensitive IoT applications for smart homes [10], smart medical [1], smart grids, smart vehicle systems [20], and more [31] [10].

Many major software companies, such as Microsoft, the Linux Foundation, and Amazon Web Services, have invested in research and development of it due to its simplicity and flexibility [22]. Fog has many advantages because of its diverse nature, including but not limited to location awareness, decentralization, agility support, scalability, run-time interaction, bandwidth and storage conservation, low energy consumption, increased dependability, reliability, and possibly increased privacy due to processing close to the edge [16].

2.4.4 Direction of Fog Research

Given the volume of data that IoT devices have produced, the attack surface in fog computing is very large. For fog computing, improved privacy and security solutions are required, and trust is crucial [23]. Trust is the level of assurance that a thing or Internet of

Things device will function as intended [6]. Every device in the fog network needs to have some level of trust in order for them to cooperate and communicate with one another . For other platforms like IoT, SIOT, and Cloud, there has been a lot of study on trust management schemes [24], but there hasn't been as much done on trust management schemes for Fog as a secure platform, particularly context-aware solutions. Current research focuses primarily on new encryption schemes and protocols, as well as a few new trust models, but does not take context into account. Similar to other domains, context is crucial in trust computing. For example, you might trust a buddy with your money but probably not with a personal secret, and vice versa.

2.5 Motivation

Fog networks, also known as fog nodes, are frequently large-scale networks made up of many network objects (i.e., any device with enough processing power and memory). Attacks are more likely because these nodes are required to connect to one another for various transactions. To protect the connection from attacks like ballot-stuffing and badmouthing, the proposed two-way trust management technique should confirm that both nodes have formed a trustworthy connection before the transaction.

This study seeks to develop a dynamic context-based trust solution for time-sensitive smart healthcare system use cases. Fog computing offers the dependability that time-sensitive smart healthcare systems require because to its capability to operate near to the user and independence from central structure. Context-based trust in fogs will secure data privacy and eliminate the possibility of malevolent nodes tampering like bad-mouthing and ballot stuffing attacks with the network and important data.

Chapter 3

LITERATURE REVIEW

This chapter goes into great length about literary reviews. It outlines the research that has been done thus far on IoT, SIOT, and fog computing trust.

3.1 Trust in IoT

The security issues that heterogeneous IoT objects face and potential solutions have been the subject of extensive investigation. Trust is one of the main factors to consider in order to connect IoT items securely. Among the essential security needs for an IoT environment are the fundamental CIA triad (confidentiality, integrity, availability), access control, privacy, trustworthiness, authorization and auditing, according to a survey [25].

The degree to which an IoT device will behave exactly as it was designed to behave is known as the level of trust [15]. In an IT environment, a set of protocols and procedures specify how it should behave; if it succeeds in doing so, only then will it be referred to as "trustworthy." As depicted in Fig. 3, A planned interaction between two entities is trust. The entity that needs the trustee's services is known as the 1 trustor, and it needs to have a certain amount of faith that the trustee will act in accordance with its intentions. Trust enables this confidence. However, due to the context and environmental aspects that are constantly changing, trust must also be updated periodically. Because trust makes it possible for entities to work together safely, it is crucial for distributed networks like the Internet of Things IOT.

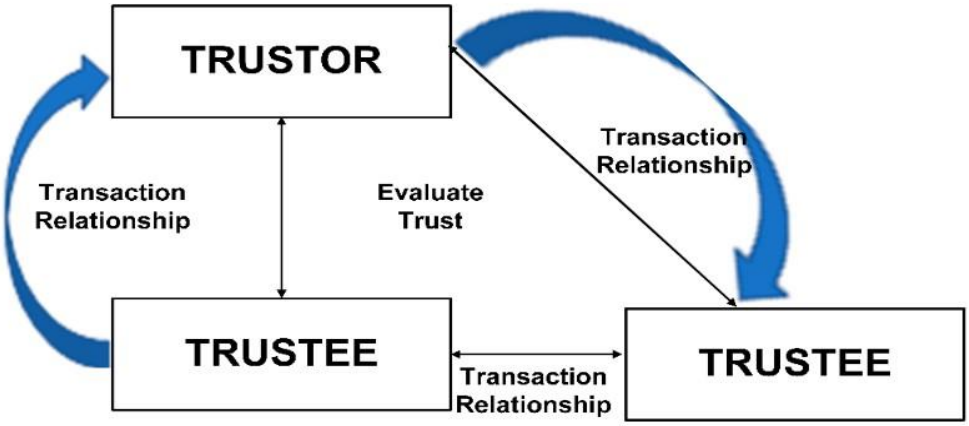

Fig 3.1 Trustor and Trustee Relationship

Trust management is the process of enabling entities to build trust between one another. Trust computing is another division of trust management that deals with establishment, updating, and the processes that support trust establishment. It deals with acquiring trust values, choosing which characteristics to employ, and figuring out how to combine them to determine final trust value [22]. Depending on the application and situation, trust can be transferable or not, dynamic, asymmetrical, and context-dependent, according to Cho et al. [26].

3.2 Structure of Trust

IoT trust computation techniques were categorized by Guo et al. [24] into five design dimensions: trust composition, trust propagation, trust aggregation, trust update, and trust formation.

3.2.1 Composition of Trust

Which trust indicators or components will be used in the computation of trust will depend on this dimension.

You can divide all the metrics into Social Trust and Quality of Service (QoS) categories.

3.2.2 Trust establishment

This attribute determines how the trust should be computed and stored. Trust systems employ two propagation strategies: distributed and centralized.

3.2.3 Trust Updating

This dimension determines how frequently the network's trust value should be updated. Event-driven and time-driven trust updates are the two-basic method.

3.2.4 Trust Building

The combination of the trust indicators, as assessed by trust composition, is defined by trust formation. While some systems simply take into account one trust indication, others take into account numerous trust qualities (multi-trust) (single trust).

3.2.5 Aggregation of Trust

Through self-observation and other people's experiences, this dimension decides how to blend trust into a single value. The weighted sum approach, fuzzy logic, Bayesian inference, logistic regression, and other examples are examples.

3.3 Attacks Based on Trust

A trust management strategy aims to create a extremely reliable network with very honest items. To remain under the threshold and continue to be chosen to provide services, a service may strive to increase its trust value. Due to these trust-related attacks, an SP may act trustworthy for its own gain despite being inferior.

Therefore, a few of the few prevalent trust-related attacks are mentioned below in the context of IoT:

3.3.1 Bad Mouthing Attack (BMA)

To reduce the likelihood of a good node being chosen again, a malicious node may make unfavorable recommendations against it.

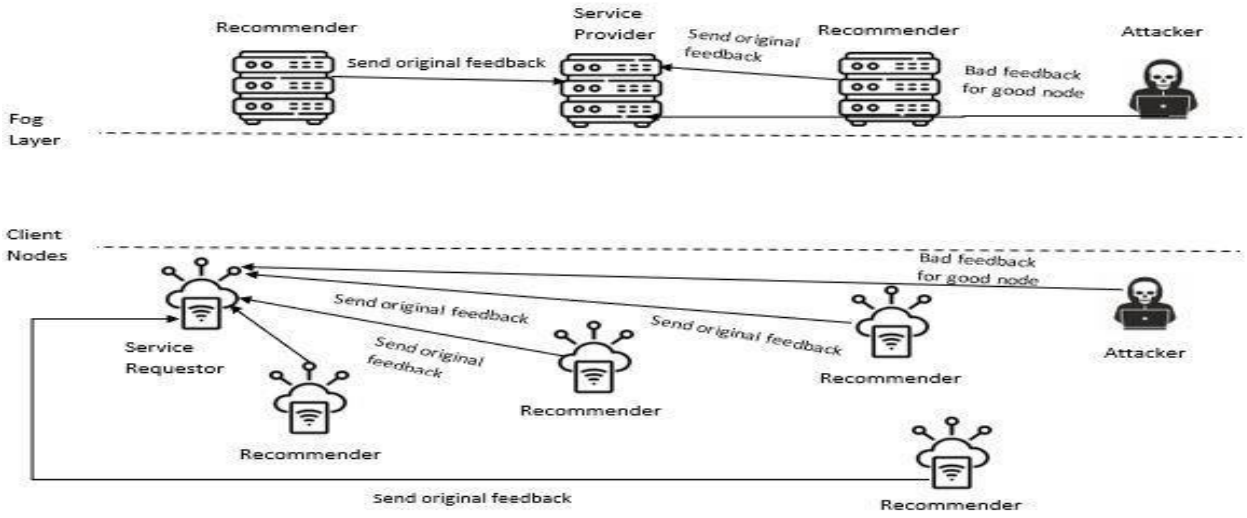

Fig 3.2 Bad mouthing attack

3.3.2 Ballot-stuffing Attacks (BSA)

Even if it is evil, a malicious node can work with other malicious nodes to make positive recommendations for them, which increases their chances of being chosen to provide a service and disrupting the network.

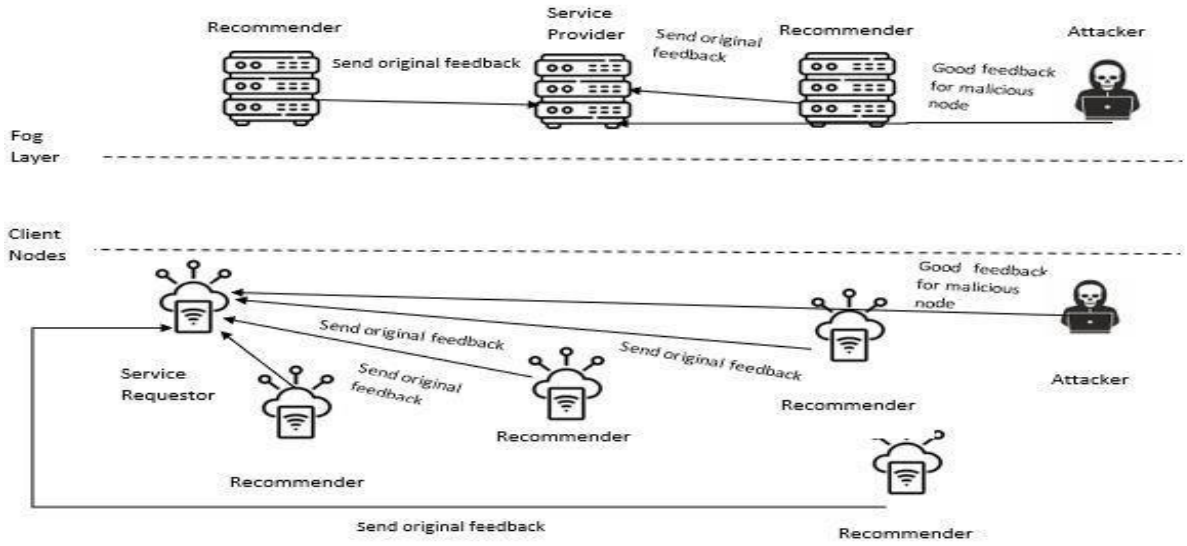

Fig 3.3 Ballot Stuffing Attack

3.3.3 Self-promotion Attacks (SPA)

In order to promote itself, a rogue node may make positive recommendations for itself.

Once it has gained trust, it may then offer subpar services.

3.3.4 On-Off Attacks (OOA)

Malicious nodes randomly aromatize bad and good services in an effort to escape being classified as a bad node. This exploit is used by malicious nodes to avoid detection. In their suggested trust schemes, a number of scholars have made an effort to provide a defense against these trust-based attacks.

3.3.5 Opportunistic Service Attacks (OSA)

This assault takes place when a malicious node notices that its standing within the network is deteriorating; it then starts acting nefariously once it regains its good standing.

Several effective defenses against these attacks are contact likeness, filtering, response circulation, and reliability rating, according to the literature [27].

3.4 Survey of Trust Models

In the recent years, both academia and business have placed a strong emphasis on trust-based security solutions. In the literature, trust calculation systems (TMS) for the IOT have modelled trust using a variation of trust metrics (QoS, social, etc.), the most popular of which are response time, user satisfaction, credibility, honesty, packet deliver ratio, energy consumption, latency, and others [32] [29] [30]. This section reviews the various trust strategies that have been employed thus far in the literature and describes recent developments in trust management schemes.

Researchers use a variety of methods to calculate trust, including weighted method, subjective logic, analysis by regression and the fuzzy logic[22]. A novel reputation system was proposed by Josang et al. [21] using the Bayesian inference technique for the first time.

As a random variable that may be modelled as a probability distribution, trust is plotted. With each optimistic and undesirable event, probability distribution parameters are mapped, an d the average is chosen to calculate trust.

The similar method was utilized by Altaf et al. in [32] to propose a context based adaptive trust model for the Internet of Things utilizing reaction time as the trust parameter. To

defend on off and Sybil attacks, they dynamic assignment of weight factor and used cosine similarity. The adoption of the dynamic weight parameter has been found to behave better in recognizing and isolating malicious nodes when compared to a few other t rust solutions. One of the earliest and most popular methods for aggregating trust is weighted sum, which makes sure that the entities with the best reputation have the greatest influence on the final trust score [22].

A likeness-based model for trust incorporating friendship, contact, and COI similarity has been proposed in [6]. In this study, there was no discussion of an attack model. In a mobile ad hoc network, Wang et al. [28] employed logistic regression to enable SR to forecast SP's future behavior in light of the network's environmental factors, such as capacity restrictions, energy sensitivity, and profit recognition (MANET). It addressed bad mouthing and ballot stuffing but omitted context. A similarity-based methodology was proposed in another study [38] to assist in protecting against IoT badmouthing, collusion and packet dropping attacks. Packet forwarding behavior, feedback reliability, and communal neighbors are the trust features used in this architecture. The similarity-based credibility computation makes this system resistant to fraudulent recommendations.

Table 3.1 depicts the trust models of IOT by different researchers the technique used and the attack cater in these.

Table 3.1 Analysis of IoT Trust Models

Year	Title	Features	Method	Attacks	Main Concern
2017	Trust-Based Decision Making for Health IOT Systems	Aggregate sensing data, derive the probability of health loss, risk,	Direct and indirect observations.	No specific attack mentioned.	Trust assessment
2022	A Novel Hybrid Trustworthy Decentralized Authentication and Data Preservation Model for Digital Healthcare IoT Based CPS	To identify honest miners and restrict malicious miners,	Deep learning, edge blockchain, Cloud blockchain, neural network	Not mentioned	Preventing data fraud, privacy
2020	Analysis of factors affecting IoT-based smart hospital design	Five-layer architecture, a standard protocol design, robust security and privacy design	Clustering, decision tree, logistics regression		Optimization of the power consumption
2021	A Survey on IoT Smart Healthcare: Emerging Technologies, Applications, Challenges, and Future Trends	Upgrades in decentralized capacity, circulated record, interoperability, confirmation, dependable, changelessness.	Blockchain, fog computing,		To encourage secure and astoundingly successful associations.
2021	Security, Privacy and Trust in IoMT Enabled Smart Healthcare System: A Systematic Review of Current and Future Trends	Granularity, compliance with standards, defining the permissions and roles.	Blockchain, lightweight security approach, authentication and authorization		Security, privacy and trust
2020	MITIGATING IOTATTACKS IN SMART MEDICAL NETWORKS USING ENHANCED DIRICHLET BASED ALGORITHM FOR TRUST MANAGEMENT SYSTEM	Distributed trust computation system, considering the service level of the node and the number of packets exchanged	Direct observations and recommendations, Dirichlet algorithm	Selective and Newcomer attacks,	Trust calculation.

2021	A Security Awareness and Protection System for 5G Smart Healthcare Based on Zero-Trust Architecture	Four-dimensional security framework, Risk Scoring, Risk Reporting, Risk Judgment Mechanism	Zero-Trust Architecture		Security Awareness and Protection.
2021	Internet of things in health: Requirements, issues, and gaps	Attribute based encryption	Edge computing interoperability, blockchain		Security, interoperability, usability

making use of the cloud owing to the absence of agility support, incapability to monitor data while it is being transported to the cloud, and other factors, trust management methodologies, while well-established, cannot be immediately used to fog computing. and the scattered nature of fogs themselves. Due to the already established security measures between cloud users and SP, trust is a unidirectional requirement for cloud services.

However, because trust in fog is bi-directional due to its flexible nature, it is challenging to apply the same tactics as utilized in cloud. Since a few years ago, fog computing has been a popular area of study in both business and academia.

The 5G edge cloud's security capabilities, such as the creation of a virtual access network with dynamic scheduling and elastic growth and a closed loop of awareness, analysis, and execution at the edges, are necessary to provide fine-grained security monitoring and control [60].

For fog computing, stronger privacy and security solutions are required [23]. Given the volume of data generated by the sensors and edge devices, the attack surface in fog computing is very large.

For fog computing, stronger privacy and security solutions are required [23]. There is very little research on trust calculating systems in Fog as a safe platform, and context-aware solutions are especially hard to come by. As in other sectors, context is crucial in trust computing. For example, you might be able to trust a node for video surveillance but probably not for a biometric attendance system. The accessible research papers have undergone extensive analysis and study [36] [20] [30] [37].

Al-Khafajiy et al. [36] presented a load balancing technique to effectively manage the assets of fogs and presented a trust and feedback model for fogs utilizing bayes theorem and fuzzy logic. Using parameters for quality of service (QoS) and quality of protection (QOP), SP and SR can both assess one another based on the services offered.

Mansi et al [59] Dirichlet algorithm, where the weight of a node is determined by taking into account the node's service level and the volume of packets exchanged over a period of time Selective and Newcomer attacks are successfully mitigated by the enhanced Dirichlet algorithm.

This methodology employs user experience to forecast future collaborators' behavior, however it omits to address the attack model and merely addresses general trust assaults like forgery and dos. Only a few other trust solutions [34] [42] in fog exist, and while [34] employed a fuzzy neural network combined with weighted weakest link to identify nodes with an accuracy of 0.9969, few other solutions [34] [42] used fuzzy networks for trust estimates.

Aggarwal S. et al. [63] present a fully functional solution for the integration of the proposed consensus algorithm with the Ethereum Framework, but Sabotage is not covered in this research. The proposed Proof of Improved Consensus algorithm is designed for blockchain to validate the blocks before they are committed to the ledger.

Muhammad Asif et al. [61] proposed a technique to ensure privacy of medical data especially against the threats emerging internally within SHS. The system had implemented authorization defining the permissions and roles merely for medical staff. , the system does not facilitate to perform copy and move operations on directory resource. Seyed Morteza et al. [61] presented an effective and secure method called “MedSBA” for storing medical data in SHS. The method is based on blockchain technology to ensure user privacy The system employed private blockchain to revoke the instant access which is very challenging in ABE. Nonetheless, the system did not support the exchange of cryptocurrency between the data consumer organizations and the individuals for data sharing.

Various academics have employed fog as an additional layer to help with trust estimate [20] [44] [41]. Fuzzy logic was used by Soleymani et al. [12] to propose a trust model for vehicle ad hoc networks. To calculate trust, a combination of experience, location accuracy, and plausibility model is used. In [40], Regression analysis was utilized to model fog as an additional layer to represent confidence in a sensor based centralized system. The fog layer

was used to reduce computational costs and to improve the network's storage capacity. However, they did not explicitly provide a trust model for the fog network; rather, they used fog as an enabler to assess position correctness. In [37], a fuzzy logic-based broker-based trust management system for fogs was originally presented. Availability, feedback, security, QoS, and cost were among the trust metrics employed in this model, however the usage of a broker suggests a single point of failure, and it also neglected to address the prevention of harmful recommendations. The clue of a fog manager or trust manager was first offered in [31] and [42]. It is in charge of requests handling on the basis of resource limitations and time-criticality in order to enhance QoS at the layer of fog. A review-grounded system increases the system's dependability and credibility.

If items with a shared purpose develop social connections, the load on the Internet of Things system as a whole can be decreased. [49]. IoT trust models must be developed using the Social Internet of Things (SIOT). TMS in SIOT can serve as an inspiration for fog trust management systems. CoI, social contact, friendship, centrality, co-location, honesty, cowork, centrality ownership, mutuality, and other relationships are some of the most often utilized trust metrics in the SIOT. [20] [49] [50] [3] [48].

A node of fog might propose various facilities to various systems, and just because it is trusted for one service does not mean that it will be reliable for all. For instance, a node that is reliable for video scrutiny might not be reliable enough for a management system of traffic or a health-related Internet of Things application. [48]. Context should always be at the forefront when discussing trust in fogs since context and environmental elements affect trust differently. There are now some trust solutions for fog computing, as was previously said, although this field of study is still in its early stages. To the best of our knowledge, there are just two studies that explore context and trust in fogs, and even those papers have certain limitations. In an effort to present a context-aware trust model employing a feedback crawler system to create confidence, Yasir Hussain et al. [20] have introduced a monitor mode for observing the activity of hostile nodes. The likeness between two model sets has been calculated in works using a variety of statistical techniques, i-e Cosine, Pearson and Jaccard correlation, each with advantages and disadvantages [48].

Table 3.2 Analysis of Fog Trust Models

Paper	Method	Context dependent	Similarity measures	Adaptive	Limitations	Attacks
Future generation computer system	Subjective logic	Yes	No	Yes	High overhead Fog to fog no collaboration	No
A fog computing trust management approach," Journal of Parallel and Distributed Computing	Bayes fuzzy logic	No	No	No	Fog workload Service workload	No
A novel trust mechanism based on fog computing in sensor–cloud system	Regression analysis	No	No	Yes	Static environment No heterogeneity	Jamming, Spoofing
Context-aware trust and reputation model for fog-based iot,"	Feedback based	Yes	No	No	Not deal with malicious trust recommendations	No
Trust management in fog computing,	Logistics regression, subjective logic	No	No	No	No practical evaluation	No
A trust management system for fog computing services," Internet of Things,	Fuzzy logic	No	No	No	QOS,QOoS ans social relationship	No
A Context-based Trust Management System for the Social Internet of Things	Decision tree	Yes	No	Yes	Trust prediction onlybon decision tree	No
Trust Management for SOA-based IoT and Its Application to Service Composition		No	Yes	Yes	Similarity rating of friendship, social contract.	Collusion attack, opportunistic service

Table 3.2 lists the trust models we've looked at in fogs. Our analysis of the literature specifies that there is a need for adaptive trust calculation resolutions for fog systems nonetheless, as far as we are aware, no context-oriented adaptive trust management for fogs has ever included a social similarity measure to determine trust. As was previously

mentioned, similarity measures are employed to screen out fraudulent recommendations from nodes that are in the same scenario as the SR. We have discovered a huge research requirement given the flexible character of trust, particularly in a dispersed fog environment, especially when trust is a crucial factor for evaluating the safety of a specific endpoint that you are delivering or receiving facilities from. So, utilizing the Bayes approach and the context similarity measure as a suggestion filtering methodology, our goal is to present a adaptive scenario-based trust resolution for the fog network. Our use case will be the smart healthcare system.

Smart healthcare is one of the expanding use cases for fog computing. [1]. Fog computing offers the reliability that time-sensitive smart medical systems require as to its ability to operate near to the user and independence from central architecture. Fog computing has been used and can be used in a number of healthcare-related applications, including automated patient supervision, remote monitoring, and smart medical equipment [14] [9]. There aren't many proposals for system architecture, encryption models, or fog models in the literature [2][11].

In order to develop a trust model for fogs utilizing healthcare as a use case, Mutahdi et al. [47] employed subjective reasoning; nevertheless, they neglected to take context into account when addressing how crucial data loss, data breach, and Denial of service (DoS) attack is in a smart healthcare setup.

Due to increased device connectivity brought on by the need for ongoing observation and the sensitivity of patient data, security measures are required. Therefore, the goal of our research is to offer a fog computing-based context- adaptive trust resolution for time-sensitive smart medical systems.

Chapter 4

PROPOSED CONTEXT-BASED FOG TRUST MODEL

The proposed model of trust, mathematical model, flow of event, and evaluation mechanism are all presented in this chapter. The Internet of Things (IoT) devices, fog nodes, and cloud servers are the three stakeholders in a fog computing ecosystem. Using this proposed approach, trust between fog nodes that are in communication in a smart medical is calculated. It is a dispersed trust paradigm where, depending on the context, each node determines how much it may trust the node with which it is interacting.

4.1 System Model:

Service Requestors (SR), Service Providers (SP), Trust Managers (TM), and Recommenders are the four entities in our suggested model. Client nodes that seek services are called service requestors (SR). Because they have restricted processing and storage power, they must trust on unrestricted nodes, or TM, to perform computations and data processing that is above their capacity.

4.1.1 Attacks Used in Model

Our research focuses on mainly two attacks namely badmouthing and ballot stuffing. In badmouthing attacker collude bad recommendation about the victim to damage or destroy its reputation. While in ballot stuffing it boost trust value of another bad note by providing good recommendation for collusion purpose.

4.1.2 Fog Model

As much as they are able to, SR keeps a local trust table, but when their processing power is fully utilized, they rely on TM to do trust calculations and service discovery. Unrestricted nodes include Trust Managers with sufficient processing and storage power. Trust Managers are highly competent fog nodes that relieve fog nodes with lesser capacity of the

task of performing difficult calculations. In order to boost dependability and prevent a single point of failure, there can be a chain of TM, so if one does not have a certain piece of data that SP or SR wants, it can forward the request on to the TM in line behind it. Service providers (SP) are the endpoint that deliver the necessary facilities, as opposed to feedbacks, which are endpoint having any prior involvement utilizing the server for specific facility. This approach will address the development of trust between nodes of fog for resource dumping and sharing.

4.1.3 Fog Layers

Instead of considering a multi-layered fog environment for simplicity's sake, we will focus on a single layer of fog that nevertheless has all of the desired characteristics. Fog nodes can interact with nearby fog servers or clients through a single hop. Trust Managers are thought to be very trustworthy. Nodes having a trust rating greater than 0.5 are considered to be trustworthy.

4.1.4 Proposed Paradigm

In our suggested paradigm, the SR asks the SP for a service, but they first need to confirm one another based on previous interactions. This stops a connection from being made to a malicious or rogue node. A service requester (SR) will ping its neighbors to inquire about the availability of a service S, and a potential service provider (SP) will then answer.

Nevertheless, in order to protect themselves from malicious nodes, SP and SR must first verify one another. If the SR's local trust database does not show any evidence of prior contacts with the SP, a default value of 0.5 (neutral) will be given to experience that is direct. SR solicits recommendations from its neighbors, then determines indirect trust based on these recommendations, and ultimately determines overall trust by combining indirect and direct trust. The request will be immediately refused if SP's trust rating is below the cutoff, and SR will search the network to look for a substitute connection. SR directs a connection demand to SP if SP's trust score above the cutoff. The SP will now perform the same procedures to validate the SR before connecting. If the value exceeds the threshold, the connection is formed. SP offers SR the service S. For use in subsequent contacts, SR saves customer input in the structure of experience and overall trust score after receiving the service. This will lower the chances for both the attacks to occur as because of recommendation and previous interaction record. So, when the attacker gives bad

recommendation to good node it will not selected because of context of the transactions and vice versa.

4.2 Trust Estimation

The method for computing trust is described in this section. It contains scheming direct trust with the help of Bayes algorithm, calculating indirect trust by by means of suggestions from adjacent nodes and evaluating their similarity values to SR, and lastly calculating whole trust. The parameters of our trust model are provided.

Lists are used to store the following data in each node:

List of Co-work of a node $Co_a = \{Co_1, Co_2, Co_3, \ldots Co_n\}$

Servers list, a node has networked with and gotten services from $S_{\acute{a}} = \{s_1, s_2, s_3, \ldots s_n\}$

Service list a node has obtained $Sr_a = \{sr_1, sr_2, sr_3, \ldots. sr_n\}$

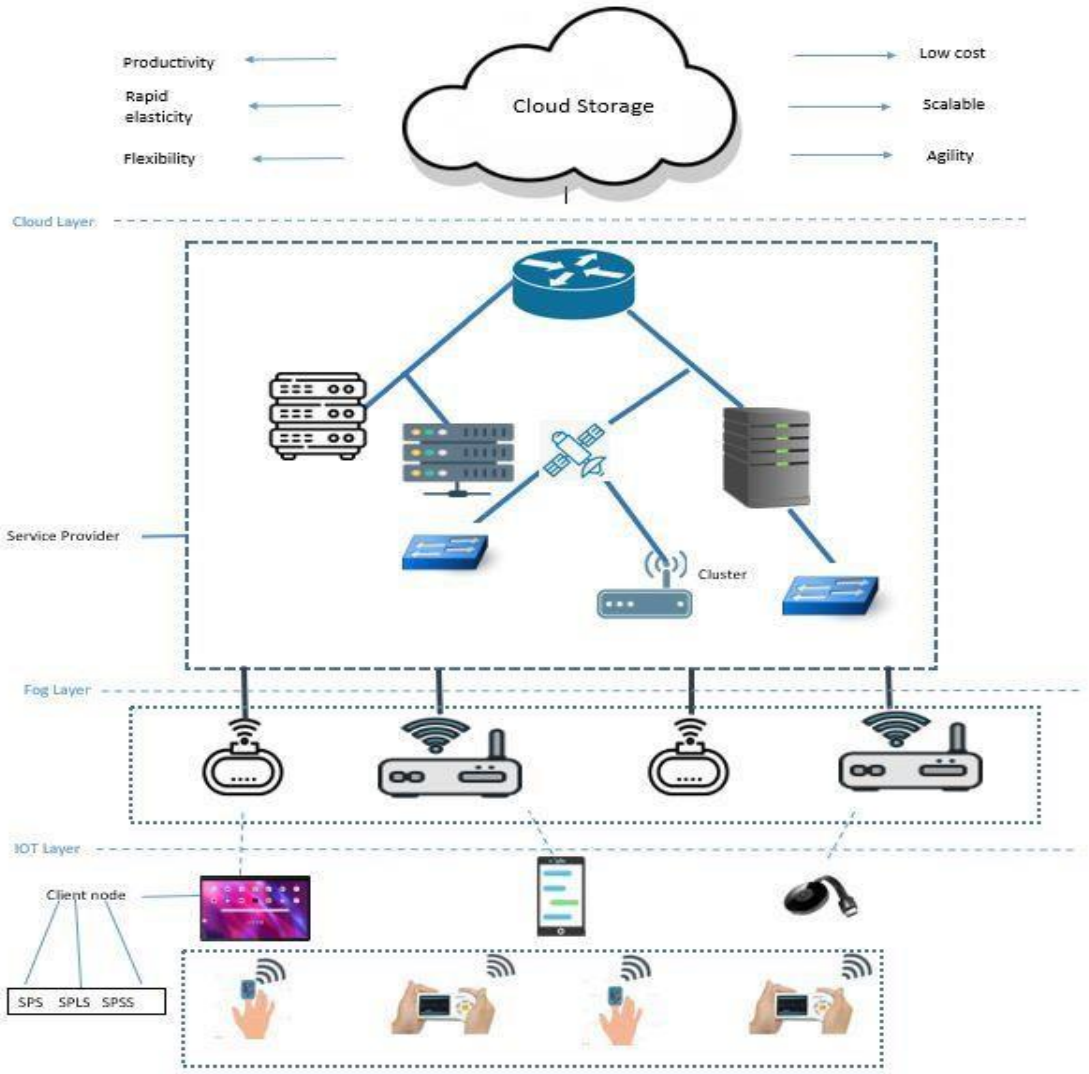

Figure 4.1 List Storage Nodes

Table 4.1 Description of Parameters

Parameters	Description
$S_{a,b}$	User Satisfaction experience
$e^{-d\Delta t}$	Exponential Decay
Δt	Trust Update Cycle
w	Weight Parameter
T_{abd}	Direct Trust
T_{abr}	Indirect Trust
T_{ab}	Total Trust

sim_{acs}	Server Similarity of node a to take recommendation from node c
sim_{acCo}	Co work Similarity of node a to take recommendation from node c
sim_{acSr}	Service Similarity of node a to take recommendation from node c
L_{Sac}	List of shortlisted users on the basis of Server Similarity
$L_{Co_{ac}}$	List of shortlisted users on the basis of Co work Similarity
$L_{Sr_{ac}}$	List of shortlisted users on the basis of Service Similarity
L_{ac}	List of shortlisted users on the basis of highest similarity value with a

4.2.1 Parameter for Direct Trust

We practice the Bayesian approach to determine the direct trust between the communicating nodes in our model. Because of its successful track record with effective trust modelling, Bayes is often used. Direct User Satisfaction Experience S_{ab} and Bayes are used. User experience is calculated using a amount of nonfunctional features, such as latency, packet delivery ratio, response time, etc.

4.2.1.1 Packet Delivery Ratio

The packet delivery ratio can be calculated by dividing the total number of data packets that have reached their destinations by the total number of packets that have been delivered from sourcesThe ratio of packets supplied from the source to those received at the destination is known as the packet delivery ratio. The ratio of successfully delivered packets to all packets sent is measured.

$$\frac{\Sigma\ (\text{Total packets received by all destination node})}{\Sigma\ (\text{Total packets send by all source node})} \tag{4.1}$$

Based on the quality-of-service metric, in this example packet delivery ratio, user satisfaction is a binary number that can range from for satisfied 1 to for not satisfied 0. By considering the number of packets sent and the number of packets received user satisfaction experience is calculated for every round of communication. When the packets are sent to

the requestor and it receive the packets and then send the response then it is one round of communication.

According to the Bernoulli trial distribution, the experience is split between good and unsatisfactory. Direct trust between node a and node b can be depicted as:

$$T_{ab}^{d} = \frac{\alpha_{ab}}{\alpha_{ab}+\beta ab} \quad (4.2)$$

In equation 4.2, α and β are beta dispersal's parameters. Value of α and β is derived taking into account trust deterioration over a time period t. The effect of prior trust levels on the present value is modelled by trust decay when the time elapse between two activities is taken into account. When there hasn't been any interaction between two entities for a while, trust between them gradually declines, much like in the real world. Because it is based on the trustor's estimation of the trustee's trustworthiness, trust decay only affects direct trust and not total trust. Hence, it is possible to compute the beta distribution's parameters using following equation 4.3 and 4.4:

$$\alpha_{a,b} = e^{-d\Delta t} \times \alpha'_{a,b} + S_{a,b} \quad (4.3)$$

$$\beta_{a,b} = e^{-d\Delta t} \times \beta'_{a,b} + \left(1 - S_{a,b}\right) \quad (4.4)$$

Here, satisfaction experience of a towards b is represented by $S_{a,\,b,}$ it is a binary value according to $S_{a,\,b}$, where 1 denotes a satisfied experience and 0 a dissatisfied experience. In aforementioned equations, $S_{a,\,b}$ donates to positive observations whereas $(1-S_{a,\,b})$ donates to observations that are negative, $\alpha_{a,b}{}'$ and $\beta_{a,b}{}'$ represent old score where as $\alpha_{a,\,b}$and $\beta_{a,\,b}$represent new values. $e^{-d\Delta t}$ represents exponential decay where over a period Δt d is the decay factor.

4.2.2 Measures for Indirect Trust

We may assess indirect trust in this case by using suggestions from adjacent nodes with a history of communications with the same server in the same situation. Node a will ask its

neighboring nodes for their trust recommendations for node b, and those nodes will then share the total trust values with SR. and, if necessary, use TM to determine the node's closeness to the recommender nodes using a similarity measure. To determine how similar two entities are, there are a number of similarity approaches [51] that can be used, such as Cosine, Pearson Correlation Jaccard, etc. Jaccard will be used in this research for resource limited and time-critical IoT systems, it is simple and computationally efficient. The Jaccard ratio is defined as the size of the intersection of sample sets divided by the size of the union of sample sets. The scale for similarity ranges from 0 to 1, with values between 0.5 and 1 denoting similarity and values between 0 and 0.499 denoting dissimilarity.
Similarity of server, co work and service are considered in this trust model.

4.2.2.1 Co-work Similarity:

Co work similarity indicates same work collaboratively done by the recommender and hence, SR feels the same way about the service providers who offer the same service. This will lessen the chances of bad recommender to provide recommendations. List L^s_{ac}will be filtered by comparing the recommenders' and SR's co-work lists to see how comparable their co-work is using following equations:

$$sim^s_{ac} = \frac{|Co_a \cap Co_c|}{|Co_a \cup Co_c|} \quad (4.5)$$

Where $Co_a$and $Co_c$are co-work lists of node a and recommender c, if similarity is equal to or above from 0.5, then it will be added to the list L^c_{ac}. Now, recommenders with similar coworkers and similar attitudes towards servers will be included in this filtered list. The data will then be further filtered in the following phase to only include nodes that have utilized identical services from same servers inside identical circumstances.

Server Similarity:

If SP receives suggestions from numerous nearby nodes, the initial step will be to determine server similarity. As a result, only recommenders who have utilized the same servers for their services will be taken into account. This will get rid of bad recommendations. After two nodes a and c communicate their lists of servers S_a and S_c, Jaccard similarity can be calculated as:

$$sim_{ac}^{s} = \frac{|S_a \cap S_c|}{|S_a \cup S_c|} \quad (4.6)$$

If similarity is equal to or above 0.5, then in list $L^{s}_{a,c}$ it will be added. This comprise nodes with similarity above than 0.5 with SR_a. This list will then be further assessed and calculated after being obtained. Other similarity measures that are used are

Service Similarity:

List L^{c}_{ac}will be filtered to only pick nodes that received SR's services in same situation from same server. Nodes will swap their services list Sr and the following equation will be used that calculate the similarity with the help of Jaccard method:

$$sim_{ac}^{Sr} = \frac{|Sr_a \cap Sr_c|}{|Sr_a \cup Sr_c|} \quad (4.7)$$

If similarity is or equal to or above 0.5, then to the final list $L^{Sr}_{a,c}$ it will be added which currently only includes nodes that are socially similar and hence can be trusted with suggestions. The sum of the three similarity measurements will determine how similar two nodes are to one another.

$$sim_{ac} = \sum_{i=S,Co,Sr} . sim_{ac}$$

Using the highest determined similarity value, service requester can now choose from the list $L^{Sr}_{a,c}$ the topmost n recommenders. A list L_{ac} can be used to represent the top n nodes. Using equation 4.8, it is now possible to determine indirect trust or suggested trust.:

$$T^{r}_{ab} = \sum_{L_{ac}} \frac{sim_{a,c}}{\sum_{L_{ac}} sim_{a,c}} \cdot T^{d}_{c,b} \quad (4.8)$$

Here, list L_{ac} which has the highest similarity values, represents the top n nodes. $T^{d}_{c,b}$is the node c's direct trust in b, which represents the direct interaction between c and b. Each interaction is weighed by the proportion of a recommender's similarity score to the total similarity value of all feedbacks.

4.2.3 Total Trust:

Total trust gathering direct and indirect trust will yield the SR_a T_{ab} towards SP_c.

$$T_{ac} = w.\, T_{ac}{}^{d}\ (1- w).\, T^{r}{}_{ac} \tag{4.9}$$

Where *w* is the parameter of weight used to balance the weights of indirect and direct trust when determining the final level of trust. It is changed constantly and dynamically to thwart hostile attempts, such as bad mouthing and ballot stuffing . With the adaptive weighting the chances of malicious recommender in interaction will weaken.

4.2.4 Entropy based Weight Parameter

To reduce the biasness of estimation of trust, the weighing parameter *w* (0 <*w* <1) is adaptively modified. Indirect trust or direct trust value will be given more weight depending on the value of the *w*. This dynamic weight parameter assignment aids in choosing between direct trust and recommended as the value to put more faith in. Both dynamic and static weighting have been employed to derive weights in the literature. To make our trust model more adaptable, we'll use an entropy-based weighting mechanism to generate the indirect and direct trust weights on a dynamic basis. Entropy evaluate the degree of randomness or uncertainty in an incident or event [52]. The values of entropy of indirect and direct trust are considered by next formulas:

$$\mathrm{H}(T_{ab}{}^{d}) = -\,T^{d}{}_{ab}\,\log_2(T_{ab}{}^{d}) - (1 - T_{ab}{}^{d})\log_2(1 - T^{d}{}_{ab}) \tag{4.10}$$

$$\mathrm{H}(T_{ab}{}^{r}) = -\,T^{r}{}_{ab}\,\log_2(T_{ab}{}^{r}) - (1 - T^{r}{}_{ab})\log_2(1 - T_{ab}{}^{r}) \tag{4.11}$$

Where H ($T_{ab}{}^{d}$) and H ($T_{ab}{}^{r}$) are the entropy value of direct trust and indirect trust respectively. Using both of the entropy values determined previously, the value of w is computed using the following equation:

$$w = \frac{1-\frac{H\left(T^{d}_{ab}\right)}{\log_2\left(T^{d}_{ab}\right)}}{\left(1-\frac{H\left(T^{d}_{ab}\right)}{\log_2\left(T^{d}_{ab}\right)}\right)+\left(1-\frac{H\left(T^{r}_{ab}\right)}{\log_2\left(T^{r}_{ab}\right)}\right)} \tag{4.12}$$

Weights that are flexible help stop a variety of trust-related assaults, such as opportunistic service attacks, vote-stuffing, ballot-stuffing, and badmouthing. Based on the previous data it has saved, each node determines value of itself w and update it adaptively with each cycle of trust. In the static weighing approach, the weights' values are fixed before the final trust calculation. The higher the weights' values, the more significant direct trust is to the final score; conversely, if they are given low values, the trust more significant to total trust is direct trust. Thus, our study employs an adaptive weighting strategy to ensure that the weight values are adaptively determined on the basis of prior trust values and one bad direct value experience and recommendation won't change the overall trust perspective of a node. Static weight creates a false foundation since it enables malevolent nodes to manipulate trust scores at will and may result in an incorrect assessment of trust values. Dynamic weighing is impartial and trustworthy and the chances for ballot-stuffing and badmouthing attack to occur can be removed. In the next sections, we'll examine how weights with adaptive values affect direct, indirect, and total trust.

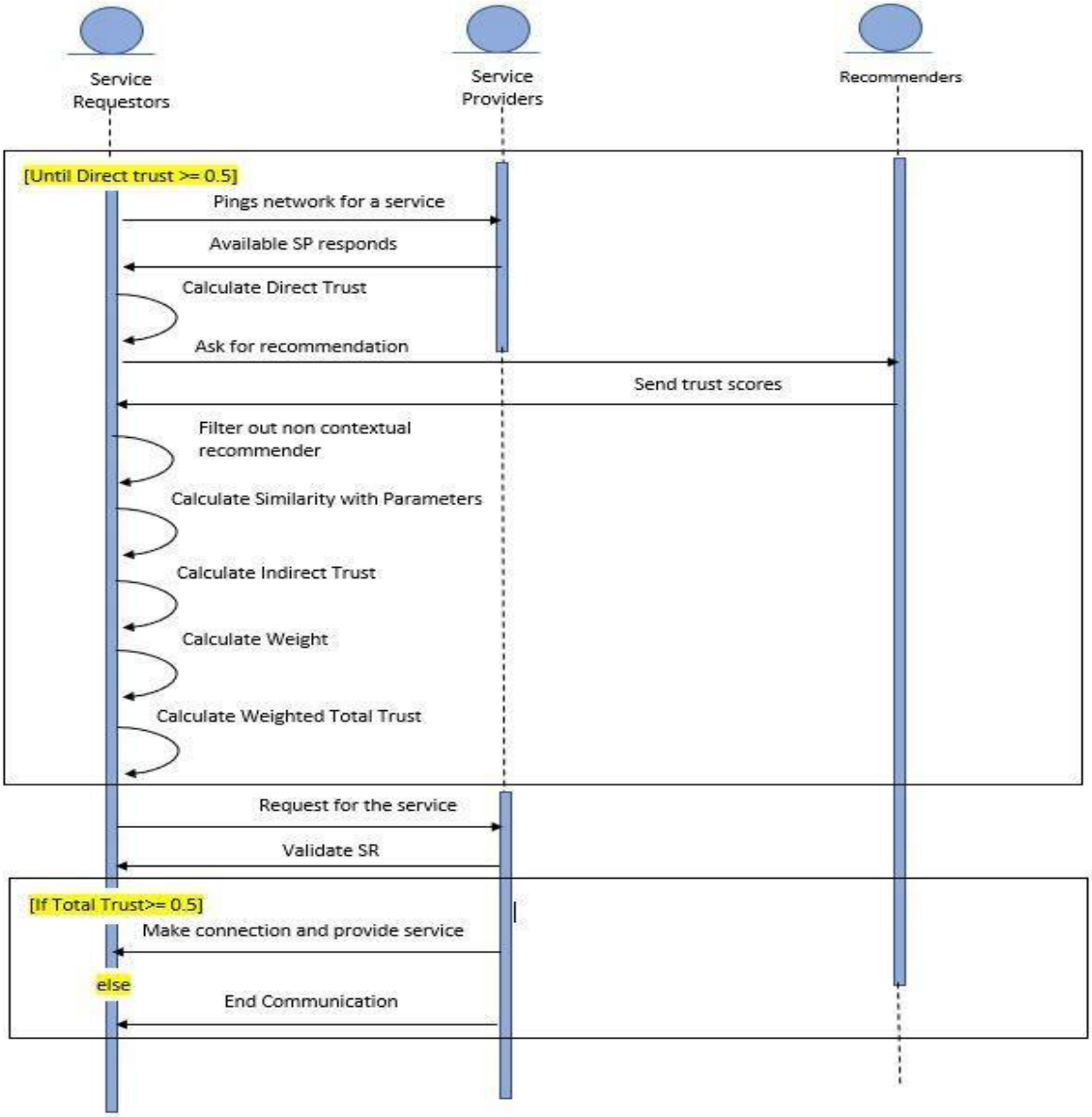

Fig 4.2 Sequence Diagram

4.3 Attacks on Trust Management Systems and the Resilience of SQT to these Attacks

The purpose of adaptive trust system is to construct a reliable network with a high trust threshold; in such a system, malevolent objects often function well to rise in the network hierarchy only to later cause trouble. Our feedback aggregation solution counters bad

mouthing and ballot stuffing attack by taking into account the degree of trust between the recommender and the trustor and weighting the recommendation accordingly. By doing this no malicious node will be considered as recommender for the server. Also, by taking context into account these attacks can also be minimized.

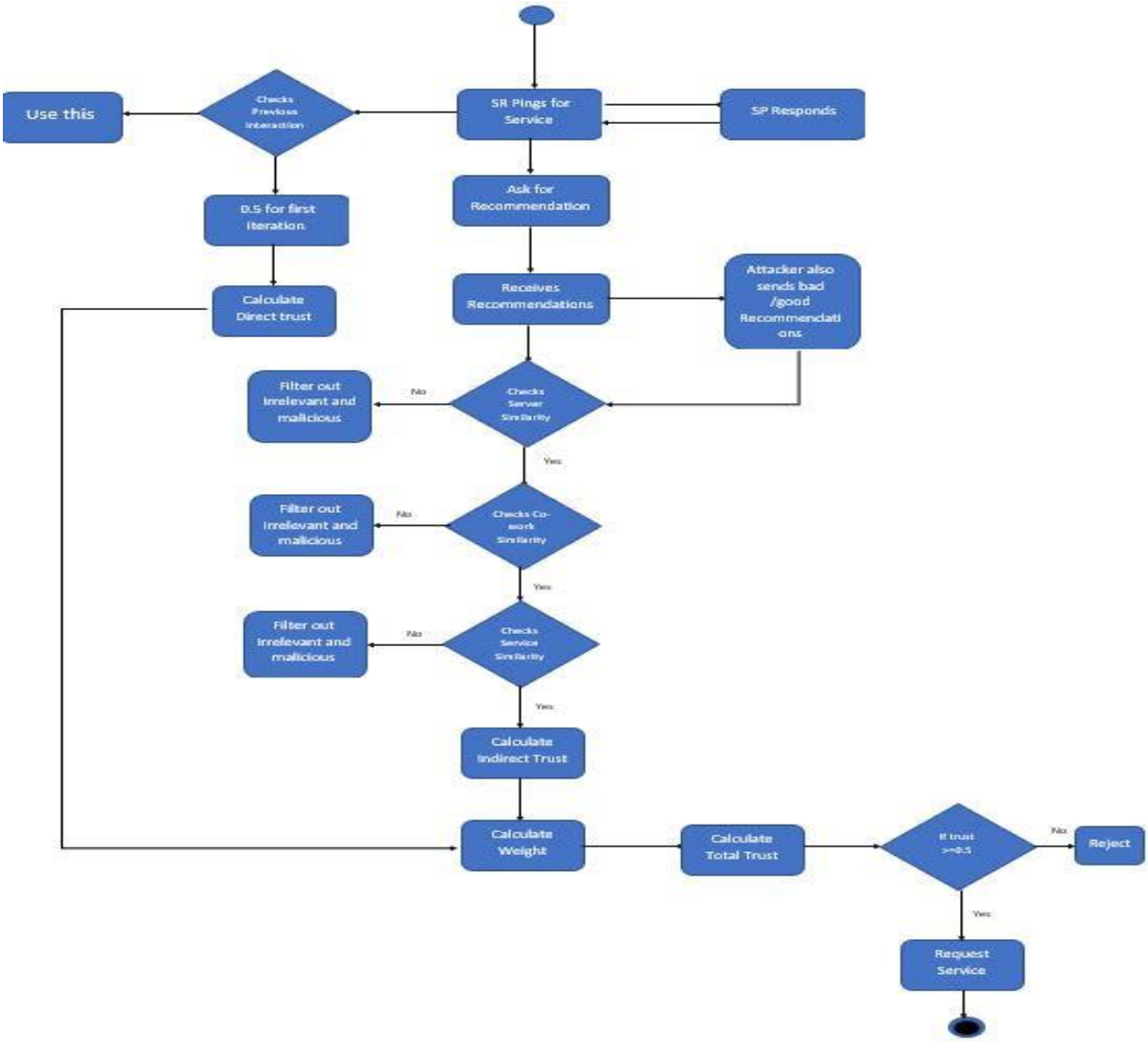

Fig 4.3 Activity Diagram

Chapter 5

EVALUATION AND SIMULATION OF PROPOSED MODEL

In this part, we present the results by simulating our system and conducted using our suggested context-based adaptive trust model. With the simulation a scenario involving smart medical system, we examine the performance of our suggested approach. In our suggested scenario, fog server nodes and client nodes and malicious nodes interact with one another, and overall trust is calculated based on client node user experience.

5.1 Simulation

Let's look at a scenario of a smart healthcare network in Fig. 5.1, where a remote monitoring system based on fog is in place to keep track of patients who need exhaustive care. Every patient is equipped with a variety of sensor that are wearable, equipment, including as a heart rate, a monitor of temperature, ECG, and pulse monitoring, a respiration rate monetization, etc., which gather and transmit raw medical data to the layer of fog for processing. Continuous monitoring of the health state of patient is necessary; low latency and time sensitivity are two of the utmost crucial factors, so we use fog nodes unlike the cloud, they are located closer to the sensor devices, allowing them to provide real-time analysis without latency. Fog nodes can both create emergency medical warnings for patients and clinicians and occasionally for in depth study and for storage sends updates to cloud. Nodes of fog must first build confidence among themselves because data is so crucially important. To achieve this, we put our suggested model into practice and evaluated its effectiveness. The platforms Contiki-NG and Cooja are used to execute our simulations. A platform called Contiki was created to connect low-power Internet of Things devices to the cloud, while the network simulator Cooja uses Contiki motes to model both large and small networks under varied circumstances.

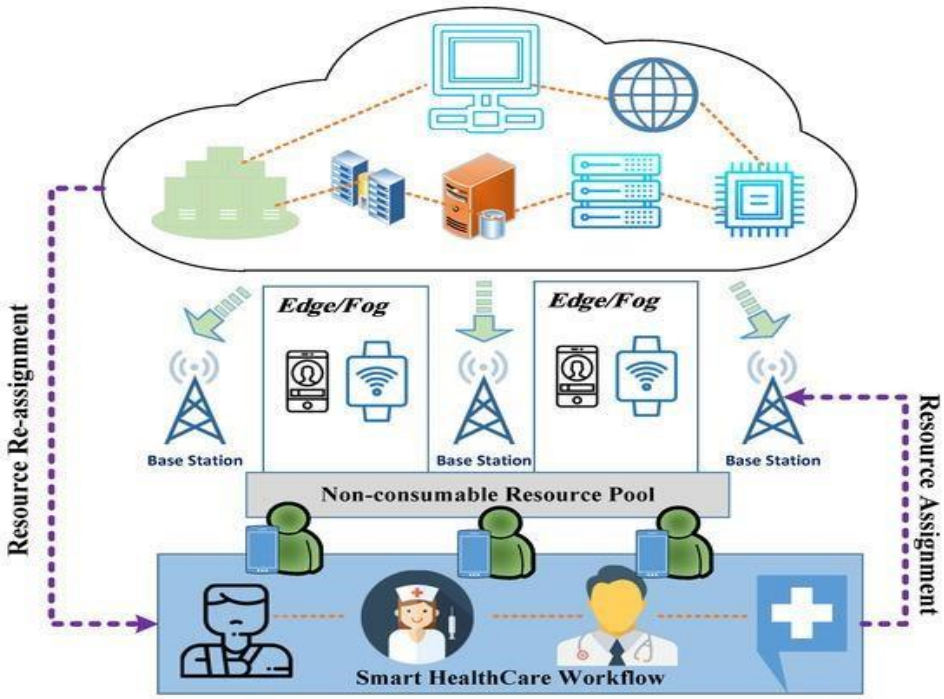

Figure 5.1 Smart healthcare System

An open-source operating system for the IoT of the future is Contiki-NG [55]. The number of packets sent or transmitted from sender to the actual number of packets received to receiver in one complete round from sending request to establish communication and receive the required services is packet delivery ratio and we use this as our parameter. To obtain the user experience value S_{ab} (0 S_{ab} 1), the packet delivery ratio is normalized. S_{ab} is then employed in direct trust calculations.

For this article, we used a straightforward fog system scenario, as seen in Fig. 5.2. Service requesters are the green highlighted nodes, and providers of service are the orange nodes while the purple ones are the malicious nodes. Each node in our simulated network will consider suggestions from neighbors of its 1-hop to create a judgement about its service provider. In this network, any node can be a service provider or a service requester SR. We've taken a modest network for the purpose of simplicity, with node 4 acting as our service requester, node 1 as our service provider, and 5 nodes acting as recommenders in which node 7 is attacker node. Our simulation took two hours and six minutes to complete and is event driven in our trust model.

5.2 Case Study

Let's assume that node (1) answers to a ping sent by SR node (4) looking for a service. Assuming this is their initial encounter, SR will look at their prior interactions with node 1. We'll take the value of S_{ab} to be 0.5 initially, and value of and will be treated as 1 for the first encounter. For the sake of this example, let's say that the SR requests approvals from five neighboring nodes. The neighbors exchange their trust scores with the SR, and then it uses the recommenders' trust scores to generate its indirect trust value by contextually filtering them. Using Jaccard measures, we shall determine similarity.

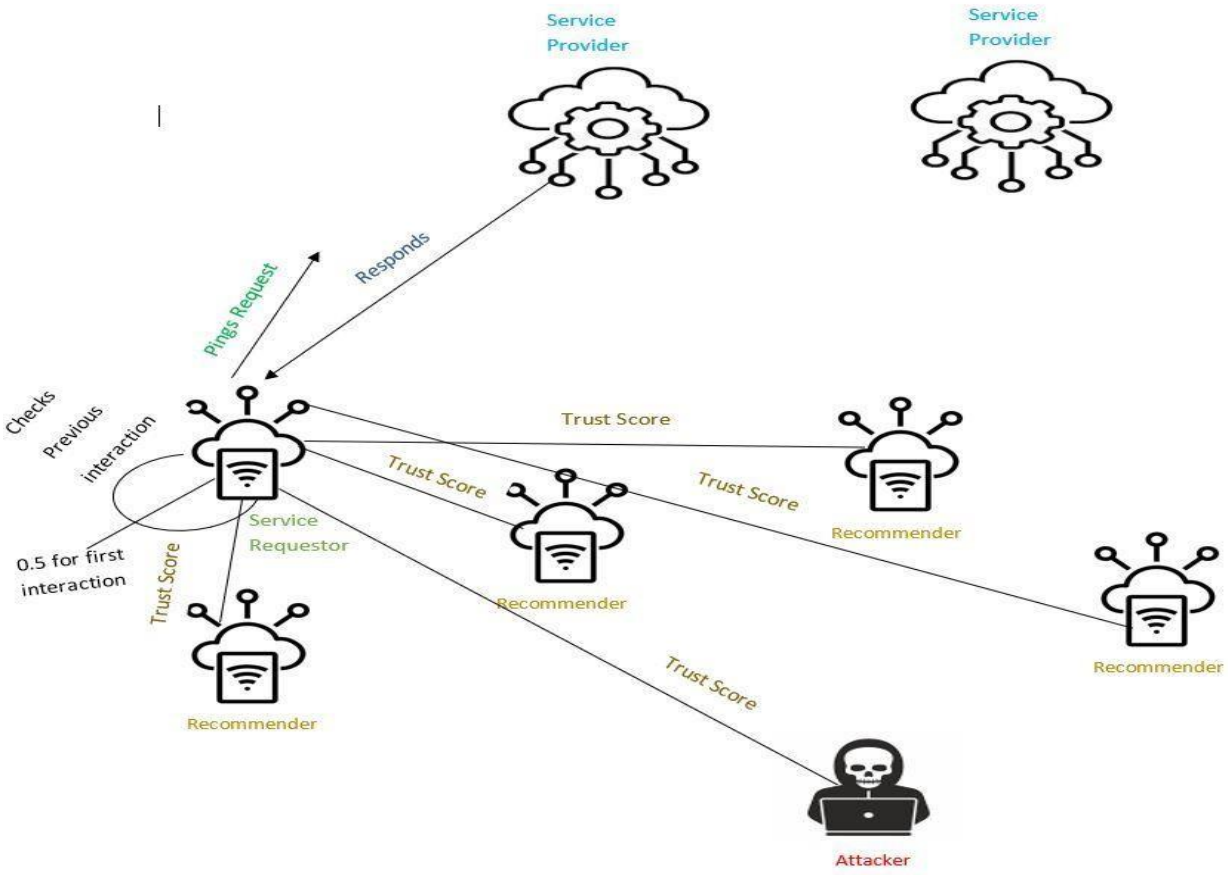

Fig 5.2 Case Study Visual Representation

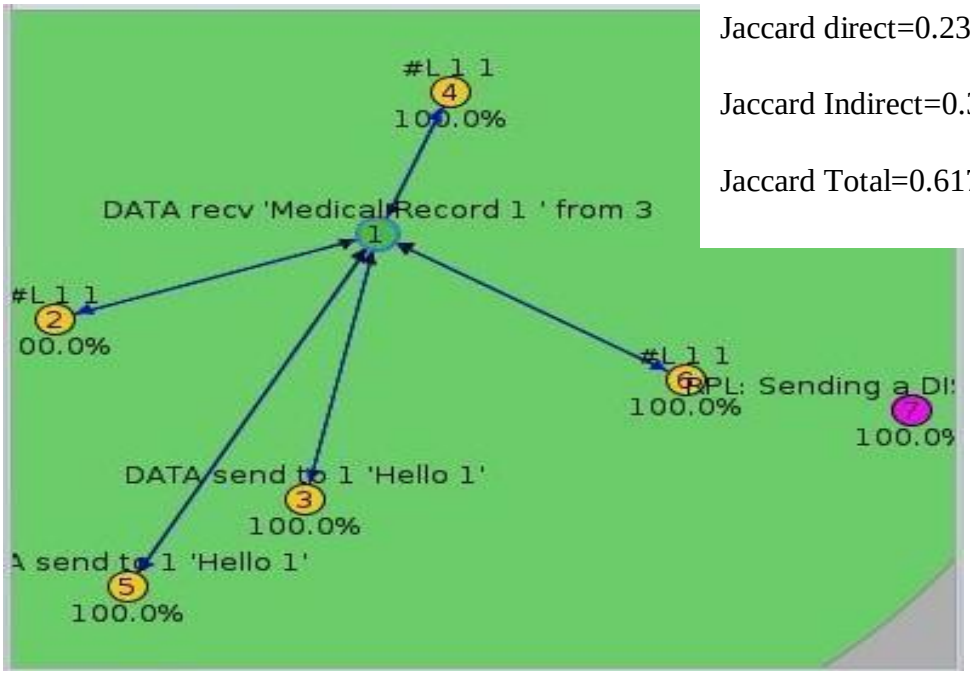

Fig 5.3 Contiki, Cooja fog network for ballot stuffing

First, co work similarity is generated for that narrowed list, and only those with similar co work connections to the SR are taken into consideration. Then, similarity of server is determined, by which the list contracts to recommenders with same servers. The final step is to calculate service similarity, which contracts down the list to concluding recommenders list, L_{ac}, which now only includes individuals who use the same services from the same servers and share the same co work. After final filtering, our list is down to two recommenders because only the top n suggestions will be taken into account and the malicious node due to not matching the contextual similarity will evicted. Equation 4.2 will be used to compute indirect trust. And indirect trust score is 0.8523523using Jaccard, and the direct trust value is 0.5. We'll use 0.5 for both during the first interaction, and for subsequent sessions, we'll use the historic value saved from the previous interaction to calculate entropy and finally weights w. The value of weight will be dynamically calculated using historical values of indirect and direct trust. SR sends a connection request if whole trust is more than 0.5 (in this case, it is 0.6761using Jaccard), and SP uses the same procedures to determine trust if it is within the threshold. If it is, the connection is formed, the service is supplied, For the following interaction, in the form of user satisfaction the reaction time is stored by SR. A node is prohibited and detached from the network if five times in a row its trust value is less than 0.5. We can avoid ballot stuffing and badmouthing attacks by doing this and ensuring that no node is unfairly eliminated.

Table 5.1 Results and Value

No	Experience	Direct Trust unweighted	Sum of Similarity Jaccard	Indirect Trust Unweighted	Total Trust Unweighted	Direct weight using Jaccard	Indirect weight using Jaccard	Jaccard Direct	Jaccard Indirect	Total Trust Jaccard	Past alpha	Past beta
1	0.5	0.5	1.8, 1.98	0.852353	1.352353	0.5	0.5	0.25	0.4261	0.6761	1	1
2	0.6175	0.585532	1.8 ,1.98	0.852353	1.437884 3	0.8116517 8	1.2296289	0.2417541 8	0.3519009 22	0.5936291 76	0.16624	0.61082
3	0.56	0.479678	1.8, 1.98	0.852353	1.332030 3	0.7988941 8	0.9357513 9	0.2209498 2	0.3926653 58	0.6135538 92	0.5784239 8	0.6276295
4	0.288	0.496196	1.8, 1.98	0.852353	1.348548 3	0.8809321 6	0.9663397 4	0.2163490 3	0.3711382 96	0.5871294 03	0.3527207	0.3575736
5	0.336	0.499631 4	1.8, 1.98	0.852353	0.425861 97	0.7524848 98	0.9514278 4	0.2227634 9	0.3800672 35	0.6027261 94	0.3757309	0.3756398 7
6	0.8432	0.499969 4	1.8, 1.98	0.852353	0.426150 06	0.7647675 3	0.9580483 2	0.2225139 8	0.3793739 51	0.6018540 98	0.8847284	0.8848387
7	0.681	0.5	1.8, 1.98	0.852353	0.426176 15	0.7786439	0.9598764 7	0.2233853 2	0.3762980 52	0.6034189 41	0.77904	0.7790535
8	0.85 4	0.500186	1.8, 1.98	0.852353	0.426334 68	0.7656075	0.9580417 4	0.2227252	0.3795396	0.6022648	0.940123	0.940356
9	0.10096	0.499685	1.8 ,1.98	0.852353	0.426149 3	0.7651125	0.9577162	0.2226610 6	0.3795952	0.6022256 3	0.2051443	0.2051702 5
10	0.32934	0.499997 9	1.8, 1.98	0.852353	0.426174 36	0.7649952 1	0.9577556 22	0.2226533 03	0.3795476 49	0.6022009 52	0.3520741	0.3520769

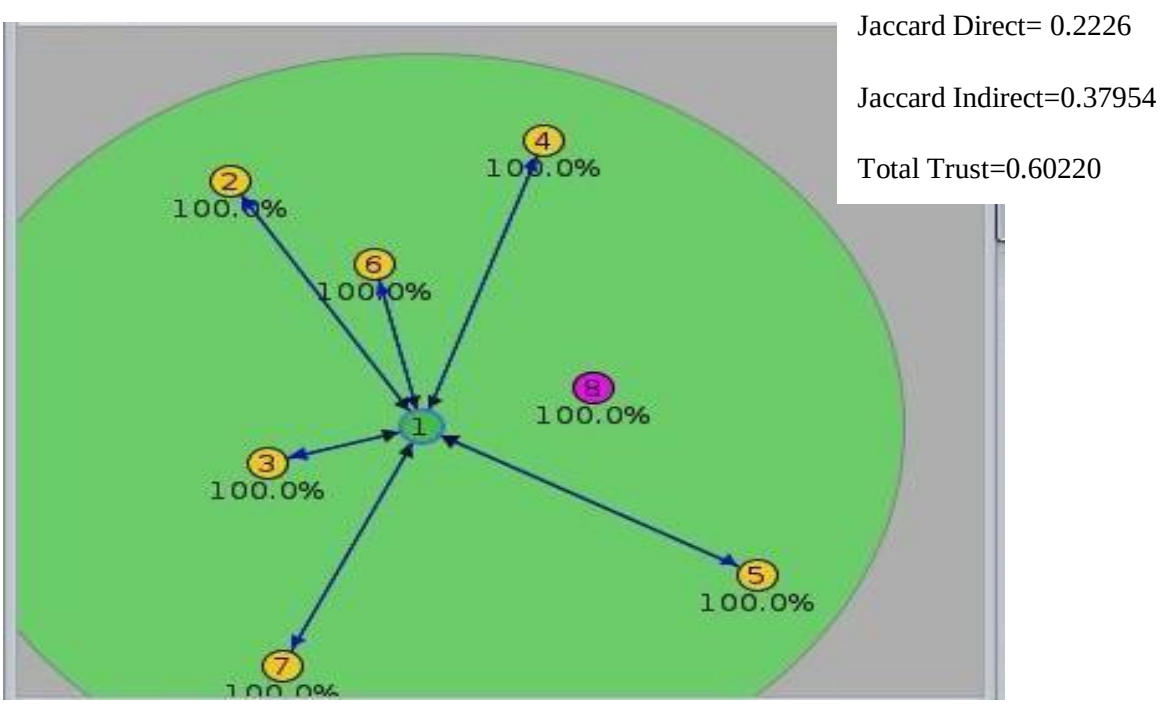

Figure 5.4 Contiki, Cooja fog network for badmouthing attack

5.3 Analysis and Findings

Let's look at how time-consuming our suggested model is. using n as the number of interactions that is maximum possible for a node, let's assume that our program will run n times. If the value of experience S_{ab} is required to be calculated n times (maximum), then the complexity is O(n), which is in the linear class. In order to select the recommenders based on servers, co work, and service similarity, three for-loops are used in the indirect trust T_{ab}^{r} calculation using equation 4.8, one of which will be called three times. In the worst-case situation, complexity is therefore O(n), which is again of the linear class. When calculating total trust using equation 4.10, the weights w and the indirect and direct trust values which are determined for each interaction in n using equation 4.12bare taken into account. The process takes O. (n). Being in the linear complexity class O(n), it is clear that our technique reduces the overhead associated with trust computation. As a result, it is more effective and economical, making it ideal for time-sensitive real-time applications.

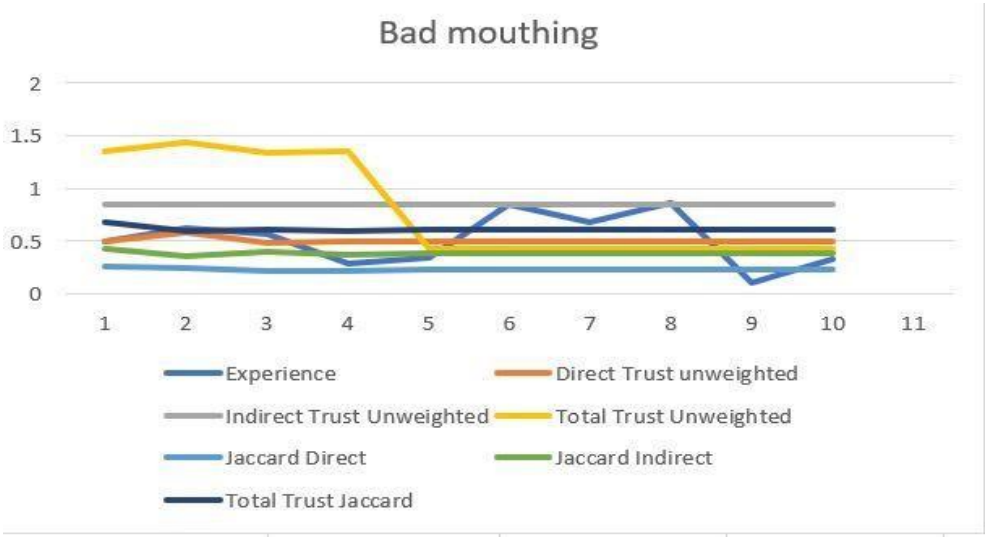

Figure 5.5 Badmouthing Attack with Adaptive weighting

As shown in Fig. 5.4, our model's adaptive weighting strategy outperforms statically allocated weights. Based on prior performance, it effectively raises and lowers trust scores without favoring direct or indirect trust value. Dynamic weight helps avoid bad mouthing attack by limiting the malicious node's involvement in trust and preventing it from negatively disrupting the model. When past trust values are used in later iterations and malicious behavior is eventually observed, a node that engages in bad mouthing attack can be identified using our adaptive weights and removed from the network.

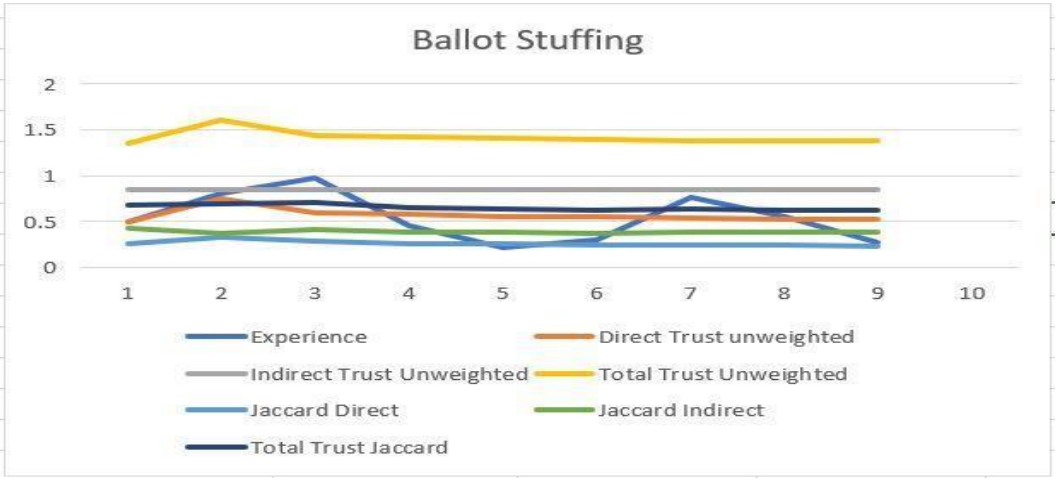

Figure 5.6 Ballot stuffing Attack using Adaptive Weighting

In fig 5.5 our model, we use a dynamic value of w that offers the adaptability required to stop the network from engaging in ballot-stuffing attack. Monitoring a node's behavior over time and removing it from the only recommendations from coworker have been taken into account in our model and recommendations are taken into account A node's indicated trust score, which is derived from a number of nodes, will require a large number of malicious nodes to lower it., similarity based filtering also aids in preventing badmouthing and ballot stuffing attacks.

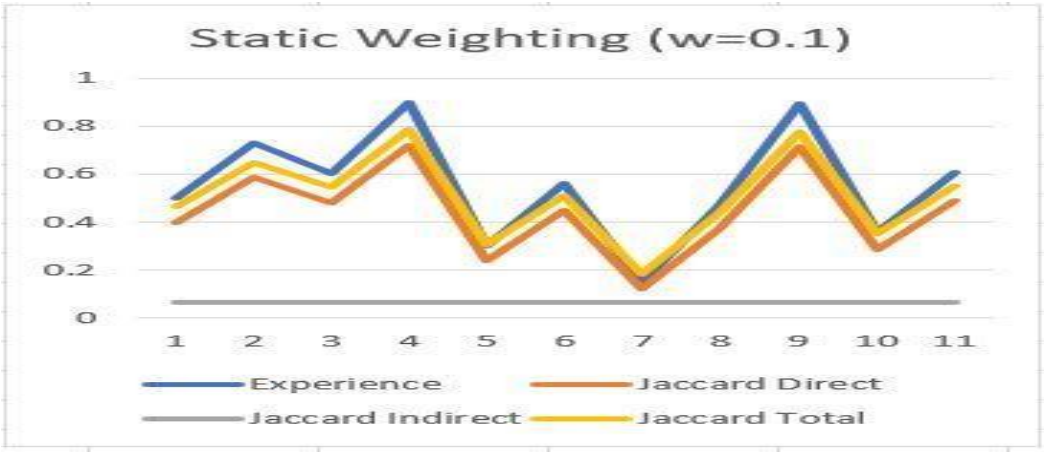

Fig 5.7 Static Weight W=0.1

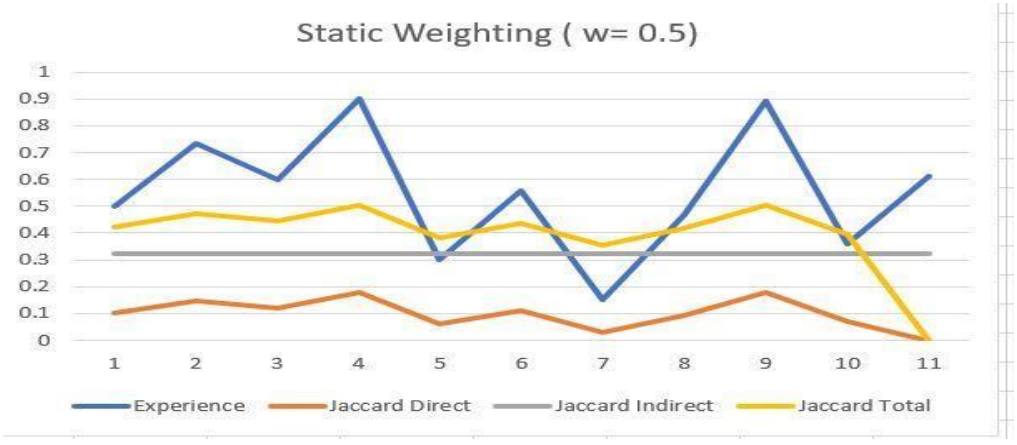

Fig 5.8 Static Weight W=0.5

Figures 5.6 and 5.7 show what happens when direct trust is given static weights of 0.1 and 0.5, respectively. We can see that, when employing static weights, our model displays trust

partiality; that is, when w = 0.5, the whole trust value is nearer to that of direct trust, as depicted in the diagram of the Jaccard trust calculation made with static weights in Fig. 5.4. Because it has been given a higher value of weight, total trust is more comparable to experience and direct trust. Similar to this, we can see that overall trust tends to be more indirect when w = 0.1.

5.4 Comparative Analysis

The suggested model is a two-way trust strategy; prior to connecting, both communicating nodes must confirm one another. Moreover, SQT is resistant to additional assaults like:

1. Bad-mouthing attack (BMA) as it only considers recommendations from trusted neighbors.
2. Ballot-stuffing attack (BSA) due to weighted recommendations.

Table 5.2 provides a thorough comparison of the models and 5.3 provides comparison of parameters used.

Table 5.2 Comparative Analysis of Fog with Proposed Model

Contribution	FGCS	CTR	Proposed Model
Distributed approach	✓	×	✓
Resilient against Bad mouthing attack	×	×	✓
Resilient against ballot stuffing attack	×	×	✓
Context dependent	✓	✓	✓
Two-way trust approach	×	✓	✓

Table 5.3 Comparative Analysis of Parameters

Contribution	**Indirect Feedback**	**Server Similarity**	**Co-work Similarity**	**Location Similarity**	**Packet delivery Ratio**
FGCS	✓	×	×	×	×
CTR	×	×	×	×	×
MSMN	✓	×	×	×	✓
Proposed Model	✓	✓	✓	✓	✓

Chapter 6

Conclusion and future work

Based on fog computing for smart medical systems, we have established an adaptive, context-dependent trust solution in this study. To identify only those recommenders whose context aligns with the inquiring node, we built a filtering approach based on Jaccard similarity metrics. We've analyzed effects of the attacks like bad mouthing and ballot stuffing in the model. We took into account three social similarity metrics, including: Server, co work and Service Similarity. Additionally, utilizing the Entropy theory, we created an adaptive weighting method that allows every node of fog to alter the weight of indirect and direct trust depending on historical data, minimizing the bias caused by trust. We've shown how employing static weighting methods compares to our dynamic weighting mechanism, demonstrating how the former exhibits skewed performance towards specific trust value while the latter does not. Additionally, the latter helps guard against numerous trust-related assaults. Due to the fact that it is of linear class O, our suggested model has the benefit of being very effective in terms of time complexity (n). In the future, we want to practically recreate and analyses the risk model and show how our trust model responds to diverse situations and different other types of attacks. We also want to examine how indirect suggestions are affected by various similarity measures, such as Pearson, Euclidean distance, etc. More research can be done on the use of blockchain to model trust in fogs.

Bibliography

[1] A. M. Rahmani, T. N. Gia, B. Negash, A. Anzanpour, I. Azimi, M. Jiang, and P. Liljeberg, "Exploiting smart e-health gateways at the edge of healthcare internet-of-things: A fog computing approach," Future Generation Computer Systems, vol. 78, pp. 641– 658, 2018.

[2] P. Verma and S. K. Sood, "Fog assisted-iot enabled patient health monitoring in smart homes," IEEE Internet of Things Journal, vol. 5, no. 3, pp. 1789–1796, 2018.

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

[4] Y. Hussain, H. Zhiqiu, M. A. Akbar, A. Alsanad, A. A.-A. Alsanad, A. Nawaz, I. A. Khan, and Z. U. Khan, "Context-aware trust and reputation model for fog-based iot," IEEE Access, vol. 8, pp. 31622–31632, 2020.

[5] 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), pp. 1–7, IEEE, 2013

[6] "What Is Edge Computing?" Available at: https://www.cisco.com/c/ en/us/solutions/computing/what-is-edge-computing.html# ˜revenue-opportunities.

[7].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), pp. 9–15, IEEE, 2017.

[8] "IoT Growth Demands Rethink of Long-Term Storage Strategies, says IDC." Available at: https://www.idc.com/getdoc.jsp?containerId= prAP46737220.

[9] A. A. Mutlag, M. K. Abd Ghani, N. a. Arunkumar, M. A. Mohammed, and O. Mohd, "Enabling technologies for fog computing in healthcare iot systems," Future Generation Computer Systems, vol. 90, pp. 62–78, 2019.

[10] D. Ravindran, “Fog computing: An efficient platform for the cloud-resource management,” Journal of Emerging Technologies and Innovative Research, 2019.

[11] R. Mahmud, F. L. Koch, and R. Buyya, “Cloud-fog interoperability in iot-enabled healthcare solutions,” in Proceedings of the 19th international conference on distributed computing and networking, pp. 1–10, 2018.

[12] S. A. Soleymani, A. H. Abdullah, M. Zareei, M. H. Anisi, C. Vargas-Rosales, M. K. Khan, and S. Goudarzi, “A secure trust model based on fuzzy logic in vehicular ad hoc networks with fog computing,” IEEE Access, vol. 5, pp. 15619–15629, 2017.

[13] “What is the Internet of Things (IoT)?.” Available at: https://www.oracle. com/internet-of-things/what-is-iot/.

[14] F. A. Kraemer, A. E. Braten, N. Tamkittikhun, and D. Palma, “Fog computing in healthcare–a review and discussion,” IEEE Access, vol. 5, pp. 9206–9222, 2017.

[15] S. Sagar, A. Mahmood, J. Kumar, and Q. Z. Sheng, “A time-aware similarity-based trust computational model for social internet of things,” in GLOBECOM 2020-2020 IEEE Global Communications Conference, pp. 1–6, IEEE, 2020.

[16] A. Altaf, H. Abbas, F. Iqbal, and A. Derhab, “Trust models of internet of smart things: A survey, open issues, and future directions,” Journal of Network and Computer Applications, vol. 137, pp. 93–111, 2019.

[17] “ OpenFog - OPC Foundation.” Available at: https://opcfoundation.org/ marketscollaboration/openfog/.

[18] R. Verma and S. Chandra, “A systematic survey on fog steered iot: Architecture, prevalent threats and trust models,” International Journal of Wireless Information Networks, vol. 28, no. 1, pp. 116–133, 2021.

[19] “Industry IoT Consortium.” Available at: https://www.iiconsortium.org/ index.htm.

[20] R. Chen, J. Guo, and F. Bao, “Trust management for soa-based iot and its application to service composition,” IEEE Transactions on Services Computing, vol. 9, no. 3, pp. 482– 495, 2014.

[21] A. Josang and R. Ismail, “The beta reputation system,” in Proceedings of the 15th bled electronic commerce conference, vol. 5, pp. 2502–2511, Citeseer, 2002.

[22] T. S. Dybedokken, “Trust management in fog computing,” Master’s thesis, NTNU, 2017.

[23] A. Kumari, S. Tanwar, S. Tyagi, and N. Kumar, “Fog computing for healthcare 4.0 environment: Opportunities and challenges,” Computers & Electrical Engineering, vol. 72, pp. 1–13, 2018.

[24] J. Guo and R. Chen, “A classification of trust computation models for service-oriented internet of things systems,” in 2015 IEEE International Conference on Services Computing, pp. 324–331, IEEE, 2015.

[25] A. A.-N. Patwary, A. Fu, R. K. Naha, S. K. Battula, S. Garg, M. A. K. Patwary, and E. Aghasian, “Authentication, access control, privacy, threats and trust management towards securing fog computing environments: A review,” arXiv preprint arXiv:2003.00395, 2020.

[26] A. Altaf, H. Abbas, F. Iqbal, M. M. Z. M. Khan, and M. Daneshmand, “Robust, secure, and adaptive trust-oriented service selection in iot-based smart buildings,” IEEE Internet of Things Journal, vol. 8, no. 9, pp. 7497–7509, 2020

[27] J.-H. Cho, A. Swami, and R. Chen, “A survey on trust management for mobile ad hoc networks,” IEEE communications surveys & tutorials, vol. 13, no. 4, pp. 562–583, 2010.

[28] 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, pp. 1–10, Citeseer, 2014.

[29] D. Chen, G. Chang, D. Sun, J. Li, J. Jia, and X. Wang, “Trm-iot: A trust management model based on fuzzy reputation for internet of things,” Computer Science and Information Systems, vol. 8, no. 4, pp. 1207–1228, 2011.

[30] E. Alemneh, S.-M. Senouci, P. Brunet, and T. Tegegne, “A two-way trust management system for fog computing,” Future Generation Computer Systems, vol. 106, pp. 206– 220, 2020.

[31] S. Prabhdeep and K. Rajbir, “Design and develop quality of service framework using fog computing for smart city applications,” International Journal of Innovative Technology and Exploring Engineering (IJITEE), vol. 9, no. 1S, 2019.

[32] A. Altaf, H. Abbas, F. Iqbal, M. M. Z. M. Khan, A. Rauf, and T. Kanwal, “Mitigating service-oriented attacks using context-based trust for smart cities in iot networks,” Journal of Systems Architecture, vol. 115, p. 102028, 2021.

[33] M. Apte, S. Kelkar, A. Dorge, S. Deshpande, P. Bomble, and A. Dhamankar, “Gateway based trust management system for internet of things,” REVISTA GEINTECGESTAO INOVACAO E TECNOLOGIAS, vol. 11, no. 4, pp. 4750–4763, 2021.

[34] M. Zineddine, “A novel trust model for fog computing using fuzzy neural networks and weighted weakest link,” Information & Computer Security, 2020.

[35]] 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. s

[36] M. Al-Khafajiy, T. Baker, M. Asim, Z. Guo, R. Ranjan, A. Longo, D. Puthal, and M. Taylor, “Comitment: A fog computing trust management approach,” Journal of Parallel and Distributed Computing, vol. 137, pp. 1–16, 2020.

[37] F. H. Rahman, T.-W. Au, S. S. Newaz, W. S. Suhaili, and G. M. Lee, “Find my trustworthy fogs: A fuzzy-based trust evaluation framework,” Future Generation Computer Systems, vol. 109, pp. 562–572, 2020.

[38] G. Rathee, R. Sandhu, H. Saini, M. Sivaram, and V. Dhasarathan, “A trust computed framework for iot devices and fog computing environment,” Wireless Networks, vol. 26, no. 4, pp. 2339–2351, 2020.

[39] S. O. Ogundoyin and I. A. Kamil, “A trust management system for fog computing services,” Internet of Things, vol. 14, p. 100382, 2021.

[40] 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), pp. 600–605, IEEE, 2015.

[42] V. B. Reddy, A. Negi, S. Venkataraman, and V. R. Venkataraman, “A similarity based trust model to mitigate badmouthing attacks in internet of things (iot),” in 2019 IEEE 5th World Forum on Internet of Things (WF-IoT), pp. 278–282, IEEE, 2019.

[43] Y. Hussain and Z. Huang, “Trfiot: Trust and reputation model for fog-based iot,” in International conference on cloud computing and security, pp. 187–198, Springer, 2018.

[44] 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,” Future Generation Computer Systems, vol. 109, pp. 573–582, 2020.

[45] L. Atzori, A. Iera, and G. Morabito, “Siot: Giving a social structure to the internet of things,” IEEE communications letters, vol. 15, no. 11, pp. 1193–1195, 2011.

[46] A. M. Kowshalya and M. 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.

[47] J. Al Muhtadi, R. A. Alamri, F. A. Khan, and K. Saleem, “Subjective logic-based trust model for fog computing,” Computer Communications, vol. 178, pp. 221–233, 2021.

[48] L. Zahrotun, “Comparison jaccard similarity, cosine similarity and combined both of the data clustering with shared nearest neighbor method,” Computer Engineering and Applications Journal, vol. 5, no. 1, pp. 11–18, 2016.

[49] O. B. Abderrahim, M. H. Elhedhili, 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), pp. 1903–1908, IEEE, 2017.

[50] U. Jayasinghe, H.-W. Lee, and G. M. Lee, “A computational model to evaluate honesty in social internet of things,” in Proceedings of the symposium on applied computing, pp. 1830–1835, 2017.

[51] A. M. Ali-Eldin, “A cloud-based trust computing model for the social internet of things,” in 2021 International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC), pp. 161–165, IEEE, 2021.

[52] S. Che, R. Feng, X. Liang, and X. Wang, “A lightweight trust management based on bayesian and entropy for wireless sensor networks,” Security and Communication Networks, vol. 8, no. 2, pp. 168–175, 2015.

[53] Cloud-Fog Interoperability in IoT-enabled Healthcare Solutions

[54] Y. Winnie, E. Umamaheswari, and D. Ajay, "Enhancing data security in iot healthcare services using fog computing," in 2018 International Conference on Recent Trends in Advance Computing (ICRTAC), pp. 200–205, IEEE, 2018.

[55] "NG, The OS for Next Generation IOT devices." Available at: https://www. contiking.org/.

[56] Fog computing for Healthcare 4.0 environment: Opportunities and challenges

[57] Towards Trust-aware Health Monitoring Body Area Sensor Networks

[58] A lightweight replay attack detection framework for battery depended IoT devices designed for healthcare

[59] MITIGATING IOTATTACKS IN SMART MEDICAL NETWORKS USING ENHANCED DIRICHLET BASED ALGORITHM FOR TRUST MANAGEMENT SYSTEM

[60] A Security Awareness and Protection System for 5G Smart Healthcare Based on Zero-Trust Architecture

[61] Security, Privacy and Trust in IoMT Enabled Smart Healthcare System: A Systematic Review of Current and Future Trends**.**

[62] Robust, Secure and Adaptive Trust-Oriented Service Selection in IoT-Based Smart Buildings Ayesha Altaf, Haider Abbas, Senior Member, IEEE, Faiza Iqbal, Malik Muhammad Zaki Murtaza Khan and Mahmoud Daneshmand Life Senior Member, IEEE

[63] A Novel Hybrid Trustworthy Decentralized Authentication and Data Preservation Model for Digital Healthcare IoT Based CPS

[64] Analysis of factors affecting IoT-based smart hospital design

[65] A Survey on IoT Smart Healthcare: Emerging Technologies, Applications, Challenges, and Future Trends