CONTEXT ORIENTED TRUST MANAGEMENT SOLUTION FOR FOG-BASED IOT

By

Sarawish Altaf

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 Information Security

September2023

THESIS ACCEPTANCE CERTIFICATE

Certified that final copy of MS Thesis written by **Sarawish Altaf**, Registration No. **00000361860**, of **Military College of Signals** has been vetted by undersigned, found complete in all respects as per NUST Statutes/Regulations/MS Policy, is free of plagiarism, errors, and mistakes and is accepted as partial fulfillment for award of MS degree. It is further certified that necessary amendments as pointed out by GEC members and local evaluators of the scholar have also been incorporated in the said thesis.

Signature: _____

Name of Supervisor: Assoc. Prof Dr. Mian M Waseem Iqbal

Date: _____

Signature (HOD): _____

HoD
Information Security
Military College of Sigs

Date: _____

Signature (Dean/Principal) _____

Brig
Dean, MCS (NUST)
(Asif Masood, Phd)

Date: 21/9/23

ABSTRACT

Fog computing has transformed the Internet of Things by delivering computing facilities of the Cloud at the edge of the network. Compared to Cloud, Fog offers various advantages, including reduced latency and enhanced reliability. The Fog architecture is very dynamic in nature and its topology continuously changes as new objects enter and leave the network. This opens the door for malicious objects to infiltrate and disrupt the network. As the Fog-based IoT network expands rapidly, effective security in such a scenario becomes critical. Trust-based approaches have recently piqued the academic community's interest to mitigate the security concerns of the Fog-based IoT. Trust among objects is vital because it allows them to distinguish and govern information exchange in the network. However, in the Fog-based IoT network context, which is a fundamental aspect of trust, has not been given sufficient attention yet. Thus, we propose in this paper a context-oriented, multi-source, similarity measures-based trust management system for Fog-based IoT. Our trust management model effectively incorporates context in the trust calculation process with respect to both servers and recommenders. By utilizing Bayesian inference and similarity scores to filter contextually similar service providers and recommenders respectively, our proposed trust management system is effective in calculating a context aware trust score.

DEDICATION

In loving memory of my mother, whose enduring wisdom and love continue to inspire me daily. This thesis stands as a tribute to her profound influence on my journey.

ACKNOWLEDGEMENTS

I am grateful to God Almighty for granting me the strength and passion to accomplish this thesis. His mercy and benevolence have been essential to its completion.

I would like to extend my heartfelt gratitude to my dedicated supervisor, Assoc. Prof. Dr. Mian Muhammad Waseem Iqbal. His unwavering guidance and support have been invaluable in shaping this thesis into a reality.

I also want to express my appreciation to my family and friends for their unwavering encouragement and belief in my abilities.

Lastly, I extend my thanks to all those who have played a part in this endeavor, no matter how small, for your contributions have been instrumental in bringing this thesis to fruition.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

InternetofThings	IoT
QualityofService	QoS
ServiceProviding Node	SP
ServiceRequesting Node	SR
Recommenders	R
Trust Management System	TMS
List of Services	L_S
List of Service Providers	L_{SP}
List of Social Contacts	L_C
List of Past Experiences	L_T
Alpha (used in Bayesian Inference)	α
Beta (used in Bayesian Inference)	β
Time Difference	Δt
Direct Trust	$T^d_{a,b}$
Indirect Trust	$T^r_{a,b}$
Total Trust	$T^t_{a,b}$
PacketDeliveryRatio	PDR
Network Simulator 2	NS2

INTRODUCTION

The Internet of Things (IoT) has rapidly swept our lives and converted our world into a cyber network. By utilizing the interconnection of physical objects to the internet, IoT enables the delivery of real-time information anywhere, at any time, with a minimum of human intervention. Cloud has traditionally been used by heterogeneous and resource-constraint IoT devices for its data processing and analytics. While cloud computing enabled the IoT to offer new advancements, it also raised various issues regarding security. As the number of IoT-integrated devices increases rapidly, it is necessary to examine the bandwidth restrictions, latency, and reliability challenges that arise with the utilization of the Cloud. As a result of these issues, Fog computing architecture was introduced in IoT. The Fog computing architecture comprises several devices known as nodes that are positioned at the network's edge as depicted in Figure 1.1.

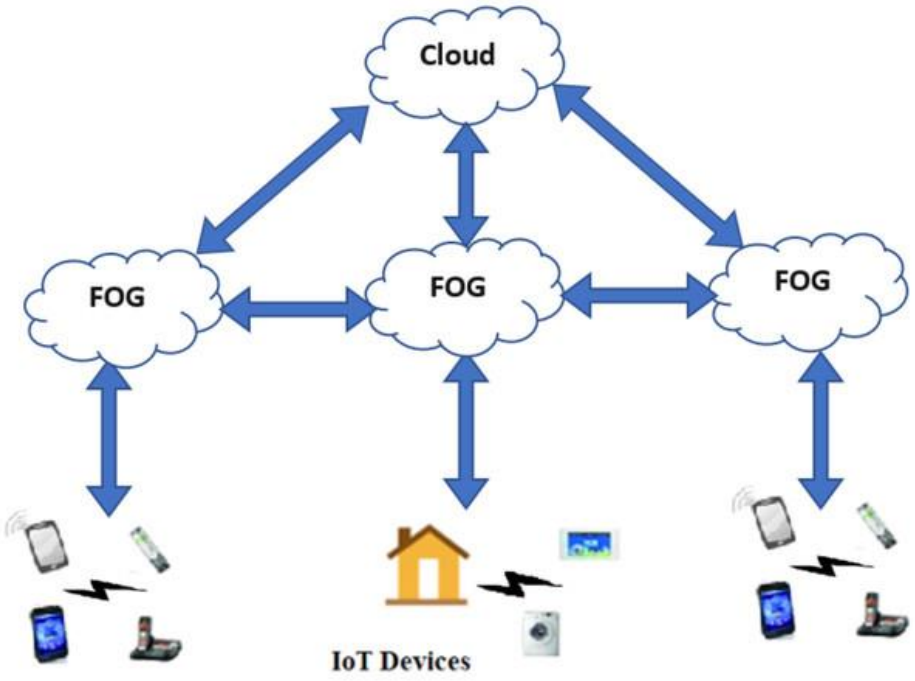

Figure 0.1: Fog-based IoT Structure

1.1Motivation

By delivering Cloud services at the network's edge, Fog computing has transformed Cloud-based IoT. However, several requirements and difficulties in Fog computing remain unresolved.Ideally, IoT and Fog nodes should be able to form autonomous collaborations with one another. Also, the exchange of service should be accompanied by a sense of confidence and assurance regarding the quality of service as well as the behaviour of nodes

involved in service exchange [1][2][3]. Because of the dynamic nature of a Fog-based IoT network, it is vulnerable to a malicious node entering and disrupting the network. To combat these issues, trust management solutions for effective and safe communication between IoT devices and Fog nodes can be deployed. Given the ever-changing trust dynamics along with the significance of data generated by fog-based IoT networks, a noteworthy gap in the research was identified. This gap is particularly evident when trust becomes a pivotal evaluation factor for a node's security assessment based on which the assessor will get involved in providing or receiving services. Therefore, our objective is to introduce a dynamic and adaptive context-oriented trust management system designed specifically for fog-based IoT environments.

1.2Scope and Objectives

A context-oriented trust management system for fog-based internet of things that manages or assesses trustworthiness of different entities including devices, applications, end-users, etc. The primary goals of this research involve studying the existing trust management systems employed within the realms of the Internet of Things (IoT) and Fog computing and to propose a context-oriented, multi-score, similarity measure-based trust management system for fog-based Internet of Things. The trust management system employs context in the trust calculation with respect to both recommenders and servers. The scope gravitates around utilizing Bayesian inference and similarity scores to filter similar service providers and recommenders.

1.3Contributions

The contributions made through this researchare as follows:

- Designed a context-oriented trust management system, based on Bayesian Inference and similarity measures for Fog-based IoT.
- Demonstrated how the proposed solution accurately evaluates a service provider's abilities based on the specific service they are providing.
- Evaluated different similarity measures such as Jaccard, Cosine, Euclidean distance and Pearson Correlation with 50 different data sets, and analyzed the differences in their results.
- Demonstrated that conventional Pearson Correlation method is not suitable for binary data. Assessed a modified Pearson formula specifically designed for binary data.

- Demonstrated the use of information gain for effective calculation of adaptive weights for direct and indirect trust.

1.4Thesis Outline

Thisthesisisdividedintosixchapters:

- Chapter1: Thischaptercontainsintroduction,scope,objectives,andthecontributions we have made in this thesis research.
- Chapter 2:In this chapter,we briefly discuss IoT, trust and trust management system indetail.
- Chapter 3: An overview and comparison of the trust management models as proposed by several researchers.
- Chapter 4: Proposed solution is discussed in this chapter including mathematical modelling in detail.
- Chapter5: Thischapterdiscussesevaluation and analysis of the proposed solution's results.
- Chapter6:Thischapterconcludestheresearchandfutureworkisproposed.

PRELIMINARIES

This chapter will give an overview of IoT, cloud, fog, trust and will explore trust management system in detail including its various components and dimensions.

2.1 Internet of Things (IoT)

The concept of the Internet of Things (IoT) represents a revolutionary shift in the field of information and communication technology. It revolves around the widespread connectivity of various physical devices, sensors, and objects, each equipped with computational functionalities and linked to the global network. This array of devices encompasses a broad spectrum, encompassing everyday consumer gadgets as well as specialized industrial equipment and environmental monitoring sensors.

IoT leverages the power of network connectivity and data exchange to facilitate real-time communication and collaboration among these devices, enabling them to collect, transmit, and receive data autonomously. This data can include environmental information, operational status, user interactions, and more. The seamless integration of IoT devices into our surroundings and daily lives has ushered in a new era of data-driven decision-making and automation.

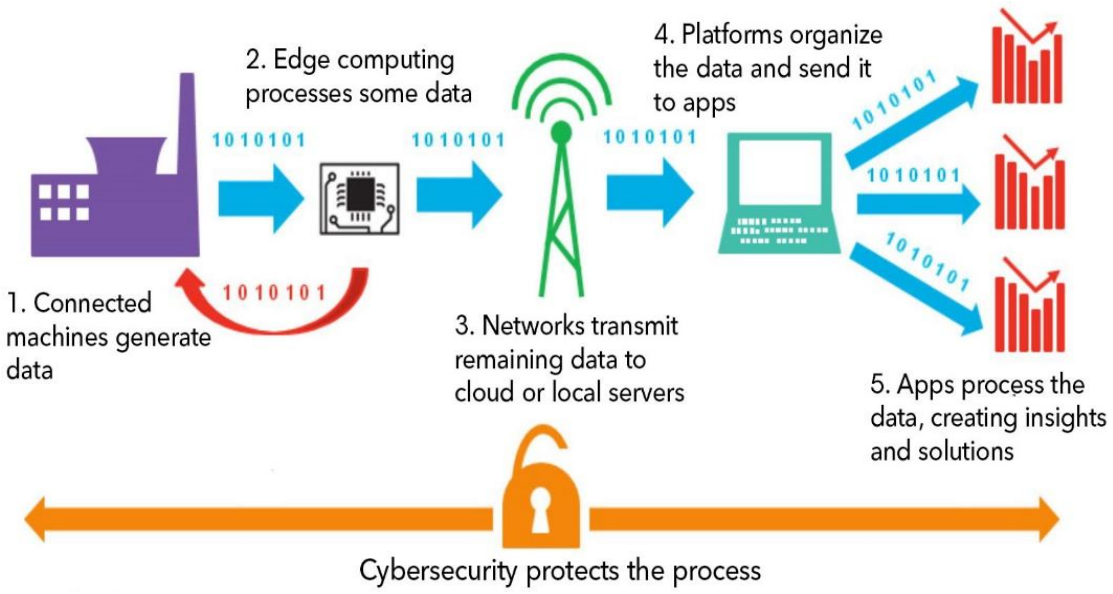

Figure 0.1: IoT Network

Within this interconnected ecosystem, IoT has found applications across numerous domains, including but not limited to smart cities, healthcare, agriculture, transportation, and industrial

automation. It offers the potential to enhance efficiency, improve resource utilization, optimize processes, and provide valuable insights for decision-makers.

The concept and use of trust in IoT are of paramount importance. Trust mechanisms and protocols are integral to establishing secure and reliable communication among IoT devices. Ensuring the authenticity, integrity, and confidentiality of data transmitted in IoT networks is crucial, particularly in applications where safety, privacy, and critical decision-making are involved. Therefore, the study of trust in IoT is not only a significant research focus but also an essential consideration for the development and deployment of IoT solutions.

2.2 Trust

In the most basic terms, trust can be defined as a subjective value or belief that measures the relationship between two nodes and keeps the communication between them reliable. Trust can be characterized as:

- Context Sensitive: Trust is measured subject to capability in the context of an existing relationship.
- Subjective: Trust depends on how an agent perceives the behavior of a subject
- Unidirectional: Agent’s trust is based on knowledge about the subject and is not necessarily reciprocated.
- Non-transitive: If a Node X trusts Node Y and Node Y trusts Node Z, then Node X does not necessarily trust Node Z.

In Fog-based IoT, trust-based security solutions help in establishing a user's trustworthiness score and detect malicious objects. Objects in IoT networks assess Fog nodes and make decisions about future engagementbased on their past interactions and current assessment of the Fog node. This is called Direct Trust. The other aspect of trust is Indirect or Recommended Trust [4] in which objects share their opinion regarding trust in neighboring objects as recommendations. However, false recommendations might be provided by a malicious node in a bad-mouthing attack to give low trust values to legitimate nodes. A collusion attack occurs when hostile nodes collaborate and transmit bad recommendations for a specific target node. To counteract these types of attacks, recommendations must be weighted based on their similarity to the recommender. The similarity method enables the

correlation of recommendations to find a similarity score which helps to filter credible and non-credible recommending nodes.

2.3 IoT Trust Management System

Trust management is the process used to measure, propagate, and update the assurance based on trust score between devices. The main idea of trust management is to enable two nodes in forming autonomous collaboration and building trustworthy interactions. The several dimensions of trust used for calculating the trust as discussed in the literature are shown in Figure 2.2 and elaborated in the paragraphs that follow.

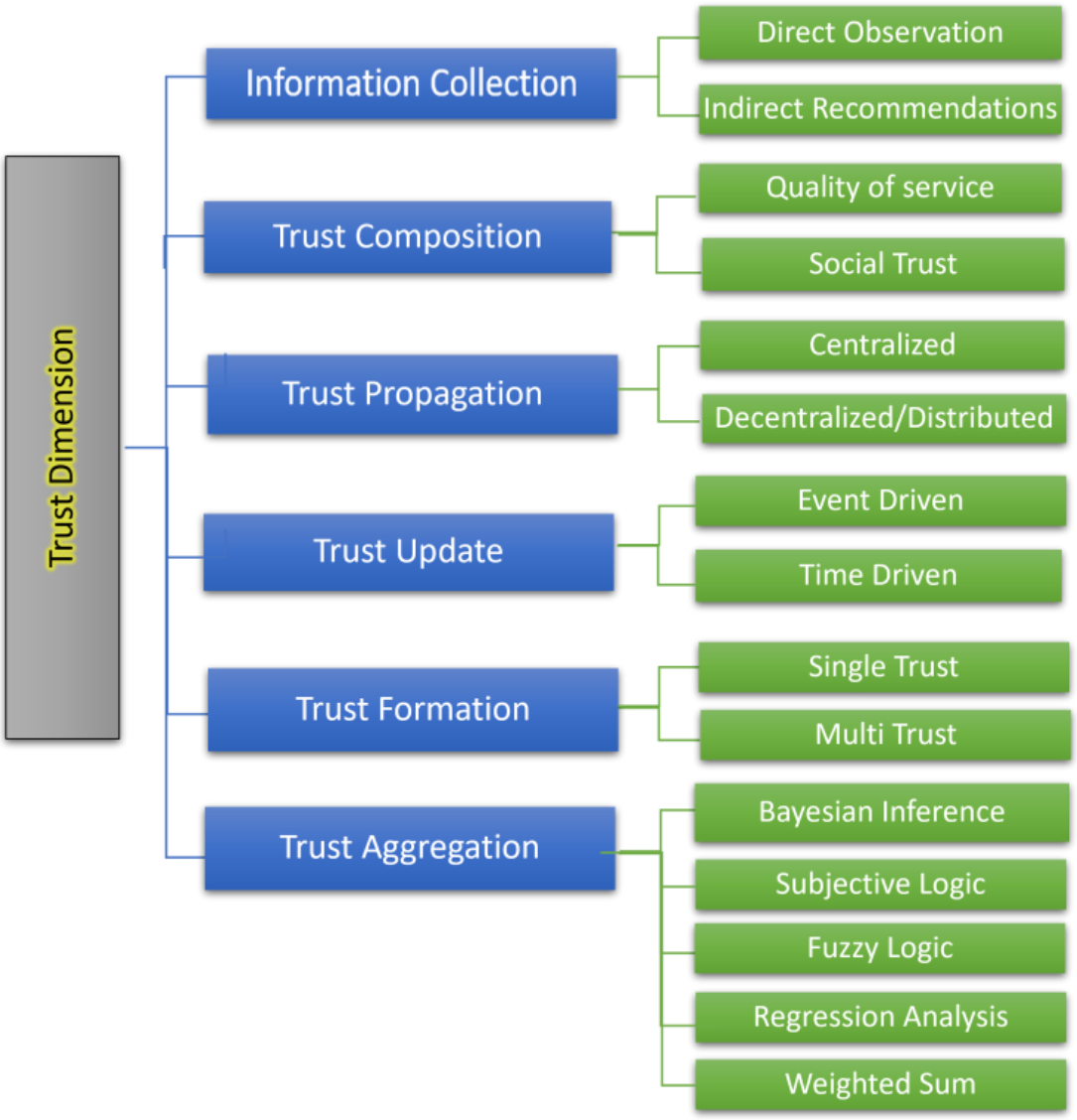

Figure 0.2: Multiple Trust Dimensions

2.3.1 Information Collection

Information collection is carried out either directly via observation or indirectly via recommendations as shown in Figure 2.3. In the direct method, trust is evaluated directly based on one-to-one interaction through communication, observation, or social behaviour. A node uses its observations or experience with a particular node to gather information and calculate a trust value for that node. For example, depending on a node's experience of the service it receives from a particular node, it either comes to trust that node or not. This is called direct trust.

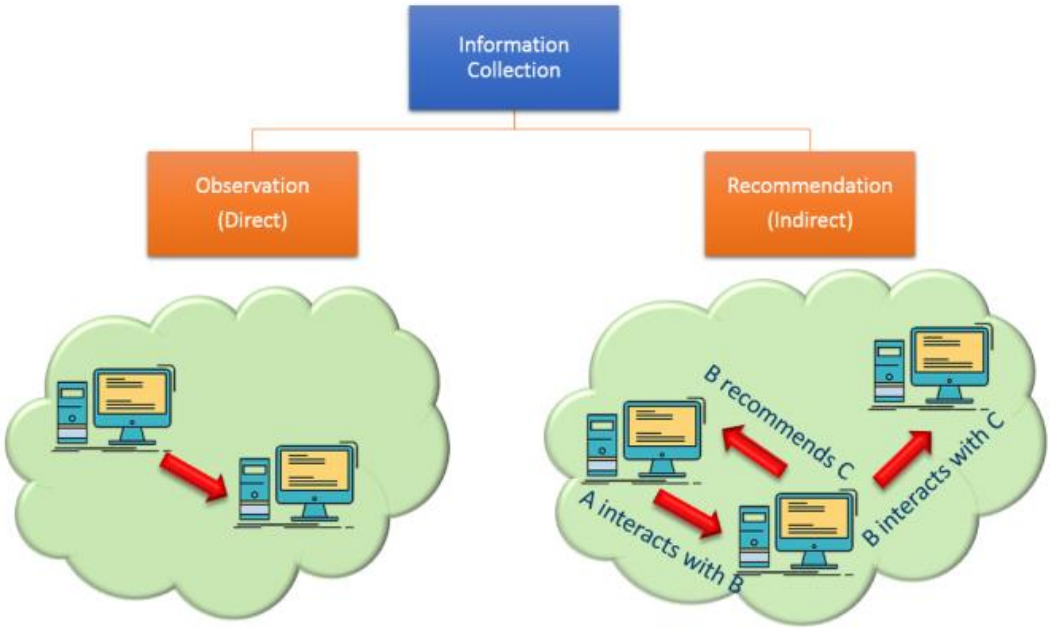

Figure 0.3:Information Collection Component of Trust Calculation

In the indirect method, a node does not have a direct interaction or experience with a particular node, so to calculate that node's trust score it takes information or trust recommendations from other nodes which had a direct interaction or experience with that node. Hence, objects exchange information about their trust in the neighbouring objects with each other. These recommendations are then accumulated by every node. This is called indirect trust. It is important to note here that both direct and indirect methods can also be utilized in combination for calculating trust [4][5]. Irrespective of the method used, the quality of service received from and social interaction with a node, are key components that will determine that node's trust score [6].

2.3.2 Trust Composition

The trust composition dimension determines the components to be included in the trust computation process. All trust components can be separated into two categories: quality of service (QoS) trust and social trust. The former relies on the object's belief in the QoS given, i.e., the degree of assurance that a node can provide the desired service. Performance variables may include throughput, latency, or errors, as well as availability, dependability, irregularity, and capacity. On the other hand, social relationships and interactions between the objects' owners establish the social trust component. Friendship, social contact, and community of interest (COI) similarities are all ways to quantify social similarity. When dealing with social trust, other variables can be considered in addition to the relationships between two objects, such as:

i. Centrality: How important an object is in a network, how many interactions it has with other objects.

ii. Credibility: How credible the information presented is.

iii. Social Similarity: How similar things are when they interact.

2.3.3Trust Propagation

The trust score is calculated either in a centralized manner or in a decentralized/distributed manner as depicted in Figure 2.4 below:

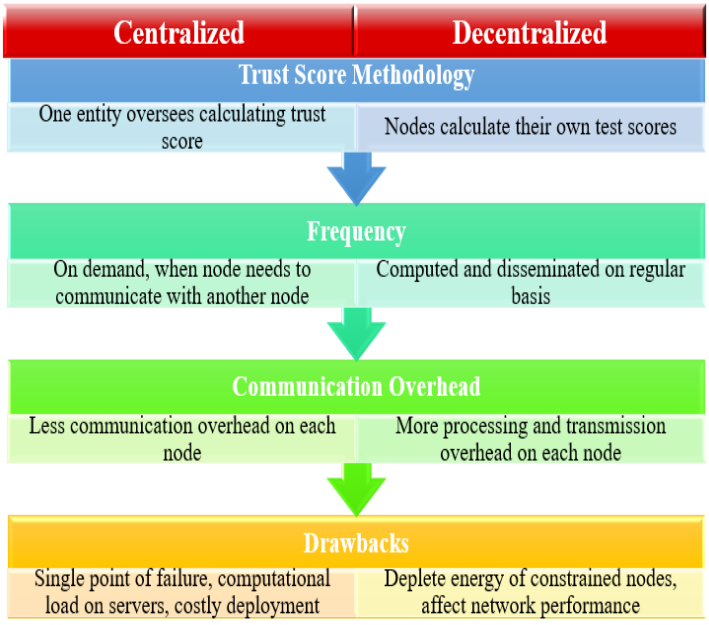

Figure 0.4: Trust Placement Strategy

2.3.4Trust Update

In trust models, updating the trust score regularly is critical. The following scenarios can be used to update the trust scores.

i. Event Driven: Trust is updated as soon as a new event occurs. The event could be the introduction of a new node, the commencement or termination of a transaction. When a node rates a transaction, the trust score for that node is updated.

ii. Time Driven: Rather than waiting for an event to occur, trust is updated regularly, after a pre-set time.

2.3.5Trust Formation

The basic objective of trust formation is to weigh trust properties based on their importance. Trust formation shows how to calculate trust from a collection of trust properties. Trust can be formed by considering only one trust property (single trust). The other method is to combine multiple trust properties (multi-trust). In single trust models as only one trust property is considered so it is usually the most significant one, namely the QoS. Because trust is viewed as a multidimensional, single-trust formation is one-sided. However, if the object's resources are restricted, it may be considered a viable solution because it requires less storage and computation. To compute an object's trustworthiness, several trust attributes are considered in multi-trust formation. There are two ways to accomplish this. The first option is to assign a threshold value to each trust property. Several trust properties are examined in this technique, but they are not consolidated into one. A threshold value is applied to each trust property. The threshold value shows the property's relevance in the application. If the value of one of the characteristics falls below the threshold, the entire object is considered untrustworthy. The second alternative is to unite the trust properties into a single value. A weighted sum can be utilized in this case. If one value of trust is more significant than another, it will be given a higher weight. The weights assigned can be set and altered depending on the context.

2.3.6 Trust Aggregation

After the trust properties have been defined, the network members' trust must be gathered. The trust aggregation specifies how to collect experiences from other objects in the network. Some object experiences may be more useful and accurate than others; the trust aggregation dimension accounts for this. This step involves combining trust values from one's own experiences as well as the experiences of others into a single value. In reputation systems, a

weighted sum can be utilized such that the users with the highest reputations have the greatest influence on the total trust value. Fuzzy logic, belief theory, bayesian inference, and regression analysis are other methods commonly employed for trust aggregation.

2.4 Cloud Computing

Cloud computing has been an evolving paradigm over thelastdecade.Themajoradvantageprovidedbyincorporating cloudcomputing isthe abilitytoperform computationalandstoragefunctionsfarfromtheenddevices.Forthese specific reasons,datacentersarebiltthatoffersttheprocessingand computationalcapabilitiesrequiredbytheusers. Cloud computing environment is suited for wireless sensorswhichcanbedeployedwitheasefordatacollectioninsuchanarchitecture.Afterdata collection, WiFi or any compatible network is used fordata transmission, synchronization and uploading data on theserver.A wide range ofcloud applications include real-time monitoring, predictive maintenance, remote control, and data-driven decision-making in various industries such as manufacturing, agriculture, healthcare, smart cities, and logistics. Additionally, cloud-based IoT platforms offer the scalability and security needed to support the growing ecosystem of interconnected devices and sensors, making them a critical component of the IoT landscape. Theexponentialincreaseinthebidirectionalflowofdatafromand towardscloudfromadiverseofsetofsensornodes,i.e.,bigdataposesachallengefortheservicesprovidedbythecloudintermsofdatatransmission,storageandcomputation.Cloudcomputinghasbeenbolsteringtheservicesprovidingby IoTdevicesovertheyears.However,duetotheevolvingrequirementsespeciallypertinenttoprivacyconcerns,latencyandresourceconstraintapplicationsmadeitdifficultforthecloudtomanage.Thisisbecauseofthedistanceinvolvedbetweencloudandenddevices.

2.5 Fog Computing

Due to the problems faced by the cloud, fog computing, a concept of bringing services near the end devices was proposed. Fog computing is not a replacement of cloud rather it augments the services provided by the cloud by reducing the burden of cloud in terms of local processing and computation. Fog computing is well suited for dependable and interactive services that require lower service response time. Fog computing improves performance of the overall system in the following ways:

- Latency in fog architecture is significantly reduced due to proximity of end devices to the fog nodes.

- As compared to cloud architecture, fog computing addresses the issue of privacy by locally analyzing sensitive data rather than processing user's data on a server not under user's control.
- Fog computing reduces bandwidth consumption by reducing the amount of data sent to the cloud. This is achieved by locally analyzing, processing, and compressing the data, thereby, sending a significantly reduced amount of data being sent to the cloud. Moreover, devices can receive answer requests from local nodes, thereby, bypassing the need to communicate with the cloud.
- Scalability is improved by the fog architecture by being customizable and reducing computation load from centralized sources and can be expanded according to the need of the application.
- Dependability in fog architecture is achieved by allowing multiple nodes in the network to provide same functionality. Moreover, computation takes place in the proximity of sensor nodes making the architecture less dependent on availability of internet connection various centralized sources.

Figure 2.5 below shows an architectural diagram of Fog-based IoT.

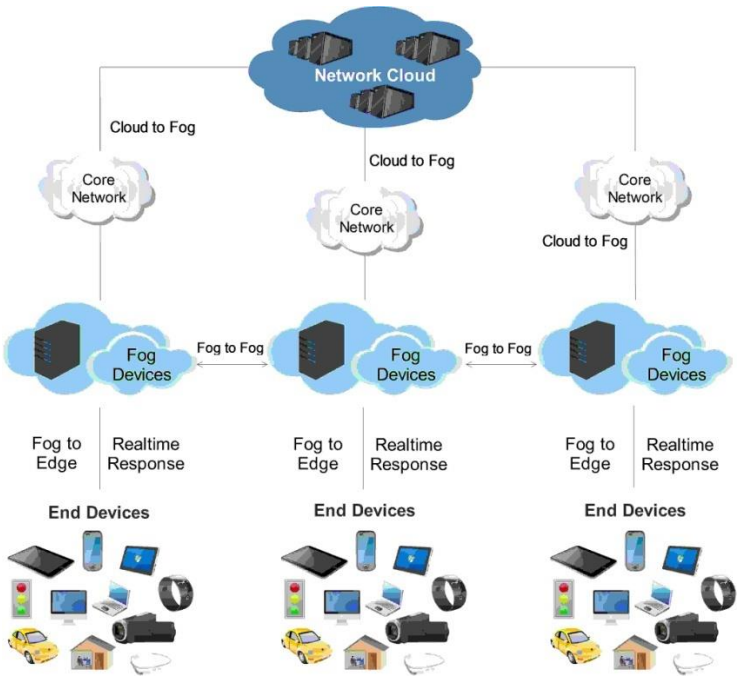

Figure 0.5: Fog-based IoT Network

Table 2.1 below presents a comprehensive comparison between cloud computing, fog computing, and edge computing. These three paradigms represent different tiers of computing infrastructure, each with distinct characteristics and use cases. This comparison helps understand the key differences and advantages of each approach in various applications and industries.

Table 0-1: General Comparison of the Cloud, Fog, and Edge Computing

CHARACTERISTICS	CLOUD	FOG	EDGE
Architecture	Centralized	Decentralized	Decentralized
Latency	Highest	Medium	Lowest
Scalability	High, easy to scale	Scalablewithin network	Hard to scale
Distance	High (Far from the edge)	Less (Close to the edge)	Low (At the edge)
Data analysis	Data at rest (permanent storage). Less time-sensitive data processing	Data is motion. Real-time data analysis. Decides to process locally or send to cloud	Data is motion. Real-time data analysis. Instant decision making
Processing power	High	Limited	Limited
Interoperability	High	High	Low
Bandwidth cost	Highest	Medium	Lowest
Benefits	Rich resources Scalability Easy Maintenance Cost efficiency	Low latency Better data control/ privacy Flexible storage system Connecting centralized and decentralized	No delays in data processing Real-time data analysis Low network traffic Reduced operating costs

Certainly, let's expand on the idea that fog computing is the future, the research in fog computing focuses on securing data and communication links, and that trust in fog computing, especially context-oriented trust, hasn't been explored enough:

2.6 Research Direction in Fog Computing

Fog computing is rapidly emerging as a pivotal technological advancement, particularly within the domains of the Internet of Things (IoT) and edge computing. Its significance lies

in its unique capability to bring computational resources closer to the origin of data, resulting in reduced latency, improved real-time processing, and enhanced overall efficiency of IoT systems. This proximity to edge devices and data sources positions fog computing as an incredibly promising solution across various industries. Looking ahead, we can envision fog computing playing a central and transformative role in facilitating innovative applications, such as autonomous vehicles, smart cities, industrial automation, and healthcare. By dispersing computational capabilities across fog nodes strategically placed at the network's periphery, organizations can tap into novel opportunities for data analysis, informed decision-making, and automation.

The research in fog computing has prominently concentrated on addressing the security challenges associated with the decentralized nature of edge and fog environments. Two key areas of focus have been securing data and communication links:

- Data Security: Fog computing research emphasizes safeguarding sensitive data generated and processed at the edge. This involves implementing encryption, access controls, data anonymization techniques, and secure data storage solutions. Researchers are also working on developing efficient data sharing and data provenance mechanisms to ensure data integrity and traceability.
- Communication Link Security: As fog nodes and edge devices communicate over potentially untrusted networks, securing these communication links is paramount. This involves the use of secure communication protocols (e.g., TLS/SSL), intrusion detection systems, and network segmentation to protect against eavesdropping, man-in-the-middle attacks, and unauthorized access.

While significant strides have been made in securing data and communication links in fog computing, trust management remains a challenging and evolving area of research. Trust is critical for ensuring the reliability, integrity, and security of the fog computing ecosystem. Notably, context-oriented trust, or the ability to assess trustworthiness based on specific contextual factors, is an aspect that warrants further exploration:

Fog computing environments are dynamic, and trust should not be assessed solely based on historical behavior or static criteria. Context-oriented trust takes into account real-time contextual information, such as device location, network conditions, and environmental factors, to make trust decisions. Research in this area seeks to develop adaptive trust models that can dynamically adjust trust levels based on the changing context, ensuring more

accurate and context-aware trust assessments. As fog computing becomes more pervasive, addressing trust challenges will be pivotal in realizing its full potential and ensuring the security and reliability of fog-based systems in diverse applications and industries.

LITERATURE REVIEW

This chapter discusses literature review in detail. It presents work done so far related to trust in IoT and Fog Computing.

3.1 Survey of Trust Models

In literature, trust management issues for Fog-based IoT have been addressed but very few have incorporated context in their proposed trust management solutions. For our literature review, we explored existing trust-related solutions in Fog [10], [11], [3], [14] as well as IoT [5], [6], [12] [13].

The literature review started with a survey paper [8]. Mohammadi et al. presented a systematic literature review through 59 published works of literature to ascend recommendations in IoT. Distributed trust model utilizing both direct observations (interactions) and recommendations was proposed by [8] as well as by Esubalew A. et al in [11]. [11]'s major contribution is its first-of-a-kind bi-directionality. QoS trust metrics are utilized by the model to calculate the service provider's trust score while social trust metrics are used to calculate the service requestor's trust score. Also, the model [11] was shown to be more effective and accurate than one-way trust management systems in service provider selection, had lower overhead and balanced load distribution. However, both [8] and [11] did not incorporate context in their trust calculations.

The researchers have investigated context in relation to trust in a variety of methods. Some researchers have developed trust management solutions in which the threshold value for an acceptable trust score varies depending on the application. As a result, distinct trust measurements for different applications are obtained, resulting in a context-aware trust score. Three such models were proposed in [6], [7] and [3].

In [6] Abouzar Arabsorkhi et al. propose an application-sensitive model based on the social world of humans that utilizes a ranking mechanism based on past experiences to prioritize multiple potential service providers. Selection is made if the node's trust score exceeds a predefined threshold and does not contradict the node's trustworthiness assumption. Carolina V. L. Mendoza et al. proposed in [7] a decentralized trust management scheme based on direct trust only. The model aims to keep the resource capabilities of a node as well as the context of what service is being requested in consideration by having a trust score that is

affected differently by lightweight services and more resource-intensive services. T. Dybedokken offers a proof of concept in [3] and proposes an application and environmental condition-sensitive, subjective logic-based trust management system. Though these models incorporate context to some extent, however, they do not include context regarding servers and/or recommenders.

Another method used by the researchers to incorporate context in their trust score is to utilize social networking notions in the IoT. Marche, et. al used this vision in [9] to propose a Machine-Learning based decentralised trust management model. By using social trust components, the model assigns each community member a trust level, and incorporates context in terms of social similarity. However, context with regard to the capability of the service provider is not considered by the model.

Direct trust based on the server's capability, location and the type of service provided by the server is used by Altaf et al. in [12]. Naive Bayesian classification method is used to filter and classify the direct interactions and past experiences of a user with different servers in various contexts. Though the model efficiently calculates a context-aware trust score, it limits itself by only depending on a direct trust score. In [13] the research was expanded by incorporating both direct and indirect trust. Context is calculated with regard to the recommenders which leads to the fourth approach used by the researchers. The potential service provider's credibility is calculated using recommendations from context-similar recommenders.

Almas A. also uses a similar approach in [10]. The model utilizes the Bayesian approach andCosine and Jaccard similarity measures to calculate a context-aware trust score. Recommenders were filtered based on social contact, service and server similarity so that only context-similar recommendations were taken into consideration. This also helped to screen out recommending nodes with malicious intent. A major contribution of the paper is the use of entropy theory to calculate dynamic weights that are used for total trust calculation. Like [13], its limitation is that does not incorporate context with regard to the server.

Trust models based on recommendation techniques are susceptible to badmouthing and collusion attacks. To counter these attacks a node's credibility should not be assessed based on direct trust. [5] Vijender Busi et al. propose an approach based on similarity to correlate recommendations and determine a node's credibility. The model employs context similarity at recommenders to filter out malicious nodes.

In [14] Y. Hussain et al. offer a context-aware trust evaluation model that utilizes multi-trust metrics for fog-based IoT. The proposed model utilizes an evaluation mechanism based on reputation to ascertain a user's trustworthiness. For the identification of malicious nodes, the model makes use of a monitor mode.

Table 3.1 displays a general comparison of the different trust management systems as proposed in the research papers we reviewed. Various researchers used various trust dimensions to develop their proposed trust management systems. These differences are highlighted in Table 3.2.

Table 0-1: General Comparison of the Proposed Trust Models

Reference #	Year of Publishing	Application Area		Trust Decay	Trust Direction		Context Awareness	Simulation Model	Application Developed	Computation Overhead/ Efficiency
		IoT	Fog		One-directional	Bi-directional				
[5]	2019	✓	×	×	✓	×	✓	NS2	×	Not mentioned in the paper
[6]	2016	✓	×	×	✓	×	×	×	×	Not mentioned in the paper
[10]	2022	×	✓	✓	×.	✓	✓	Contiki-NG Cooja	Java application developed	Low overhead and efficient-lies in linear complexity O(n).
[11]	2020	×	✓	×	×	✓	×	Java-based simulation tool developed		Low overhead and balanced load distribution
[3]	2017	✓	✓	×	×	✓	✓	×	×	Not mentioned in the paper
[12]	2019	✓	×	×	✓	×	✓	×	×	Not mentioned in the paper
[13]	2021	✓	×	✓	✓	×	✓	×	×	Not mentioned in the paper
[14]	2020	×	✓	×	✓	×	✓	×	×	Not mentioned in the paper

Table 0-2: Comparison of The Proposed Trust Models based on Trust Dimensions

Reference #	Trust Composition							Trust Propagation		Trust Update		Trust Formation		Trust Aggregation					Types of Weights	
	Quality of service (QoS)			Social Trust																
	Response Time	Packet Delivery Ratio	Capability of Server	Friendship	Honesty	Ownership	Location of Server	Distributed	Centralized	Event Driven	Time Driven	Single Trust	Multi Trust	Bayesian Inference	Similarity measures	Subjective Logic	Logistic Regression	Weighted sum	Static	Dynamic/Adaptive
[5]	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✓	✗	✓
[6]	✗	✗	✗	✗	✗	✗	✗	✓	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
[10]	✓	✗	✗	✗	✗	✗	✗	✓	✗	✓	✗	✓	✗	✓	✓	✗	✗	✗	✗	✓
[11]	✓	✓	✗	✓	✓	✓	✗	✓	✗	✓	✗	✗	✓	✗	✗	✓	✗	✗	✗	✓
[3]	QoS (parameters not specified)			Social Trust (parameters not specified)				✓	✓	✓	✓	✓	✓	✗	✗	✓	✓	✗	✗	✗
[12]	✗	✗	✓	✗	✗	✗	✓	✓	✗	✗	✓	✗	✓	✓	✗	✗	✗	✗	✓	✗
[13]	✓	✗	✗	✗	✗	✗	✗	✓	✗	✓	✗	✓	✗	✓	✓	✗	✗	✗	✗	✓
[14]	✗	✗	✗	Social Trust(parameters not specified)				✓	✗	✗	✓	✗	✓	✗	✗	✗	✗	✓	✗	✓

3.2 Trust Related Attacks

Trust-based security solutions play a crucial role in evaluating a user's trustworthiness score and identifying potentially malicious entities within IoT networks. In these networks, connected objects assess the reliability of Fog nodes and make decisions about future interactions based on both their historical interactions with the Fog node (known as direct trust) and recommendations from other objects (referred to as indirect or recommended trust).

Indirect or Recommended Trust involves objects sharing their assessments of trust in neighboring objects as recommendations. However, it's important to note that false recommendations can be intentionally provided by malicious nodes. Such deceptive actions

can result in legitimate nodes receiving unfairly low trust values, which is akin to a bad-mouthing attack. In a bad-mouthing attack, malicious entities attempt to tarnish the reputation of others by spreading negative opinions or false information about them.

Additionally, a collusion attack is a more organized form of manipulation where hostile nodes collaborate to transmit detrimental recommendations targeting a specific node. In such attacks, the collaborative malicious entities work together to undermine trust in the targeted node, potentially leading to reputational damage or adverse consequences for the victimized entity.

Three other frequently mentioned trust-related attacks in the literature are Self-Promotion attack, On-Off attack, and Opportunistic Service attack. In the Self-promotion attack, any node can send positive reputation reports for itself (self-promotion) so to get selected as a service provider, and once it is chosen it goes on to give poor services only. In the On-Off attack, a malicious node periodically switches between good (ON) and bad (OFF) behaviour. The bad behaviour (OFF state) of the malicious node appears to be a temporary error and thus stays undetected even as the malicious node remains active in the network over a prolonged period. When an authenticated but compromised node behaves badly, it earns a low reputation; therefore, it switches its behavior to good after a while which results in a trust build up. The high trust score is then utilized to attack the network. The malicious node ensures that reputation loss due to bad behaviour is kept to a minimum and that its overall reputation is always positive, ensuring that it remains undetected. Opportunistic service attacks tarnish IoT nodes' reputation as recommenders. Different nodes or groups of nodes may receive discriminatory services from a service provider. A malicious service provider will modify its service based on who is requesting the service. For example, one node/group may receive a high-quality service while another receives a low-quality service. Particularly, non-friend nodes are discriminated against. Due to their differences in experience, the two nodes/groups will provide different recommendations for the same service. Some will deem the service as benevolent while others will label the same service as malevolent. This will result in a lowered trust value for them, as a recommender.

Trust-based security solutions are essential in fog-based IoT networks to evaluate trustworthiness, but they must be cautious about the potential trust-related attacks. The research papers we reviewed proposed different trust management systems with varying degrees of resilience against trust-related related attacks as shown in Table 3.3 below:

Table 0-3: Resilience Of The Proposed Trust Against Trust-related Attack

Reference #	Self Promotion Attack	On-Off Attack	Opportunistic Service Attack	Ballot Stuffing Attack	Badmouthing Attack	Comments
[12]	-	-	-	-	Recommendation credibility score as a weight for total indirect trust calculation, ensures that malicious nodes have a lower contribution in indirect trust computation.	Resilience towards the badmouthing attack is shown via simulation
[13]	No reference is made to trust-related related attacks in the paper					
[17]	Does not allow self recommendations	The behaviour of a node is monitored over a period and if there are continuous fluctuations then the node is discarded from the network.		Similarity-based filtering ensures that recommendations from only close social contacts are considered. Also, recommendations are taken from multiple nodes so it will require collusion of many malicious nodes to malign the recommended trust score of a node.		Resilience towards trust-related attacks is only justified theoretically
	Adaptability provided by dynamic values of weights helps prevent trust-related attacks.					
[18]	Does not allow self recommendations	Behaviour of a node is periodically monitored and if it is fluctuating, the node is removed from the network		Recommendations are weighted based on the trust level of recommenders thus a malicious nodes' contribution to the overall indirect trust will be small. Exaggerated recommendations are also ignored from the recommendations list.		Resilience towards trust-related attacks is only justified theoretically
[19]	Does not allow self recommendations	-	Both long and short-term activity of nodes is recorded. Misbehaviour in the past is marked.	(Multiple) Recommendations are rated according to the trust level associated with the recommender. If an object produces a wrong recommendation, it will get a lower recommender trust.		Resilience towards trust-related attacks is only justified theoretically
[20]	Trust-related attacks not catered in the paper					
[21]	-	-	-	Similarity-based filtering ensures that recommendations from only close social contacts are considered. Context-dependent weight also reduces the impact of these attacks		Resilience towards attacks is shown via simulation
[22]	Trust-related attacks not catered in the paper					

CONTEXT-BASEDTRUST MANAGEMENT SYSTEM

In this chapter, we introduce our proposed trust management system, outlining its key components including the trust model itself, the sequence of events, the mathematical model, and the evaluation method. Within a Fog-based IoT network, three key players exist: Cloud server(s), Fog nodes and IoT nodes/devices. Our model specifically focuses on evaluating trust levels between IoT devices and fog nodes engaged in communication. It should be noted that our model operates in a distributed fashion, where each node assesses the trustworthiness of the counterpart it is communicating with, considering the contextual factors influencing the trust calculation.

4.1 System Model

In the proposed solution there are three key players:

- SR = Service requesting node or client node requesting a particular service. The storage and processing capacity of these nodes is typically limited.
- SP = Service providing node responsible for delivering the service requested.
- R = Recommenders are nodes having experience with the service provider for the service that is being requested.

The primary focus of this model is to establish trust connections between fog nodes, enabling the seamless sharing and delegation of computing resources. To streamline the discussion, we're concentrating on a single-tier fog ecosystem, as opposed to a multi-tiered setup, while still retaining its essential capabilities.

Within this framework, fog nodes can engage in direct communication with neighboring fog clients or servers within a one-hop distance. An essential component of this trust framework is the trust threshold, which has been defined at 0.5. Nodes that possess trust values surpassing this threshold are considered reliable and trustworthy.

In our proposed model, the process unfolds as follows:

i. A Service Requester (SR) initiates a request for a specific service from a Service Provider (SP).

ii. Before establishing a connection, both SR and SP undergo mutual validation based on their past interactions. This validation step serves as a safeguard against rogue or malicious nodes.

iii. SR consults its local trust table to determine if it has any prior experience with the SP. If no previous interaction is found, a neutral 0.5 is assigned as a default.

iv. The service requesting node reaches out to its neighboring nodes for recommendations, which then contribute to the calculation of indirect trust.

v. Subsequently, the score for total trust is computed by taking a sum of the direct and indirect trust scores.

vi. In case of the service providing node's trust score falling below the established threshold, the request for connection is immediately declined, prompting the service requestor to seek an alternative connection within the network.

vii. Conversely, if the service providing node's trust score surpasses the threshold, the service requesting node proceeds forwards with the request for connection to the SP.

viii. The service providing node reciprocates by verifying the service requesting node's validity and follows the same trust evaluation steps.

ix. Calculated trust value exceeding the defined threshold results in the establishment of connection between the SP and SR.

x. The requested service (S) is provided by the SP to the SR.

xi. Following service delivery, SR records its experience with the SP, which contributes to the total trust score for future interactions.

The diagram in Figure 4.1 provides an illustrative overview of our trust management model, which forms the foundation for establishing trust within a Fog-based IoT network. This model leverages both direct interactions and indirect recommendations to foster trust relationships among the entities within the network.

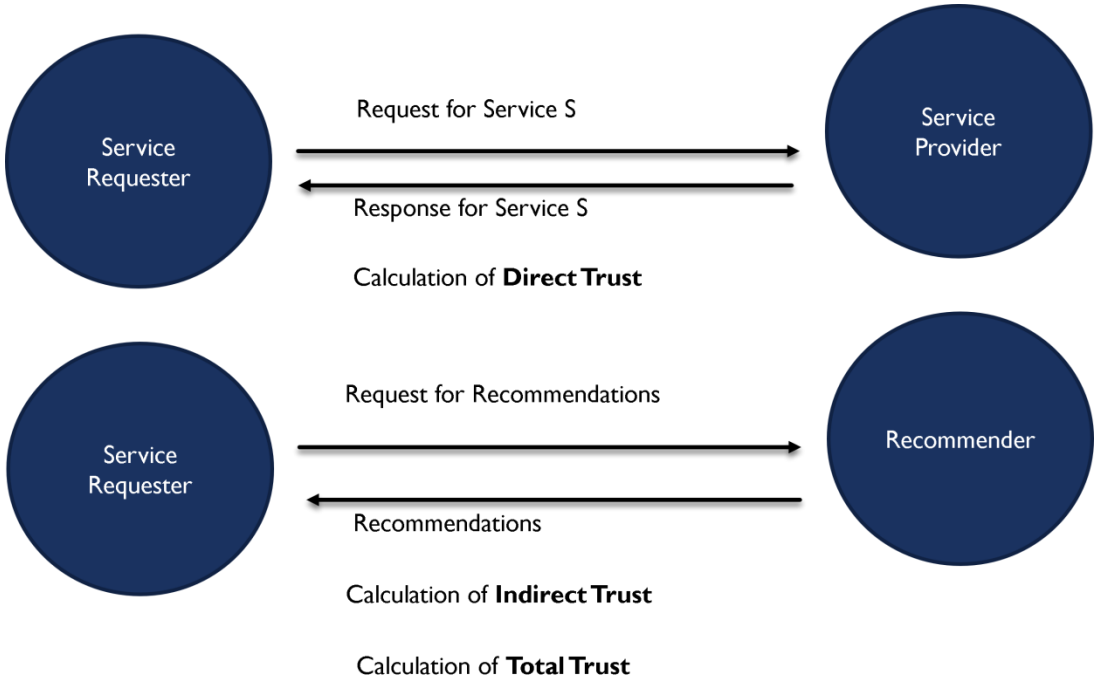

Figure 4.1: Flow Diagram of the proposed Trust Management System

This model thus fosters trust-based interactions while safeguarding against potential security threats within the Fog-based IoT network.

Context is incorporated at two points in the trust calculation process. Firstly, before any service is exchanged a service requesting node (SR) broadcasts a service request. This service request contains the information of the service required i.e., the SR specifies which service among S1 (Data Processing), S2 (Control System) or S3 (Realtime Monitoring) etc. it requires. The service providing node (SP) responds to this request for service with certain parameters. Based on SP's response and the past-experience(s) SR had with the SP, direct trust (TD) is calculated. Response time and packet delivery ratio are the QoS trust metrics used for SP's direct trust score calculation. The method utilized for this calculation is Bayesian inference and the threshold for an acceptable trust score is 0.5. By adding service information in the service request, we have enabled the TMS to measure the capability of SP in the context of the exact service being requested.

Secondly, the SR requests its neighboring nodes for recommendations regarding the SP. Context is incorporated in indirect trust (TR) as well. Similarity scores based on Jaccard similarity are used to filter recommendations based on the service, server, and location similarity. Recommendations are made context aware by only considering recommendations

from those recommenders (R) that have utilized similar services from similar servers located at similar locations as the SR. The threshold for an acceptable trust score is again 0.5.

Total trust (TT) score is the aggregate of the direct and indirect trust scores. If the resulting total trust is greater than the acceptable trust threshold value only then service is taken from SP. Also, once the service exchange has been completed the SR updates its past-experience database according to the quality of service provided by SP.

4.2 TrustCalculation Methodology

In this section, we delve into the trust calculation methodology, which encompasses the computation of direct trust through Bayesian Inference, the determination of indirect trust via recommendations received from neighboring nodes, and the assessment of their similarity scores with the service requesting node using Jaccard Similarity. Ultimately, this process culminates in the computation of the overall trust level.

Parameters used in our trust model are given in Table 4.1.

Table 0-1: TMS Parameters

Trust Parameter	Description
S	Type of Service
$I_{a,b}$	Current user interaction
Δt	Time difference between two consecutive service transactions
$e^{-d\Delta t}$	Exponential decay in trust
$T^{d}_{a,b}$	Direct trust
$T^{r}_{a,b}$	Indirect trust
$T^{t}_{a,b}$	Total trust
$sim^{S}_{x,y}$	Services based similarity score
$sim^{C}_{x,y}$	Social contacts-based similarity score
$sim^{SP}_{x,y}$	Service providers-based similarity score

Each node uses and shares certain information in the form of lists for the calculation of trust scores.

i. L_S: List of services used by the node. $L_S = \{S_1, S_2, S_3, \ldots, S_n\}$.

ii. L_{SP}: List of service providers the node took services from. $L_{SP} = \{SP_1, SP_2, SP_3, \ldots, SP_n\}$.

iii. L_C: List of social contacts of the node. L_C= {C_1, C_2, C_3, …, C_n}.

iv. L_T: List of past experiences (trust scores) of a node with different nodes. L_T={T_{SP1}, T_{SP2}, T_{SP3}, …, T_{SPn}}.

DirectTrust

Bayesian inference is used to determine the direct trust score of the SP based on the SR's current interaction $I_{x,y}$ and past-experience with the SP. Bayesian Inference is chosen due to its track record of effectiveness in trust modeling. Various parameters are leveraged to compute user interaction, encompassing factors such as latency, response time, and packet delivery ratio, among others. The quality of service (QoS) based trust metrics utilized in theproposed system are Response Time and Packet Delivery Ratio. Response time refers to the duration it takes for a system, typically a fog node or a service provider, to acknowledge and respond to a request or command from an IoT device. Packet delivery ratio (PDR) is a metric that measures the percentage of data packets successfully transmitted from a source (e.g., an IoT device) to a destination (e.g., a fog node or the cloud) without being lost or dropped during transit. It's worth emphasizing that the trust management solution we've introduced also considers the type of the requested service when performing trust calculations. This approach ensures that the assessment of the service-providing node's capabilities is contextually aligned with the specific service being requested.

Node A'a directtrusttowardsBcanbe represented by:

$$T_{a,b}^{d} = \frac{\alpha_{a,b}}{\alpha_{a,b}+\beta_{a,b}} \tag{4.1}$$

Within equation (4.1), α and β represent the parameters of the Beta distribution. Their values are determined based on the time gap between two service transactions and the subsequent reduction in trust. Trust decay, in this context, models how prior trust levels gradually lose their impact on the present trust assessment, factoring in the time that has passed between these interactions. This phenomenon mirrors real-world situations where trust naturally diminishes over time when there's a significant gap in interaction between two entities. It's worth emphasizing that trust decay primarily pertains to direct trust, as it relies on the trustor's assessment of the trustee's trustworthiness and doesn't influence the overall trust score.

The equations used for the calculation of α and β are as follows:

$$\alpha_{a,b} = e^{-d\Delta t} \times \alpha_{a,b}{}' + I_{a,b} \quad (4.2)$$

$$\beta_{a,b} = e^{-d\Delta t} \times \beta_{a,b}{}' + (1 - I_{a,b}) \quad (4.3)$$

In this context, we use the variable $I_{a,b}$,to represent the user's satisfaction experience from node A to node B. It's a binary indicator, where 1 denotes a satisfactory experience and 0 signifies dissatisfaction. In the equations presented above,$I_{a,b}$, contributes to positive observations, while $(1-I_{a,b})$ contributes to negative observations. Additionally, $\alpha_{a,b}{}'$ and $\beta_{a,b}{}'$represent previous scores, which are computed based on Node B's (SP's) historical total trust score as stored by Node A (SR), whereas $\alpha_{a,b}$and $\beta_{a,b}$denote new values.The term $e^{-d\Delta t}$signifies exponential decay, where d is the decay factor occurring over a periodΔt.

Indirect or Recommended Trust

Indirect or recommended trust in this context is derived from the neighboring recommending nodes that have prior experiences with the same server under similar circumstances. Node x seeks trust recommendations from its neighboring nodes regarding node y. These neighbors share their overall trust scores with the Service Requester (SR). However, along with the recommendations, the SR and R also share with each other their list of services obtained (L_S), servers (L_{SP}), and social contacts (L_C). SR employs a similarity measure to gauge its contextual similarity with the recommender nodes.

Several similarity methods are available for this purpose, such as Cosine, Jaccard, Euclidean distance, and Pearson Correlation, among others. After evaluating the performance of these four similarity measures, we concluded that Jaccard similarity is the most suitable choice. This decision stems from its simplicity, computational efficiency, and favorable outcomes for IoT systems characterized by resource constraints and time-sensitive operations.

Jaccard similarity is a measure that quantifies the similarity between sample sets by comparing the size of their intersection to the size of their union. It provides a similarity score ranging from 0 to 1, with values between 0 and 0.499 indicating dissimilarity and values between 0.5 and 1 indicating similarity. In the context of this trust assessment model, we place significance on the similarity between the server, service, and social contacts as pivotal factors in determining trustworthiness.

$$\text{sim}_{x,y} = \frac{L_x \cap L_y}{L_x \cup L_y} \quad (4.4)$$

Using equation 4.4, similarity scores between the SR and every recommender is calculated based on common servers, social contacts, and services respectively. The final similarity score is used as a weight for the respective recommendations. Discounting and consensus of the filtered recommendations results in the total indirect trust score as shown below.

$$T_{a,b}^{r} = \frac{sim_{R_1} \times T_{R_1} + sim_{R_2} \times T_{R_2} + \ldots + sim_{R_n} \times T_{R_n}}{\sum_{i=1}^{n} sim_{R_i}} \quad (4.5)$$

Total Trust

Finally, total trust score is computed as a weighted sum of the direct and recommended trust as shown in equation 4.6. μ is an adaptive weight calculated using information gain.

$$T^{t}_{a,b} = \mu^{d} \cdot T^{d}_{a,b} + \mu^{r} \cdot T^{r}_{a,b} \quad (4.6)$$

Adaptive Control of Weight Parameter

The weight parameter μ, which falls within the range of 0 to 1, undergoes dynamic adjustments to reduce the potential bias in trust estimation. The specific value of μ plays a pivotal role in determining whether direct trust will have greater impact on the overall trust score or indirect trust. This dynamic allocation of the weight parameter assists in determining which value, either recommended or direct trust, should be given more credence.

Within the realm of existing research, scholars have explored various methods for determining these weights, encompassing both static and dynamic weighting techniques. In our specific scenario, we have chosen to employ an information gain-driven weighting methodology to dynamically ascertain the weights associated with direct and indirect trust. This selection amplifies the versatility of our trust model, guaranteeing that the process of estimating trust remains flexible and contextually sensitive.

Information Gain is a concept from information theory and statistics that is used to measure the reduction in uncertainty (or entropy) achieved by incorporating new information. In our proposed trust management system, Information Gain is used to calculate adaptive weights for direct and indirect trust based on how much new information or uncertainty reduction each source of trust (direct and indirect) provides.

Entropy Calculation:

The trust management system calculates the entropy before and after incorporating trust information. Entropy is a measure of uncertainty or randomness in a dataset. In this case, it measures the uncertainty in trust scores before and after considering recommendations.

$$\text{Entropy} = -\sum p_i \log(p_i) \quad (4.7)$$

Where p_i represents the probability of each trust score value in the set

Information Gain Calculation:

Information Gain is calculated as the difference between the entropy before and after incorporating recommendations. High Information Gain means that the recommendations significantly reduce uncertainty, while low Information Gain implies that the recommendations don't provide much new information.

Weight Calculation:

The total weight is computed as 1 plus the Information Gain. This is because Information Gain is typically a positive value and adding 1 ensures that the weights always sum to 1.

Weight for Direct Trust: This is calculated as the reciprocal of the total weight. It represents the relative importance of direct trust concerning reducing uncertainty.

Weight for Indirect Trust: This is calculated as the Information Gain divided by the total weight. It represents the relative importance of indirect trust concerning reducing uncertainty.

Weighted Trust Calculation:

Weighted Direct Trust: This is the product of the direct trust score and the weight for direct trust.

Weighted Indirect Trust: This is the product of the indirect trust score (average of weighted recommendations) and the weight for indirect trust.

Total Trust Score Calculation:

The total trust score is calculated as the sum of the weighted direct and indirect trust. This represents the overall trustworthiness of a service provider, considering both direct and indirect trust.

The flexibility of these weight adjustments plays a pivotal role in safeguarding against various trust-related threats, such as:

- bad-mouthing attack
- ballot stuffing attack
- self-promotion attack
- opportunistic service attack

In our approach, each node independently calculates and continually updates its weight value in every trust update cycle.

In contrast, the static weighting method fixes weight values before the final trust calculation. In this approach, a higher weight value emphasizes the significance of direct trust in determining the final trust score, while a lower value implies a greater emphasis on indirect trust. However, static weights create a vulnerability by enabling malicious nodes to manipulate trust scores easily, potentially leading to erroneous judgments of trustworthiness.

To mitigate this issue, our research employs a dynamic weighting method. This dynamic approach adaptively calculates weight values. This ensures that a single unfavorable recommendation or direct experience cannot disproportionately influence a node's overall trust assessment. Dynamic weighting is characterized by its impartiality and reliability, making it a more robust choice for trust management.

EVALUATION OF PROPOSED MODEL

We initially evaluated the proposed solution mathematically, using numerous test scenarios. An important aspect of any trust management system is its ability to incorporate successfully in its trust calculations the impact of both the past-experience(s) and current interaction with a service provider. As time between two service transactions varies the impact of past-experience and current interaction on total trust score varies. As a service requester frequently engages with a service provider, the impact of former's past experience with the later carries a greater impact on the trust score. However, as time between consecutive service transactions increases the impact of past experience on trust score starts to decrease. After a delay of 30 minutes current user interaction greatly determines the trust score. These findings are depicted in the three graphs below. We created three different scenarios.

5.1 A High Performing Honest Service Provider

A high performing honest node will characteristically have high quality service response and delivery and high trust scores resultantly. However, it might give a poor service response due to any reason such as network disruption, temporary error, or being under a badmouthing attack by a malicious entity. This scenario is depicted in figure 5.1 where the honest node's service response experiences a degradation, at trust update cycle 4 and 8.

5.2 A Low Performing Service Provider

A low performing service provider will characteristically have low trust scores due to its low-quality service response and delivery. However, such a node might give a suspiciously high service response if it is carrying out a self-promotion attack, or if it has formed a collusion with malicious recommending nodes which are carrying out a ballot stuffing attack in its favour. This scenario is depicted in figure 5.2 where the low performing node's service response experiences an uncharacteristic improvement at trust update cycle 4.

5.3 A Service Provider with Fluctuating (On-Off) Behaviour

One of a common trust related attack carried out by malicious nodes is an on-off attack. Such a malicious service provider will behave in a fluctuating (on-off) manner. We

created this scenario in figure 4. At every update cycle the nodes behaviour switches from good to bad, from 0 to 1 or vice versa.

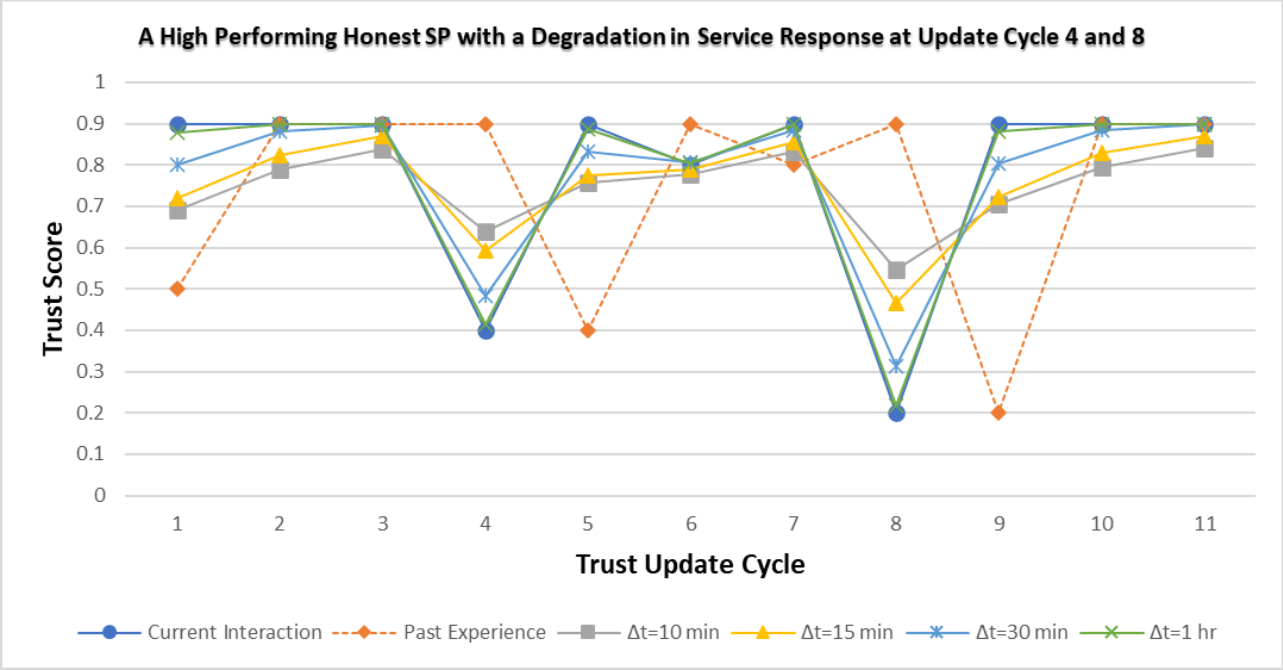

Figure 5.1: Trust score variation of a high performing service provider

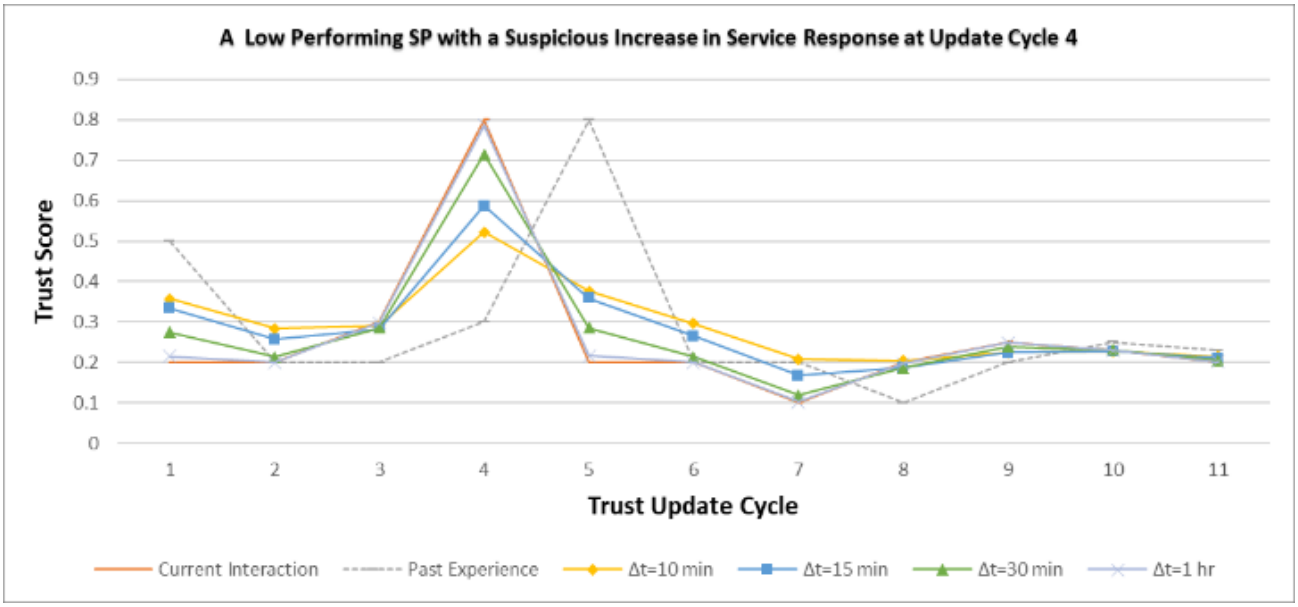

Figure 5.2: Trust score variation of a low performing service provider

All the graphs show how the trust score of service provider varies with different Δt (time difference between two consecutive service transactions) values. At lower Δt values such as 10 and 15 minutes, for every update cycle the proposed model successfully calculates the impact of both the past experience and current user interaction on trust score. However, as Δt increases from 30 minutes to 1 hour, the

impact of past experience on trust score drastically reduces. This is because trust is not a static value and with significant delays in the service transactions, past trust scores values start to decay and their impact on current trust score diminishes.

This is evident at update cycle 4 and 8 in figure 2. Not considering the high performing past behaviour of the honest node, with Δt = 1 hour, trust score is only representative of the current poor service response. Similarly in figure 3, at update cycle 4, with Δt = 1 hour the past poor performance of service provider does not contribute at all to the trust value and the S_P is granted a high trust value solely on the basis of current interaction. Similar results were obtained for a service provider with a fluctuating on-off behaviour as shown in figure 5.3.

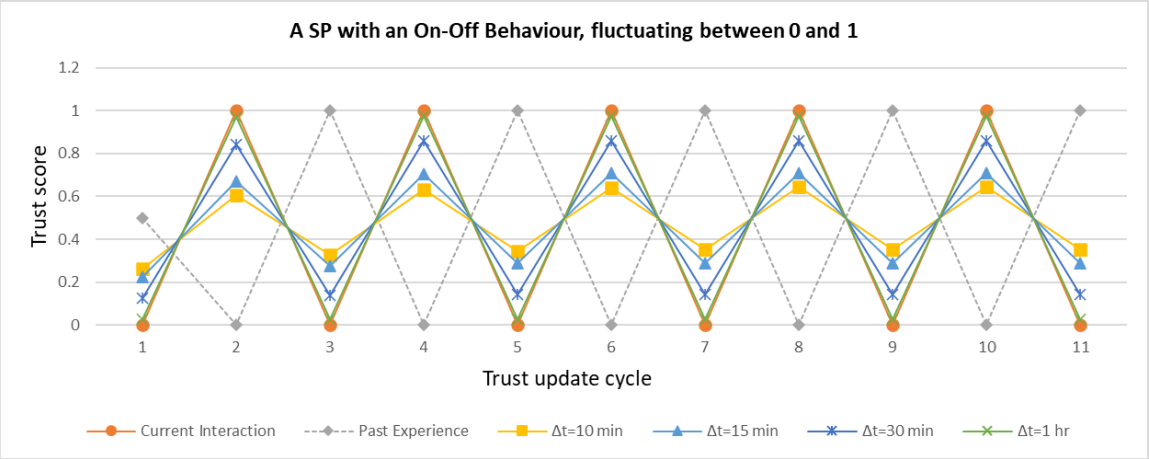

Figure 5.3: Trust score variation of a service provider with a fluctuating (on-off) behaviour.

From the graph above we can observe that the most optimum trust scores are obtained at Δt = 15 minutes, where the current user interaction as well as past behaviour of the node equally and effectively determine its trust score.

With regards to trust update dimension there are commonly two types of trust management systems:

- Event-driven TMS
- Time-driven TMS

An event-driven TMS calculates and updates trust scores based on the occurrence of an event. For example, when a service is being requested or after a service transaction has occurred. On the other hand, a time driven trust management system without waiting for an event, calculates and updates trust scores after a specific period of time. From our findings above we propose that 15 minutes is an optimum time for trust update for a

time driven TMS. Anything below will lead to excessive computation and storage overhead and anything above will result in less accurate trust scores.

In the second phase we conducted a comprehensive evaluation of our proposed solution within the context of fog-based IoT systems, employing mathematical analysis and synthetic datasets generated through a Python simulation. This synthetic dataset closely emulated real-world scenarios encountered in fog-based IoT environments. Specifically, we established a network configuration involving 20 service providers and calculated trust scores for 10 recommenders over 100 iterations. These trust scores were determined based on synthetically generated metrics, including response time, packet delivery ratio, and the type of service being sought. To ensure the fidelity of our evaluation, we defined appropriate ranges for each trust metric, taking into account the characteristics unique to fog-based IoT systems:

- Response Time: Set within the range of 10ms to 500ms.
- Packet Delivery Ratio: Varied between 0.8 to 1.0, representing percentages from 80% to 100%.
- Type of Service: We categorized services into distinct types, such as "Real-time Monitoring," "Data Processing," and "Control Systems."

Our simulation code then generated synthetic data for each service provider in every iteration, encompassing response time, packet delivery ratio, service type, time intervals between consecutive services, and timestamps indicating when each service was provided. The initial time difference between consecutive services was randomized between 1 to 10 seconds, and this pattern was maintained for subsequent services.

An essential feature of our proposed trust management system is its contextual awareness. In a fog-based IoT environment, service providers may exhibit varying capabilities, excelling in certain services while performing less effectively in others. Our system adeptly captures these disparities, resulting in dynamically adjusted trust scores based on the type of service offered by the same service provider. This dynamic adaptation is visually demonstrated in the graph below (figure 5.4), reflecting the differentiated trust scores corresponding to the various service types provided by a given service provider.

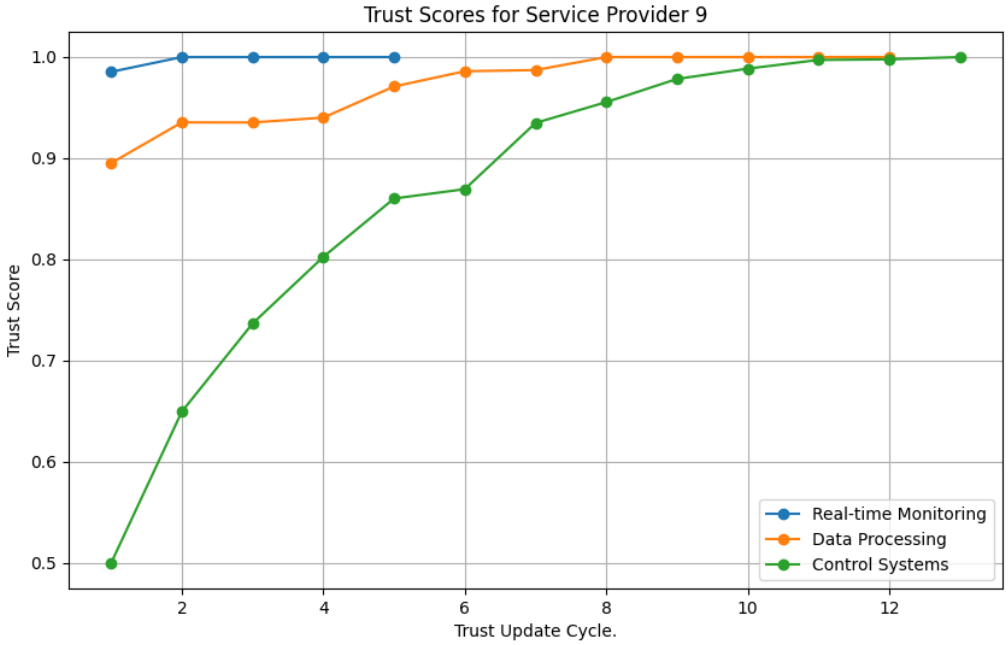

Figure 0.4:Direct Trust score variation for different type of services

Our proposed solution also utilizes similarity scores to find contextually similar recommenders from the shared server, services, and location lists respectively. To determine the most suitable similarity method, we conducted a comparison among four options: Jaccard, Cosine, Pearson Correlation, and Euclidean distance.

Jaccard Similarity

Definition: Jaccard similarity measures the similarity between two sets by comparing their intersection to their union.

Use Case: It is commonly used for text analysis, document similarity, and recommendation systems, where you want to measure the similarity between two sets of items.

Cosine Similarity

Definition: Cosine similarity measures the cosine of the angle between two non-zero vectors in a multi-dimensional space.

Use Case: It is widely used in information retrieval, text analysis, and recommendation systems. It's particularly useful when you want to find the similarity between documents or texts represented as vectors.

Euclidean Distance

Definition: Euclidean distance is the straight-line distance between two points in a multi-dimensional space. In the context of similarity, it's often used as a dissimilarity metric, so smaller values indicate greater similarity.

Use Case: Euclidean distance is used in clustering and classification tasks, as well as in anomaly detection, where you want to measure the distance or dissimilarity between data points.

Pearson Correlation Similarity

Definition: Pearson correlation measures the linear correlation between two variables. In the context of similarity, it measures how well the relationship between two variables can be represented by a straight line.

Use Case: It is commonly used in statistics and data analysis to measure the similarity or correlation between numerical data. In collaborative filtering for recommendation systems, it can be used to find similarities between users or items based on their ratings or preferences.

These similarity metrics are essential in various fields, including data mining, machine learning, natural language processing, and recommendation systems, to quantify the similarity or dissimilarity between data points or sets. The choice of similarity measure depends on the specific problem and the type of data being analyzed. Before choosing any one similarity measure, we tested the four methods mentioned above on synthetic data to rule out the best choice for our use case.

In our evaluation, involving 20 service providers and 10 recommenders, we assumed that the service-requesting node interacted with all 20 service providers. Specifically, Recommender 1 utilized services from all available service providers, while the remaining recommenders had progressively reduced service provider interactions. The graph below (figure 5.5) illustrates the similarity scores calculated for this particular scenario:

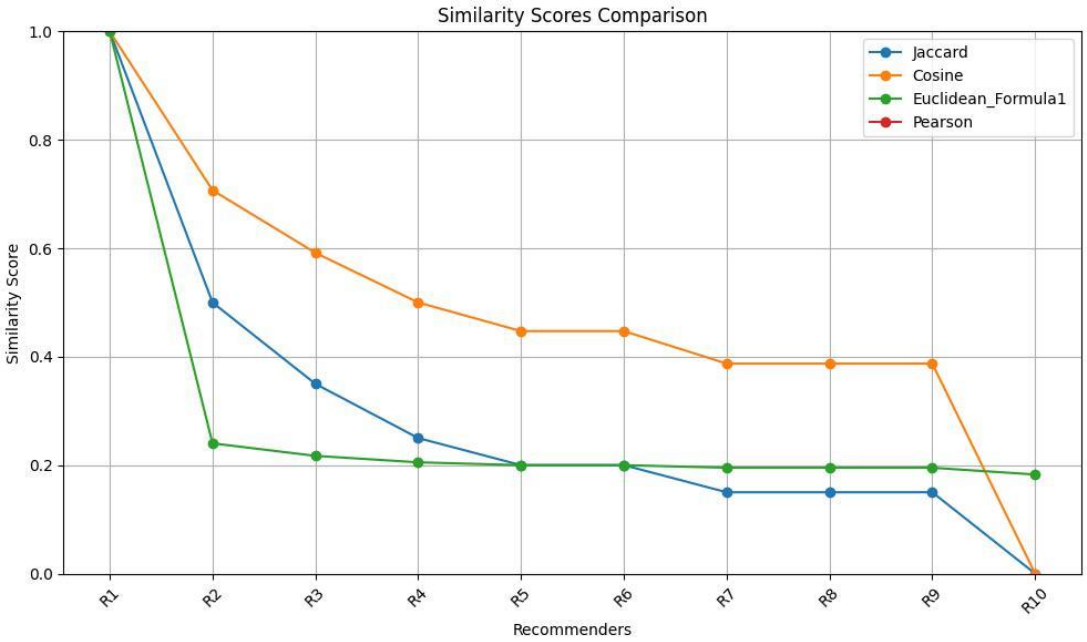

Figure 0.5: Comparison of similarity scores.

No results were obtained for Pearson Correlation initially because the data on which similarity scores were measured was in the form of binary vectors. Pearson correlation coefficient is not well-suited for binary data because it is designed to measure the linear relationship between two continuous variables. Binary data, on the other hand, is categorical and can only take on two values (usually 0 and 1), representing the absence or presence of a particular attribute or event. The Pearson correlation coefficient relies on the assumption that the variables being compared are continuous and follow a normal distribution. It calculates the degree to which two variables move together in a linear fashion. In other words, it assesses how well the data points fit a straight line. Binary data, being categorical and having only two possible values, does not exhibit the continuous and normally distributed characteristics that Pearson's correlation assumes. When applied to binary data, Pearson's correlation provided no meaningful results because it doesn't account for the binary nature of the data points.

The standard Pearson correlation coefficient is measured using the formula as follows:

$$r = \frac{n(\sum xy) - (\sum x)(\sum y)}{\sqrt{[n\sum x^2 - (\sum x)^2][n\sum y^2 - (\sum y)^2]}} \quad (5.1)$$

A modified Pearson Formula for binary data was used as follows:

$$sim_{x,y} = 1.0 - abs(mean(x) - mean(y)) \quad (5.2)$$

here x and y are binary data lists (0s and 1s) and the modified formula quantifies how similar or dissimilar x and y are in terms of the proportion of 1s. It provides a value between 0 and 1, where 0 indicates no similarity (completely different binary lists), and 1 indicates perfect similarity (identical binary lists in terms of the proportion of 1s). Output ranges from 0 to 1, indicating the degree of similarity between binary lists. This modified formula gave results as depicted in figure 5.6 below:

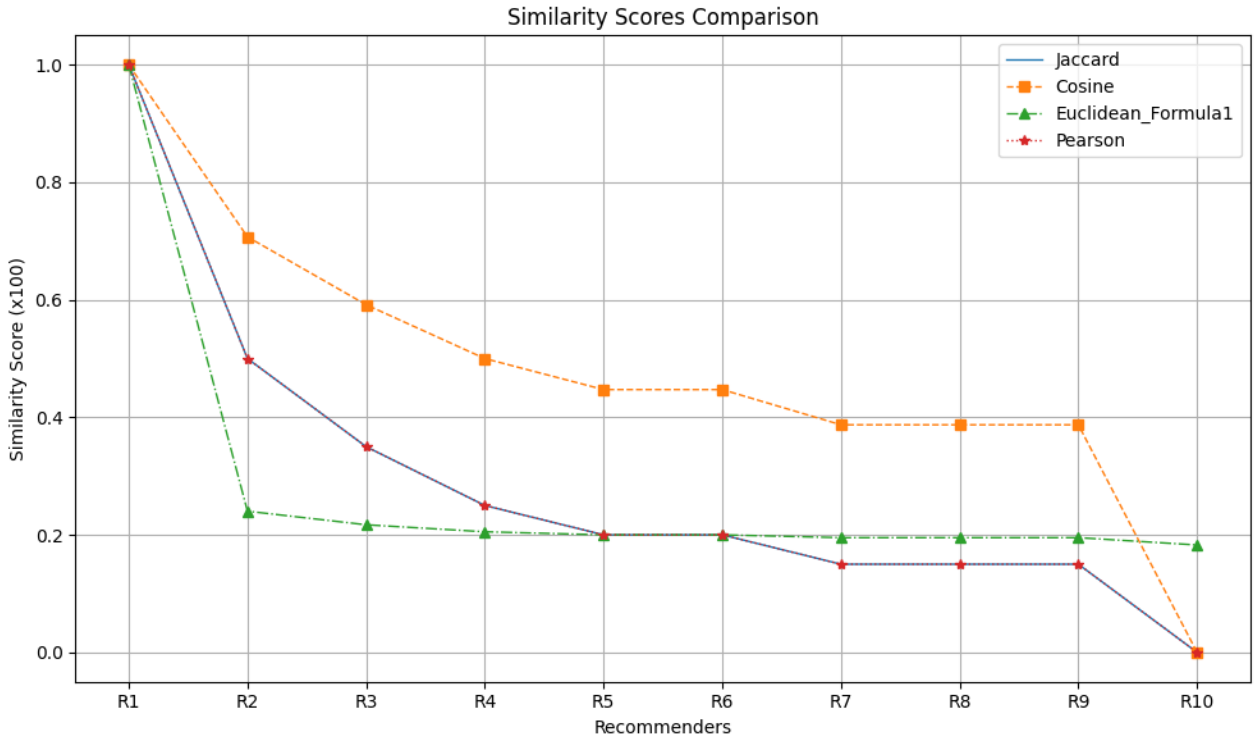

Figure 0.6: Similarity scores obtained with modified Pearson Formula

To evaluate the modified formula further, we considered a scenario comprising 100 service providers and 50 recommenders. In the first scenario, we assumed that the service-requesting node had engaged with all 100 service providers. To be specific, Recommender 1 utilized services from all the available service providers, whereas eachremaining recommender had a decreasein their interactions with service providers byhalf. The graph (figure 5.7) below depicts the similarity scores computed for this specific scenario:

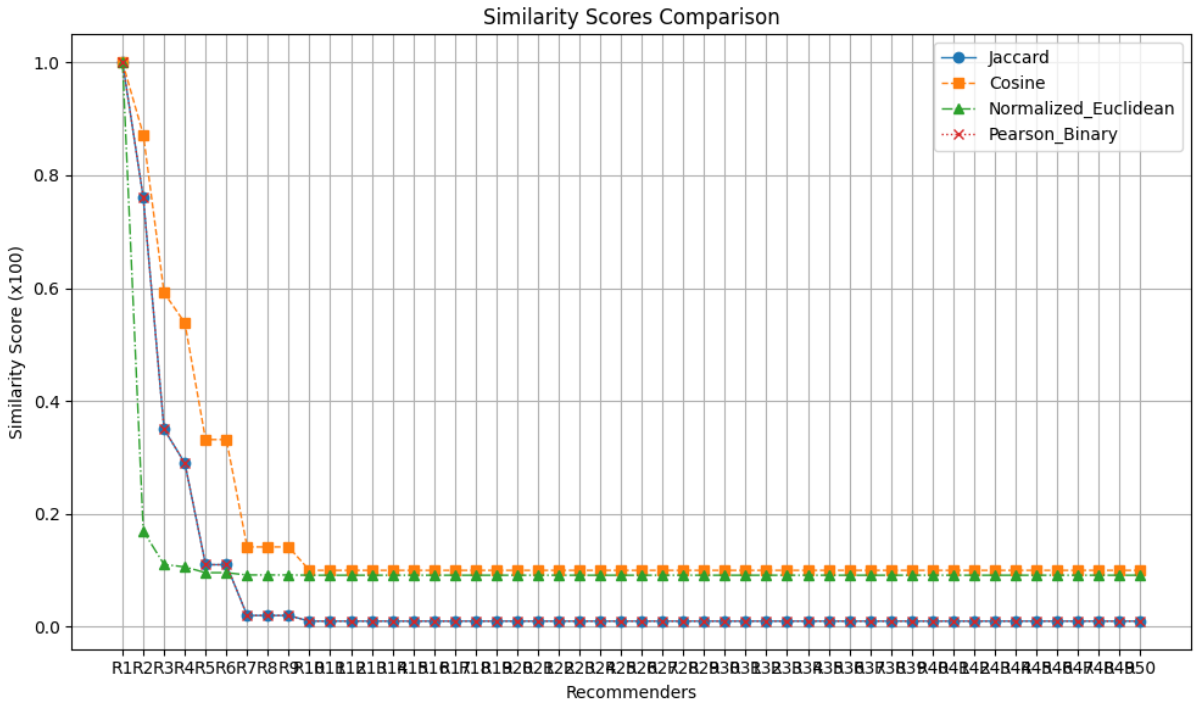

Figure 0.7: Similarity scores of recommenders with a decreasing number of common interactions by half.

In the second scenario, we again assumed that the service-requesting node had engaged with all 100 service providers. Recommender 1 utilized services from all the available service providers, whereas the remaining recommenders gradually decreased their interactions with service providers, by reducing them by a factor of 2 each time. The graph (figure 5.8) below depicts the similarity scores computed for this specific scenario:

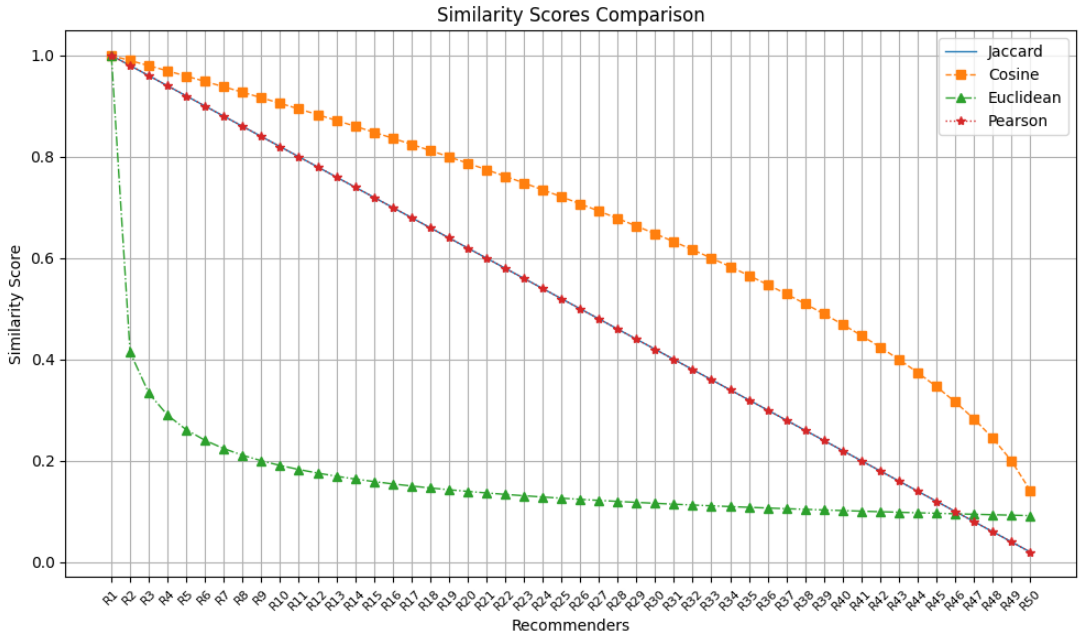

Figure 0.8: Similarity scores of recommenders with a decreasing number of common interactions by a factor of 2

From the graph above (figure 5.8) we can see that the four similarity methods (Jaccard, Cosine, Euclidean, and Pearson Binary) produce varying similarity scores for the service requesting node and the 50 recommenders. These scores can be analyzed considering both high and low similarity scenarios:

High Similarity

In the case of high similarity (e.g., between Recommender 1 and the service requesting node), all four similarity methods produce relatively high scores, close to 1.0. This indicates that these methods effectively capture the similarity when lists are identical.

Low Similarity

As the interactions between the service requesting node and the recommenders become less similar (e.g., as recommenders engage with fewer service providers), the four similarity methods show differences in their scores:

Jaccard: Jaccard similarity tends to be relatively high when there is a non-zero intersection between the sets of lists. However, it decreases as the intersection becomes smaller.

Cosine: Cosine similarity measures the cosine of the angle between two vectors. It is high even when there is some degree of commonality but does penalize the magnitude of interactions. So, it still yields relatively high scores in cases of low commonality.

Euclidean: Euclidean distance calculates the distance between points, with smaller distances indicating higher similarity. It yields lower scores when commonality decreases.

Pearson Binary: This modified Pearson correlation for binary data measures the linear relationship between binary vectors. It yields scores close to 1.0 when vectors are nearly identical but shows a decrease as interactions diverge.

The choice of similarity method should align with the specific characteristics of the data and the aspects of similarity that are most relevant to the analysis. If the priority is to capture common interactions, Jaccard and Cosine similarity methods seem more suitable. However, if the emphasis is the magnitude of interactions along with commonality, Euclidean distance could be considered. Pearson Binary may be appropriate if one is interested in assessing linear relationships in binary interactions.

Based on this analysis, we have chosen the Jaccard similarity method for our proposed solution. This decision aligns with the model's requirement to measure the similarity between

the service requesting node and the recommending node by considering their shared experiences, specifically in terms of common services, service providers, and locations. Among all 50 recommenders, the Jaccard similarity method exhibited strong performance by providing linear, and more finely detailed similarity scores.

In the proposed trust management system, total trust is computed as the combination of weighted direct and indirect trust. These adaptive weights are determined through Information Gain, a concept from information theory. Information Gain quantifies the reduction in uncertainty or entropy achieved by incorporating new data. Entropy, a measure of uncertainty, is calculated both before and after considering trust information. It reflects the degree of uncertainty in trust scores. Information Gain is then computed as the disparity between these two entropy values. Higher Information Gain signifies substantial uncertainty reduction due to recommendations, while lower values suggest less impactful input.

The total weight is derived as 1 plus the Information Gain, ensuring that the weights sum to 1. Weight for Direct Trust, the reciprocal of the total weight, gauges the importance of direct trust in mitigating uncertainty. Weight for Indirect Trust, calculated as Information Gain divided by the total weight, assesses the role of indirect trust in reducing uncertainty.

In essence, Information Gain measures the influence of trust recommendations from recommending nodes on trust decisions. Higher Information Gain underscores the value of recommendations in reducing uncertainty, indicating a more substantial impact on trust assessments.

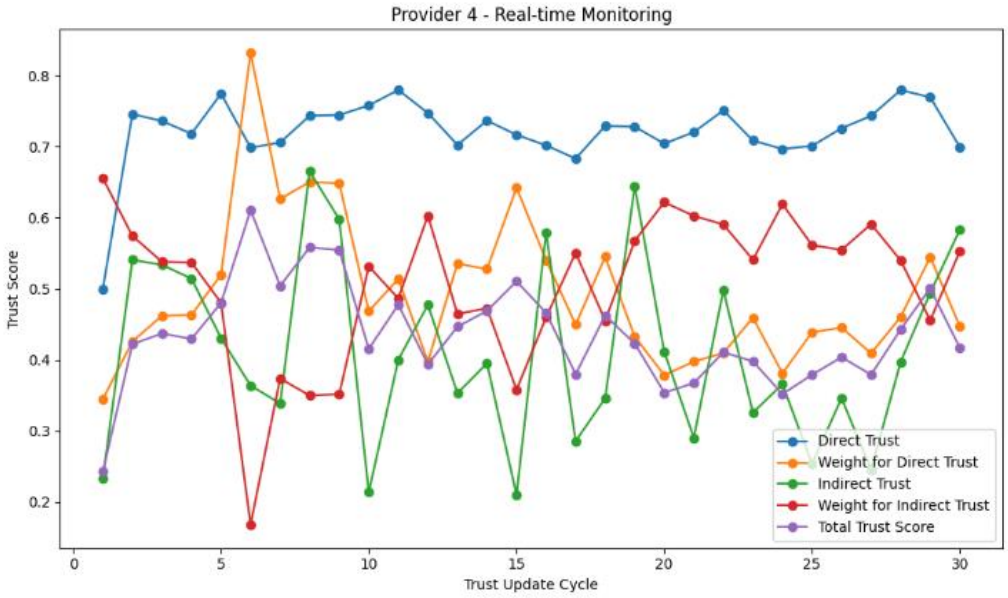

Figure 0.9: Comparison of total trust, direct and indirect trust and the adaptive weights calculated using information gain

From the figure above (figure 5.9)the total trust score is the sum of weighted direct and indirect trust, offering an overall assessment of a service provider's trustworthiness, factoring both direct and indirect trust.

The overall total trust scores calculated by the model are context sensitive both in terms of measuring the capability of the server with regards to the service being requested as well as incorporating recommendations of only those recommenders with whom the service requesting node shares a contextual similarity. This is depicted in the following figures 5.10 and 5.11 where the total trust scores of Service Providing node 3 and 19 show variation according to the type of service being requested.

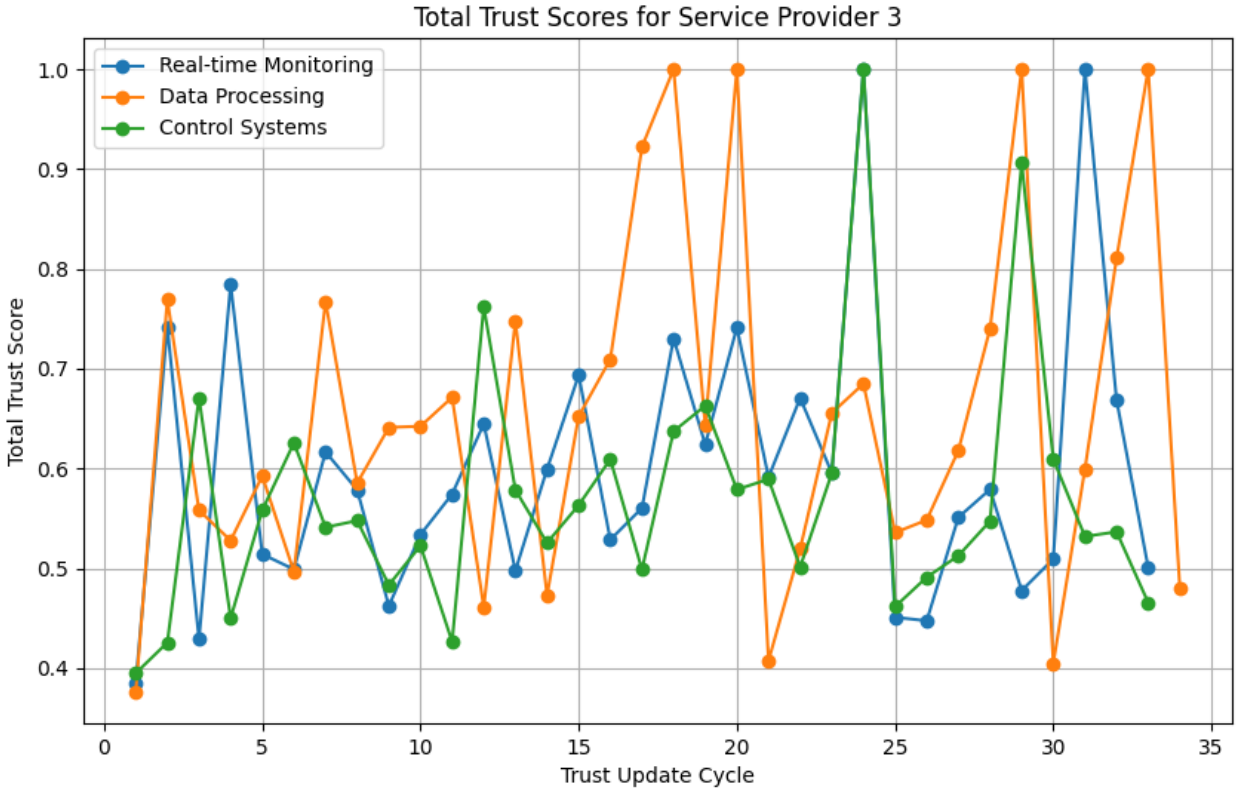

Figure 0.10: Total trust score of Service Provider 3

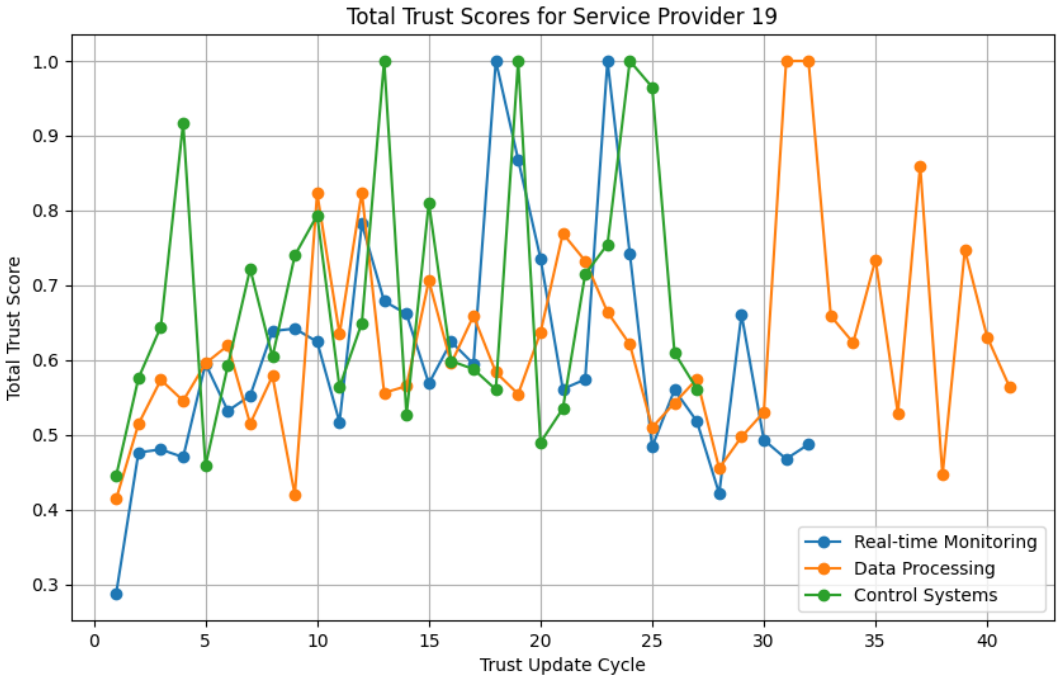

Figure 0.11: Total trust score of Service Provider 19

In conclusion, this chapter has presented a comprehensive evaluation of our proposed trust management solution within the intricate domain of fog-based IoT systems. Our assessment was conducted through mathematical analysis and the use of synthetic datasets generated via a Python simulation, designed to closely mirror the realities encountered in fog-based IoT environments.

We established a network configuration that involved 20 service providers, and trust scores were meticulously calculated for 10 recommenders over the course of 100 iterations. These trust scores were derived from synthetic metrics encompassing critical factors such as response time, packet delivery ratio, and the specific type of service sought. To ensure the accuracy and relevance of our evaluation, we meticulously defined the appropriate ranges for each trust metric, taking into account the unique characteristics of fog-based IoT systems.

An integral feature of our proposed trust management system is its contextual awareness, effectively accommodating the inherent variations in service provider capabilities across diverse service types. This dynamic adaptation of trust scores based on the service type offered by each service provider is visually demonstrated in the graph, highlighting the nuanced trust scores corresponding to different service types provided by a given service provider.

Moreover, we explored various similarity methods to identify the most suitable one for trust modeling. Our evaluation included 20 service providers and 10 recommenders, with the assumption that the service-requesting node interacted with all 20 service providers. The selection of the Jaccard similarity method was driven by its ability to measure similarity by considering shared experiences, such as common services, service providers, and locations. This method outperformed others by providing linear and finely detailed similarity scores among all 50 recommenders.

Our trust management system's adaptability is further exemplified through the utilization of adaptive weights, determined via Information Gain—a concept from information theory. Information Gain quantifies the reduction in uncertainty achieved by incorporating new trust data, ensuring that the weights sum to 1. These adaptive weights are a fundamental component in the computation of total trust, which offers an all-encompassing assessment of a service provider's trustworthiness, integrating both direct and indirect trust aspects.

In essence, our findings underscore the effectiveness and robustness of our proposed trust management system in the intricate landscape of fog-based IoT systems. The ability to adapt to dynamic contexts, the careful selection of similarity methods, and the incorporation of adaptive weights through Information Gain contribute to its reliability and efficacy in assessing trust within these complex IoT environments.

CONCLUSIONAND FUTURE WORK DIRECTIONS

In Fog-based IoT, trust is crucial for reliable data transfer, data security, and Quality of Service assurance. We designed a context-based trust management system based on Bayesian inference and similarity measures. By incorporating context in trust calculations, at both the service provider and recommenders end, the proposed solution effectively calculates a service provider's trustworthiness, factoring both contextually aware direct and indirect trust. We conducted a mathematical assessment of the performance of the proposed model, employing a synthetic dataset designed to simulate a real-time fog-based IoT environment. Following an evaluation and comparison with three other commonly used similarity measures, we selected the Jaccard similarity method. Consequently, we assert that our proposed solution is capable of efficiently computing a context-oriented trust score for fog-based IoT network. In future, we aim to simulate our proposed solution in Contiki Cooja. We also intend to make our trust management system bidirectional with different trust metrics utilized for the calculation of the trust score of the service requester and provider.

BIBLOGRAPHY

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

[2] Jia Guo, Ing-Ray Chen, and Je_rey J.P. Tsai. “A survey of trust computation models for service management in internet of things systems”. Computer Communications, 97:1 14, 2017. ISSN 0140-3664.

[3] Tuva S. Dybedokken, " Trust Management in Fog Computing", Norwegian University of Science and Technology, 2017.

[4] Victor P, De Cock M, Cornelis C, “Trust and recommendations”. In: Recommender systems handbook. Springer, Boston, pp 645–675. 2011.

[5] 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)", 2019 IEEE 5th World Forum on Internet of Things (WF-IoT) pp. 278-282.

[6] A. Arabsorkhi, M. Sayad Haghighi and R. Ghorbanloo, "A conceptual trust model for the Internet of Things interactions," 2016 8th International Symposium on Telecommunications (IST) pp. 89-93.

[7] Mendoza, Carolina VL, and João H. Kleinschmidt. "Mitigating on-off attacks in the internet of things using a distributed trust management scheme", International Journal of Distributed Sensor Networks 11, no. 11 (2015): 859731.

[8] Mohammadi, Venus, Amir Masoud Rahmani, Aso Mohammed Darwesh, and Amir Sahafi. "Trust-based recommendation systems in Internet of Things: a systematic literature review", Human-centric Computing and Information Sciences 9, no. 1 (2019): 1-61.

[9] Marche, Claudio, and Michele Nitti. "Trust-related attacks and their detection: A trust management model for the social IoT", IEEE Transactions on Network and Service Management 18, no. 3 (2020): 3297-3308.

[10] Aiman Almas, “Context based fog computing trust solution for time critical smart healthcare systems”.

[11] Alemneh, Esubalew, Sidi-Mohammed Senouci, Philippe Brunet, and Tesfa

Tegegne, "A two-way trust management system for fog computing." Future Generation Computer Systems 106 (2020): 206-220.

[12] Altaf, Ayesha, Haider Abbas, and Faiza Iqbal. "Context based trust formation using direct user-experience in the Internet of Things (IoT)", In 2019 IEEE International Conference on Cloud Computing Technology and Science (CloudCom), pp. 424-430.

[13] Altaf, Ayesha, Haider Abbas, Faiza Iqbal, Farrukh Aslam Khan, Saddaf Rubab, and Abdelouahid Derhab, "Context-oriented trust computation model for industrial Internet of Things" Computers & Electrical Engineering 92 (2021): 107123.

[14] Hussain, Yasir, Huang Zhiqiu, Muhammad Azeem Akbar, Ahmed Alsanad, Abeer Abdul-Aziz Alsanad, Asif Nawaz, Izhar Ahmed Khan, and Zaheer Ullah Khan, "Context-aware trust and reputation model for fog-based IoT", IEEE Access 8 (2020): 31622-31632.

[15] D.Ravindran, "Fogcomputing: Anefficientplatformforthecloud-resourcemanage- ment," *Journal of Emerging Technologies and Innovative Research*, 2019.

[16] F.Bao,R.Chen,andJ.Guo,"Scalable,adaptiveandsurvivabletrustmanagementfor community of interest based internet of things systems," in *2013 IEEE eleventh in- ternationalsymposiumonautonomousdecentralizedsystems(ISADS)*,pp.1–7,IEEE, 2013.

[17] F.BaoandI.-R.Chen,"Dynamictrustmanagementforinternetofthingsapplications," in *Proceedings of the 2012 international workshop on Self-aware internet of things*, pp. 1–6, 2012.

[18] 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.

[19] A.M.Rahmani,T.N.Gia,B.Negash,A.Anzanpour,I.Azimi,M.Jiang,andP.Lilje-berg,"Exploitingsmarte-healthgatewaysattheedgeofhealthcareinternet-of-things: A fog computing approach," *Future Generation Computer Systems*, vol. 78, pp. 641– 658, 2018.

[20] 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.

[21] A. A. Mutlag, M. K. Abd Ghani, N. a. Arunkumar, M. A. Mohammed, and O. Mohd, "Enablingtechnologiesforfogcomputinginhealthcareiotsystems," *FutureGenera- tion Computer Systems*, vol. 90, pp. 62–78, 2019.

[22] P.VermaandS.K.Sood,"Fogassisted-iotenabledpatienthealthmonitoringinsmart homes," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 1789–1796, 2018.

[23] R. Mahmud, F. L. Koch, and R. Buyya, "Cloud-fog interoperability in iot-enabled healthcare solutions," in *Proceedings of the 19th international conference on dis- tributed computing and networking*, pp. 1–10, 2018.

[24] 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.

[25] " OpenFog- OPCFoundation."Available at:https://opcfoundation.org/markets-collaboration/openfog/.

[26] "IndustryIoTConsortium."Availableat:https://www.iiconsortium.org/index.htm. .

[27] R.VermaandS.Chandra,"Asystematicsurveyonfogsteerediot:Architecture,preva- lentthreatsandtrustmodels,"*InternationalJournalofWirelessInformationNetworks*, vol. 28, no. 1, pp. 116–133, 2021.

[28] S. A. Soleymani, A. H. Abdullah, M. Zareei, M. H. Anisi, C. Vargas-Rosales, M. K. Khan,andS.Goudarzi,"Asecuretrustmodelbasedonfuzzylogicinvehicularadhoc networks with fog computing," *IEEE Access*, vol. 5, pp. 15619–15629, 2017.

[29] S. Prabhdeep and K. Rajbir, "Design and develop quality of service framework using fog computing for smart city applications," *International Journal of Innovative Tech- nology and Exploring Engineering (IJITEE)*, vol. 9, no. 1S, 2019.

[30] A.Kumari,S.Tanwar,S.Tyagi,andN.Kumar,"Fogcomputingforhealthcare4.0environment:Opportunitiesandchallenges,"*Computers&ElectricalEngineering*,vol.72, pp. 1–13, 2018.

[31] A.JosangandR.Ismail,“Thebetareputationsystem,”in*Proceedingsofthe15thbled electronic commerce conference*, vol. 5, pp. 2502–2511, Citeseer, 2002.

[32] Y.Wang,Y.-C.Lu,I.-R.Chen,J.-H.Cho,A.Swami,andC.-T.Lu,“Logittrust:Alogit regression-based trust model for mobile ad hoc networks,” in *6th ASE International ConferenceonPrivacy,Security,RiskandTrust,Boston,MA*,pp.1–10,Citeseer,2014.

[33] J. Yuan and X. Li, “A reliable and lightweight trust computing mechanism for iot edgedevicesbasedonmulti-sourcefeedbackinformationfusion,”*IeeeAccess*,vol.6, pp. 23626–23638, 2018.

[34] M.Apte,S.Kelkar,A.Dorge,S.Deshpande,P.Bomble,andA.Dhamankar,“Gatewa y basedtrustmanagementsystemforinternetofthings,”*REVISTAGEINTEC-GESTAO INOVACAO E TECNOLOGIAS*, vol. 11, no. 4, pp. 4750–4763, 2021.

[35] 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.

[36] F. H. Rahman, T.-W. Au, S. S. Newaz, W. S. Suhaili, and G. M. Lee, “Find my trust- worthyfogs:Afuzzy-basedtrustevaluationframework,”*FutureGenerationComputer Systems*, vol. 109, pp. 562–572, 2020.

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

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

[39] T.Wang,G.Zhang,M.Z.A.Bhuiyan,A.Liu,W.Jia,andM.Xie,“Anoveltrustmech-anismbasedonfogcomputinginsensor–cloudsystem,”*FutureGenerationComputer Systems*, vol. 109, pp. 573–582, 2020.

[40] Y.Winnie,E.Umamaheswari,andD.Ajay,“Enhancingdatasecurityiniothealthcare servicesusingfogcomputing,”in*2018InternationalConferenceonRecentTrendsin Advance Computing (ICRTAC)*, pp. 200–205, IEEE, 2018.

[41] 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.

[42] Y.HussainandZ.Huang,“Trfiot: Trustandreputationmodelforfog-basediot,”in*Internationalconferenceoncloudcomputingandsecurity*,pp.187–198,Springer,2018.

[43] H. Xiao, N. Sidhu, and B. Christianson, “Guarantor and reputation based trust model for social internet of things,” in *2015 International wireless communications and mo- bile computing conference (IWCMC)*, pp. 600–605, IEEE, 2015.

[44] O.B.Abderrahim,M.H.Elhedhili,andL.Saidane,“Ctms-siot: Acontext-basedtrust managementsystemforthesocialinternetofthings,”in*201713thInternationalWireless Communications and Mobile Computing Conference (IWCMC)*, pp. 1903–1908, IEEE, 2017.

[45] U.Jayasinghe,H.-W.Lee,andG.M.Lee,“Acomputationalmodeltoevaluatehonesty in social internet of things,” in *Proceedings of the symposium on applied computing*, pp. 1830–1835, 2017.

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

[47] J.AlMuhtadi,R.A.Alamri,F.A.Khan,andK.Saleem,“Subjectivelogic-basedtrust model for fog computing,” *Computer Communications*, vol. 178, pp. 221–233, 2021.

[48] L. Zahrotun, “Comparison jaccard similarity, cosine similarity and combined both of thedataclusteringwithsharednearestneighbormethod,” *ComputerEngineeringand Applications Journal*, vol. 5, no. 1, pp. 11–18, 2016.

[49] S.Che,R.Feng,X.Liang,andX.Wang,“Alightweighttrustmanagementbased on bayesian and entropy for wireless sensor networks,” *Security and Communication Networks*, vol. 8, no. 2, pp. 168–175, 2015.

[50] “NG, The OS for Next Generation IOT devices.”Available at: https://www.contiki-ng.org/.