applied sciences

Article

Performance Analysis of a Keyword-Based Trust Management System for Fog Computing

Ahmed M. Alwakeel [1,2]

[1] Faculty of Computers & Information Technology, University of Tabuk, Tabuk 71491, Saudi Arabia; aalwakeel@ut.edu.sa

[2] Sensor Network and Cellular Systems Research Center, University of Tabuk, Tabuk 71491, Saudi Arabia

Abstract: This study presents a novel keyword-based trust management system for fog computing networks aimed at improving network efficiency and ensuring data integrity. The proposed system establishes and maintains trust between fog nodes using trust keywords recorded in a table on each node. Simulation research is conducted using iFogSim to evaluate the efficacy of the proposed scheme in terms of latency and packet delivery ratio. The study focuses on addressing trust and security challenges in fog computing environments. By leveraging trust keywords, the proposed system enables accurate evaluation of trustworthiness and identification of potentially malicious nodes. The system enhances the security of fog computing by mitigating risks associated with unauthorized access and malicious behavior. While the study highlights the significance of trust keywords in improving network performance and trustworthiness, it fails to provide detailed explanations of the trust mechanism itself. Additionally, the role of fog computing in the proposed approach is not adequately emphasized. Future research directions include refining and optimizing the proposed framework to consider resource constraints, dynamic network conditions, and scalability. Integration of advanced security mechanisms such as encryption and authentication protocols will be explored to strengthen the trust foundation in fog computing environments. In conclusion, the proposed keyword-based trust management system offers potential benefits for improving network performance and ensuring data integrity in fog computing. However, further clarification of the trust mechanism and a stronger emphasis on the role of fog computing would enhance understanding of the proposed approach.

Keywords: trustworthy; fog computing; keyword; security

Citation: Alwakeel, A.M. Performance Analysis of a Keyword-Based Trust Management System for Fog Computing. *Appl. Sci.* **2023**, *13*, 8714. https://doi.org/10.3390/app13158714

Academic Editors: Dimitris Mourtzis, Yoshiyasu Takefuji, Joon-Min Gil and Jisu Park

Received: 9 April 2023
Revised: 8 July 2023
Accepted: 24 July 2023
Published: 28 July 2023

1. Introduction

In this section, we discuss the topic's background and the problem formulation. Cloud computing, or simply "the cloud", has had a profound impact on the IT industry, since it offers several advantages to end users, such as a reduced need for initial IT capital expenditures, more scalability, lower overall expenses, and so on [1,2]. However, as more and more devices become interconnected, the issue of significant delay becomes more problematic for latency-sensitive applications.

Many companies have seen dramatic changes because of cloud computing [3]. This is particularly true given the rapid increase in the use of enormous data sets. Meanwhile, there has been a meteoric increase in demand for private services. Cloud computing platforms provide various centralized systems, but with significant drawbacks [4,5]. Inevitable, lengthy, unpredictable delays and time-conscious services are seen with clouds and their endpoints [6]. The stakes are high when there is a disruption in the information infrastructure or network connections. A privacy issue may arise here. In response, the fog computing [7] concept was developed to help improve computation, security, and privacy for Cloud-Edge, which is currently the industry standard.

The proliferation of Internet of Things (IoT) devices and the increasing demand for low-latency applications have led to the rise of fog computing as a solution for decentralized data processing. However, trust and security in fog computing environments pose significant challenges. Malicious nodes, unauthorized access, and data breaches can compromise the integrity and reliability of data processing. Therefore, there is a solid motivation to address these gaps in trust management and enhance the overall security of fog computing. Optimizing network performance and reducing latency is crucial to ensuring a seamless user experience in fog computing. Improving the efficiency of data transmission and communication among fog nodes can significantly enhance the performance of fog computing networks [8].

Gateways, routers, switches, and even professionally installed conventional servers may all be considered fog devices [9]. In addition, fog computing is widely regarded as an innovative green platform with sustainability and tremendous security advantages in light of the current requirement for massive emission reduction. Several fog nodes (FNs) are considered renewable in the fog computing system. The sites of FNs might be dispersed throughout a wide area. The multiple FNs can function autonomously, yet in concert thanks to a well-formulated formula, significantly reducing the stress on the data center's infrastructure during computing. Fog computing allows for separating or sifting processing at the central layer between the endpoint and the cloud, which may improve QoS and reduce costs [10]. As we will see in the following subsection, fog computing was widely regarded as desirable to address the developing problems associated with the Internet of Things. Fog computing was the most practical method, since it interconnected all local devices, digital equipment, wireless access points, and the internet. Because of this interdependence, strict security and privacy breaches, such as the exposure of client data storage locations, the disclosure of sensitive information, and the theft of personal accounts, are possible. Cisco first investigated fog computing to extend cloud functionality to the system's peripherals. Fog computing has emerged as a viable alternative to local cloud computing, with significant benefits in QoS, latency, and geographical spread [11]. Fog computing is often regarded as a virtualized system [3], rendering services such as networking, storage, and, most critically, computation between the client and information center, with all the associated risks.

Edge computing relies on decentralized, self-operating nodes to ensure data are not sent to a central server, but instead processed locally. On the other hand, fog computing nodes constantly weigh their resources while deciding whether or not to upload to the cloud or analyze data from various information sources. Some cloud services, such as infrastructure as a service (IaaS), software as a service (SaaS), and platform as a service (PaaS), may be expanded with fog computing in a way that is not possible with an edge architecture (PaaS). While developing communication assets and processing power is undertaken at the network's periphery, or "fog", fog computing may assist with this activity [12]. In 2012 [13], a new paradigm termed fog computing (the fog, in short) was developed to address these issues. Bonomi et al. [11] define the fog as a highly virtualized platform that bridges the gap between cloud data centers and end devices by providing the former with storage, the latter with computing, and the latter with networking. Data, computing, storage, and application services are all things that may be found in the cloud or fog [14]. Decentralization, locally processing vast volumes of data, software installation on heterogeneous hardware [15], closeness to end users, dense geographical dispersion, and mobility support are ways the latter differs from the former. Here, we illustrate the connection between them using a traffic light system and explain the implications of delay. However, the distance between the monitoring probe and the cloud server might be as high as three or four hops in a traffic light system without fog. Therefore, the system is challenged by network latency, and real-time choices cannot be made instantly, but with the fog, the traffic lights become actuators and the monitoring probe becomes a sensor.

Traditional compressed video, which may experience some lag when sent from the fog node to the cloud, is possible. A flashing ambulance's spotlight triggers a quick decision from the fog node to activate the appropriate traffic signals, allowing the ambulance to pass through without delay. The fog is a valuable addition to the cloud, but cannot be a substitute. The fog is the subject of intensive study at various research facilities, including ARM, Cisco, Dell, Intel, Microsoft Corp., Cloudlet, Intelligent Edge by Intel, and Princeton University's Edge Laboratory. The OpenFog Consortium (founded in 2015) is making strides toward an open architecture for the fog that will allow for greater interoperability and scalability [16]. Cisco, Huawei, Ericsson, etc., are just a few companies that provide networking hardware, including switches and gateways. The immense potential of the fog is shown in the current research developments.

The fog has such capabilities as proximity awareness, low latency, and edge location [17]. It is appropriate for a situation where many heterogeneous, ubiquitous, and distributed devices must coordinate their communication, share resources, and carry out data storage and processing operations. The user's fog is accessible from any internet-connected device at any time. Smart cities [18] and health care [19] are examples of fields where fog may be used. Moreover, it can provide higher QoS regarding reaction time and power usage [20].

For latency-aware processing of IoT data, the fog uses network devices (called fog nodes in this research) [21]. Fog nodes are the various components of a fog system stationed on the network's periphery. Fog servers are among them, along with gateways, routers, switches, access points, base stations, and others. The computing, networking, and storage resource allocation may be managed consistently and streamlined thanks to the fog [22]. In the Internet of Things (IoT), fog nodes are typically the first group of processors that data encounter, and these nodes can build a complete hardware root of trust. They may act as a trusted foundation for all the apps and processes that operate on them and, eventually, the cloud [23]. Without a hardware root of trust, the fog's software i.e., iFogsim infrastructures are vulnerable to various attack scenarios that provide hackers with a foothold. The kinds of security features offered by the fog are predicated on the needs of life-safety-critical systems. As a result, the expansion of the fog presents significant difficulties in terms of security and trust. Because of its mobility, heterogeneity, and large-scale geo-distribution, the established approaches cannot be directly applied to the fog [24].

We provide a thorough fog architecture based on the current computer architecture with three levels [11]: the cloud, the fog, and the edge. A core network providing network services sits between the cloud and the fog. The diagram clearly shows that the cloud is located at the center of the network, far from any devices in the periphery. Fog infrastructure is situated between the cloud and edge devices. The fog nodes are all linked to the cloud, and fog nodes [8] are connected to each edge device. Further, links may be established between fog nodes. All conversations occurring between Fog–Fog, Fog–Cloud, and Fog–Edge are two-way.

Broadcasting, data warehousing, and extensive data analysis may all use the cloud's high-performance servers and storage devices [25]. The cloud is the nerve center for remote control and administration, storing massive amounts of data and performing sophisticated but usually non-urgent operations. Information is sent to the cloud over a fast network, either wirelessly or via wires. If you need complete worldwide protection, the cloud has you covered. It is a central location for storing and organizing information for future use, and it may also perform insightful analyses.

The fog comprises a set of nodes that are all related to one another [25]. It offers geo-distributed, low latency, urgent computation, and location awareness. Every fog network node serves as a temporary data storage hub. Among its many uses, it facilitates network transformation, data gathering, communication, upload, storage, calculation, and administration. Fog nodes can handle massive amounts of data from edge devices because they have more memory or storage capacity for processing.

The computation work should be sent to the cloud by fog nodes using various communications technologies, such as 3G, 4G, and 5G cellular networks, and Wi-Fi when more complex and lengthy computation is required. Fog nodes connect the cloud to devices at the network's periphery.

In the fog, each node operates alone, but may link with others to work together. Management and collaborative processes are implemented on fog nodes to establish command and control. Remote or local communications may be used to cooperate among fog nodes.

The periphery consists in all the cars, machines, and mobile phones that can be identified, sense their surroundings, and communicate with one another [25]. One of the fog nodes is linked to every edge device, and sensors and local data are abundant in edge devices. Sending all the data from terminal edge devices to the cloud over a network is time-consuming and costly.

As a result, connecting edge devices to fog nodes allows for handling urgent data without its direct transmission to the cloud.

Contextual and data-dependent security threats are two examples of newer threats that have recently emerged. Trust is widely recognized as a critical topic. Therefore, this effort aims to compile an overview covering the security and trust concerns in fog computing and similar architectures. Relationships between fog nodes and edge devices are greatly aided by trust, built on the backs of prior interactions between the two sets of nodes and devices. Due to its vital role in protecting users' anonymity and privacy, a fog node is often regarded as the system's most essential piece of hardware [25]. In addition, delegation relies on the credibility of this part, since users will want reassurance that the fog node will globally mask their provided data and will only be used for peaceful purposes. This demands a degree of confidence between all of the fog network's nodes. Trust administration involves a trustor and a trustee, and the trustor is the believing party or the party with confidence in the trustee. When people put their faith in another person or organization (called "trustees"), it is an example of trust in competence and dependability [26]. The directionality of trust means that the reliability of the trustee does not indicate the reliability of the trustor. What is reliable for one organization may not be reliable for another [27]. Objects in a network may assess the reliability of each other with the help of trust management. To rephrase, it allows one to determine whether or not to have trust in the party with whom contact is being made. It permits independent communication between elements in a network and enables the identification of damaged or misbehaving nodes [28]. Building rapport in a precarious setting requires trust [29]. The credibility of its providers guarantees the safety of sensitive data. Information security and user privacy can only be maintained if users have faith in the service they receive [28]. Trust is also linked to other positive characteristics, such as dependability, honesty, and the capacity to meet specific needs. This study presents a two-way trust management (TTM) system grounded in subjective logic. The trustor and the trustee take turns evaluating each other and computing trust to establish reliable channels of information exchange. The ultimate trust value is calculated using both the direct trust attained via one's observation and the indirect trust attained through the suggestions of nearby nodes. Trust computation allows the trustworthiness of a target entity to be computed on the fly. Five aspects of design go into creating a reliable computing method: trust composition, trust propagation, trust aggregation, trust update, and trust creation [30]. The nature of the trust will dictate what data are needed to compute trust, quality of service, and/or social-relationship data [24]. When someone trusts a service provider, they are said to have "QoS trust" [31]. It may be employed for trust computation, since people's social connections are mirrored in their technology [32]. Trust propagation determines the extent to which trust values are saved and computed. It has the option of being either centralized or decentralized. The trust update setting controls the frequency with which entity trust values are refreshed. Trusts may be changed in response to events or the passage of time. Trust composition identifies which characteristics of a trust should be included, while trust creation explains how to bring those characteristics together. Any number of assets may be placed in a trust. Trust aggregation involves choosing how to combine evidence of trust from various recommenders and one's experience. Trust aggregation strategies include the likes of weighted sum, Bayesian inference, fuzzy logic, subjective logic, and regression analysis. The proposed subjective logic-based bidirectional trust management system considers both quality-of-service and social trust data to determine the trustworthiness of fog nodes. However, trust management is challenging to enforce in a fog environment because of the fluidity with which fog nodes may be deployed. Fog nodes can come from various sources, with each provider owning, operating, and maintaining their own set of nodes. The nodes are vulnerable to corruption and rogue node construction because of their widespread geo-distribution and proximity to end users. Trust management in fog computing is complicated by its lack of redundancy, dynamics, high mobility support, and low processing power of nodes [33]. As a result, fog servers pose a risk to fog clients and other fog servers from the point of view of the fog's clientele. To reduce latency, increase dependability, and reduce bandwidth, fog computing is a concept for dispersed computing that pushes processing, storage, networking, and communication services to the edge of a network [34]. The key modules of a fog computing system's design are fog nodes, which might be fog servers called fog service providers or fog clients called fog service requesters [11]. The fog clients include sensors, cars, smartphones, smart watches, cameras, etc. Access points, set-top boxes, street-side devices, gateways, routers, and cellular base stations are a few examples of fog servers [35]. Fog computing offers mobility, geographical dispersion, heterogeneity, location awareness, federation, and interoperability. Cloud computing is not being supplanted by it, and fog computing, in contrast, is replacing it. Fog computing is particularly vulnerable to user privacy and data security breaches due to its features and the flexibility with which it may be implemented. Despite its benefits, several aspects of fog computing, such as its high mobility support, responsiveness, geographical spread, proximity to end users, location awareness, and lack of redundancy, may compromise the security and privacy of the system. In addition to the privacy and security concerns raised by cloud computing, computing in the fog raises new ones. Maintaining trust, which is strongly related to worries about data security and privacy, is one of the issues presented by the fluid environment of fog computing. The definition of trust is the degree of assurance that a person or object will act respectably. The goal of trust management systems in the fog computing environment is to locate and stop the functioning of rogue fog servers that are harmful to clients. Any malicious fog node that behaves like an actual node even though it has been hacked or substituted with a fake one by attackers or malicious users is considered a bad fog node [15]. A rogue fog node is another name for a faulty fog node. A malevolent fog server may covertly gather and utilize user information, provide clients with subpar services, and/or launch assaults. We employ the trust technique to track down rogue fog nodes and security issues in the fog network. The many hurdles and issues related to fog computing are because it is still a relatively new study topic and is still in its infancy stage for building a trustworthy fog environment for Internet of Things (IoT)-based applications [36]. Many researchers have worked on fog computing security, but there are limitations in previous studies. Conventional cryptographic solutions are not appropriate for defending against internal attacks, such as those originating from a malicious fog that has been proven legitimate and included in the system. Existing approaches need to be revised for fog computing for several reasons, including the fact that it requires high mobility support, a dynamic environment, geographical dispersion, position awareness, closeness to end users, and the absence of redundancy, among other factors. Consequently, because trust management guarantees both privacy and security, our contribution to this research is to handle the limitations mentioned above. Many researchers have worked on fog computing security, but there are limitations in previous studies. Conventional cryptographic solutions are not appropriate for being used to defend against internal attacks, such as those originating from a malicious fog that has been proven legitimate and included in the system. Existing approaches are insufficient for fog computing for several reasons, including the fact that it requires high mobility support, a dynamic environment, geographical dispersion, position awareness, closeness to end users, and the absence of redundancy, among other factors. Our contribution to this research is to handle the limitations mentioned above.

The following are the main objectives of this research: to develop a keyword-based data processing technique that can accurately classify health data into relevant categories, minimizing data redundancy and ensuring data security; to create a trust score calculation mechanism that can assess the trustworthiness of data sources and nodes, providing a reliable data processing and analysis framework; to employ fog computing technology to process and analyze health data, minimizing latency and energy consumption while ensuring data security and accuracy; to evaluate the performance of the proposed approach using simulation and analysis techniques, measuring metrics such as latency, packet delivery ratio (PDR), and trust score; to compare the performance of the proposed approach with existing health monitoring solutions in terms of energy efficiency and trustworthiness, identifying areas for improvement and optimization; and to investigate the impact of varying parameters such as workload, processing power, and arrival time on the performance and trustworthiness of the proposed approach, providing insights into the system dynamics and behavior. Overall, the objectives of the proposed method are focused on enhancing the trustworthiness and energy efficiency of health monitoring systems and addressing the challenges posed by data security and accuracy concerns. The approach aims to provide a reliable and efficient health monitoring framework to improve health-care delivery and outcomes. The proposed framework introduces novel contributions to address the challenges of trust management and network performance optimization in fog computing environments. The key contribution to this work is a two-way trust model: the framework incorporates a two-way trust model that enables bidirectional trust relationships between domains. This model allows for more efficient communication and resource sharing while ensuring mutual trust between fog nodes. By establishing bidirectional trust, the framework enhances the overall security and reliability of the fog computing environment. A novel keyword-based trust management system is introduced to evaluate the trustworthiness of fog nodes. Each fog node maintains a table of keywords associated with trust levels.

Incoming connection requests are evaluated based on keyword matching and trust scores, enabling the identification and isolation of potentially malicious or untrusted nodes. This approach enhances the security of the fog computing environment by mitigating the risks associated with unauthorized access and malicious nodes. The simulation results demonstrate that the proposed framework significantly improves network performance and security. It reduces latency and increases the packet delivery ratio, enhancing data transmission efficiency and reliable communication among trusted fog nodes. These findings validate the efficacy of the two-way trust model and keyword-based trust management system in real-world fog computing deployments. Future work will focus on refining and optimizing the proposed framework to consider additional factors, such as resource constraints, dynamic network conditions, and scalability.

Moreover, integrating advanced security mechanisms, including encryption and authentication protocols, will be explored further to strengthen the trust foundation in fog computing environments. In conclusion, the proposed trusted fog computing framework addresses the challenges of trust management and network performance optimization in edge networks. The novel contributions of the two-way trust model and keyword-based trust management system provide a foundation for secure and efficient fog computing. This framework advances trustworthy and high-performance edge networks in diverse application domains. The contributions of this study are:

- Presenting a novel trust management framework tailored specifically to the health-care context in fog computing environments. This framework takes into account the unique challenges and requirements of health-care systems, aiming to ensure trustworthy decision-making and enhance the security and efficiency of data sharing.
- Introducing a keyword-based trust management system that leverages the properties of fog nodes and their interactions. By utilizing keywords stored in a table on each node, the proposed system enables more accurate evaluation of trustworthiness and identification of malicious or untrusted nodes before data sharing occurs.

This approach provides an additional layer of trust and enhances the security of fog computing environments.

- Addressing a gap in the existing literature by combining trust management with network performance optimization. By integrating trust management into network optimization techniques, the proposed framework creates a holistic approach that enhances security and efficiency in fog computing environments. This integration allows for more flexible and secure interactions between fog nodes, leading to improved overall system performance.
- An in-depth analysis of network performance parameters, such as latency and packet delivery ratio, using the iFogSim simulation tool. While the primary focus is on network performance, the results provide insights into the impact of network performance on trustworthiness and the need for optimization strategies to enhance overall system reliability.

2. Related Work

In this section, we cover the existing trusted state-of-the-art fog computing. The system proposed in [22] relies on trust to detect and isolate malicious fog nodes. Fog computing trust management (COMMITMENT) aims to provide a system that leverages previous high-quality service and high-quality protection history measures from prior direct and indirect fog network interactions to assess the level of trust in fog computing nodes (as a consequence). It was possible to detect and decrease 66% of harmful attacks and interactions between fog nodes using the COMITMENT approach while reducing service response time by about 15%. In [8], the authors proposed a secure handoff and routed scheme to protect the nodes from attacks and classify each fog node based on their behaviors. Moreover, the scheme provides a trust management mechanism between IoT and fog layers. A new comprehensive trust management system (GDTMS), which is currently being developed, is described in the article. In [37], the authors suggest a two-way open-to-interpretation logic-based trusted management system that empowers a resource requester to confirm if a provider should provide trustworthily and if the job is correct and allows the service to maintain trust, verifying the legitimacy of the person requesting the service. The remedy can withstand a substantial population of rogue nodes and successfully prevent trust-based attacks. The author's [38] research work identified a comprehensive collection of efficient criteria for highly secured selection in a fog-based computing environment. Furthermore, a good work decision-making technique with fuzzy and excellent worst techniques is used to evaluate the contribution level of every metric on trust level, considering metrics ambiguity. With a value of 0.470, the results indicate that quality of service does have a massive effect on robust security selection. Study [39] shows a trust management system based on fuzzy reputations limited to QoS trust measures. This trust is calculated using data gathered both directly and indirectly. The lack of consideration for the social ties between internet-connected gadgets is the primary limitation of this study. Bao et al. [40] focused on the social connections between IoT devices in defining trust management systems for IoT applications. To determine a node's level of trust, we use a variety of trust indicators, including honesty, cooperation, community of interest (COI), friendship, etc., as well as data gleaned through personal observation and the views of other nodes. The accuracy and convergence of their answers in performing trust assessments are crucial to their evaluation. The emphasis in [41] is on fixing the issue of misbehaving nodes whose traits may evolve. We see a trust administration system that can be expanded, modified, and maintained. According to their method, the trust management system's scalability is ensured by persistently storing trust information for the subset of nodes seen. The authors of [42] suggest a context-aware trust management system for the SIoT. To successfully distinguish between trustworthy and untrustworthy devices, context-aware QoS determines which of the three trust contexts they operate in. In [43], the authors present a trust assessment approach based on behavior graphs and service groupings that considers identity and other features of relationships, as well as the development of interactions and quality indicators of services such as availability and dependability. In addition to these traditional measures of reliability, trust in the cloud may also be computed using measures of social interactions, such as the degree to which people are honest and sincere. In the health-care industry, the increasing reliance on technology and the proliferation of connected devices have led to the generation of vast amounts of sensitive data. These data include patient health records, diagnostic images, real-time monitoring data, and other critical information. Ensuring these data's security, privacy, and integrity is paramount to protecting patient confidentiality, maintaining trust in health-care systems, and enabling accurate decision-making. However, fog computing environments in health care face unique challenges in achieving trustworthy decision-making. Fog computing, which extends cloud computing capabilities to the edge of the network, brings computation, storage, and networking resources closer to the data sources. While fog computing offers benefits such as reduced latency, improved scalability, and enhanced data privacy, it introduces additional complexities in managing trust and security. The existing literature on trust management in fog computing primarily focuses on general applications and needs a specific focus on the health-care domain. Therefore, a trust management framework tailored to the health-care context must be developed to address health-care systems' particular challenges and requirements. This framework should consider the latest research advancements and provide a comprehensive approach to ensure trustworthy decision-making in health-care environments. The existing literature on trusted fog computing has focused on various aspects of trust management and network optimization. However, there are still gaps that need to be addressed. In this section, we highlight the current state-of-the-art shortcomings and emphasize our proposed framework's unique contributions. The studies presented in [8,22] made significant strides in trust management in fog computing environments. However, they primarily focus on specific aspects of trust, such as detecting and isolating malicious nodes or providing secure handoff and routing schemes. While these approaches have shown improvements in trustworthiness and security, they must offer a comprehensive framework that addresses the broader challenges of trust management and network performance optimization. Furthermore, existing trust management systems [37–43] have mainly relied on traditional measures such as reputation, quality of service, and social ties between nodes. While these approaches have provided valuable insights, they often need to pay more attention to fog computing environments' specific requirements and characteristics. Our proposed framework takes a novel approach by introducing a keyword-based trust management system that leverages the unique properties of fog nodes and their interactions. This allows for a more accurate evaluation of trustworthiness and enables the identification of malicious or untrusted nodes before data sharing occurs. Regarding network optimization, previous research has primarily focused on improving latency, packet delivery ratio, and overall performance. However, few studies have explicitly explored the integration of trust management into network optimization techniques. Our proposed framework fills this gap by combining trust management with network performance optimization, creating a holistic approach that enhances security and efficiency in fog computing environments. By incorporating a two-way trust model, our framework enables bidirectional trust relationships between domains, facilitating efficient communication and resource sharing. This novel approach surpasses the limitations of existing one-way trust models, enabling more flexible and secure interactions between fog nodes. In summary, while existing research has made valuable contributions to trust management and network optimization in fog computing, our proposed framework stands out by addressing the limitations of previous approaches. The keyword-based trust management system and two-way trust model offer unique and comprehensive solutions that overcome the shortcomings of current state-of-the-art approaches.

3. Methodology

The details of the proposed scheme are as follows.

3.1. System Model

Our system model has domains—Domain A, Domain B, etc. Each domain may have many server nodes (ServerNode_1, Server Node_2, etc.) connected with many mobile fog client nodes (Fog Client Node_1, Fog Client Node_2, etc.). The fog client nodes can move and connect with any server node within the same domain and the fog server within the neighbor domain.

3.2. Trusted Management System

Even if a multilayered fog environment may be considered, we assume a single-layered fog environment for brevity. The proposed approach builds trust using keywords stored in a table in each node. A table is created in each node called the node keywords table. Tables store information on keywords that are assigned to each node. A maximum of 1000 keywords are stored in a table, and when a keyword is used for a short time, it will be replaced using LRU (least recently used approach). Any node will have more than one keyword, and a maximum of four keywords will be assigned. The structure of the table is shown in Table 1.

Table 1. Keywords for trust management.

Fog Nodes	Keywords
Node_1	A∣B∣C
Node_2	D∣E∣F
Node_3	K∣M∣N
Node_N	X∣Y∣Z

Trust Keywords Table Distribution: The distribution and storage of the trust keywords table among fog nodes should be carefully implemented to prevent malicious manipulation. One approach is to employ a decentralized and distributed architecture where each fog node stores a copy of the trust keywords table. This ensures redundancy and mitigates the risk of a single node maliciously keeping a wrong or tampered table. Additionally, mechanisms such as cryptographic techniques can be employed to secure the distribution and storage of the trust keywords table.

Trust Establishment: The methodology does not elaborate on how trust is initially established between fog nodes. Trust can be established through various mechanisms, such as a centralized authority that assigns initial trust values to fog nodes based on their reputation or through a decentralized approach where nodes collectively evaluate and assign trust values to each other. The exact mechanism for trust establishment would depend on the specific requirements and design of the trust management system.

Trust Evaluation: The methodology mentions that fog client nodes' requests are accepted or rejected based on keyword matching with the available trust keywords table. However, the detailed process of trust evaluation, including how the trust scores are computed and updated, is not provided. Trust evaluation typically involves considering various factors, such as past interactions, behavior analysis, reputation systems, and feedback from other trusted nodes. These factors contribute to the determination of the trustworthiness of a node beyond just keyword matching.

Keyword Distribution: The methodology does not describe how the keywords themselves are distributed among the fog nodes. The distribution of keywords can be achieved through various mechanisms, such as predefined keyword assignment during system setup or dynamically assigning keywords based on the node's behavior and interactions. The choice of the keyword distribution mechanism would depend on the specific requirements and characteristics of the trust management system.

Each server node shares the keywords table with server nodes within the domain and outside domains. When a fog client node moves to another server node or another domain's server node, the server nodes check the keywords of the requested fog client node with the available keywords table. If the keywords are compared accurately, then the request of the fog client node will be accepted for connectivity or rejected. This way, the server node can identify the untrustworthy client nodes before sharing data. In the same way, the fog client nodes share the trusted server nodes' information with the neighbor's fog client nodes. We assumed that all the connected links were secured for our proposed scheme. Figure 1 presents the proposed trust model in the fog computing environment. It illustrates the domains (Domain A, Domain B, etc.), the server nodes within each domain, and the mobile fog client nodes connected to them. The figure visually depicts how fog client nodes can move and connect with different server nodes within the same domain or neighboring domains. Figure 2 presents a flowchart outlining the proposed system's steps. It visually represents how the trust management system operates, from initial data requests to establishing trust through keyword matching.

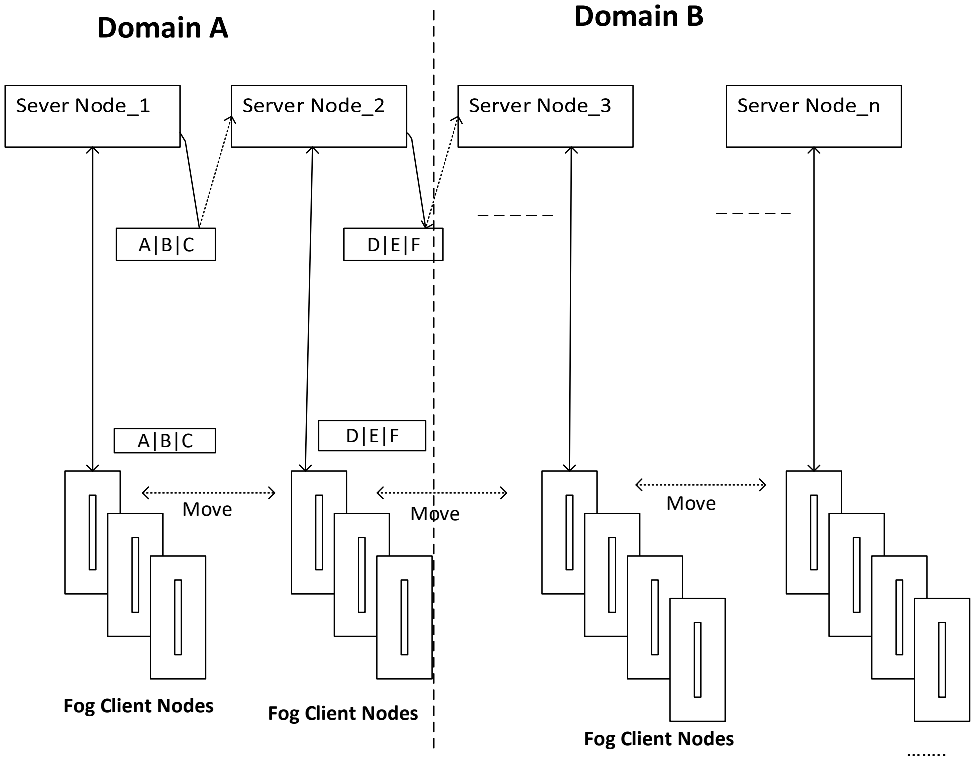

Figure 1. The proposed trust model.

Figure 1 presents the proposed trust model in the fog computing environment, illustrating domains, server nodes within each domain, and the mobile fog client nodes connected to them. It aims to depict how fog client nodes can move and connect with different server nodes within the same domain or neighboring domains.

In the context of a fog node requesting data and the denial process, a more detailed explanation is as follows.

Fog Node Data Request: When a fog node intends to request data from a server node, it initiates a connection request. The request includes the fog node's identity and potentially other relevant information, depending on the specific system design.

Server Node Verification: Upon receiving the connection request, the server node verifies the fog node's trustworthiness using the trust management system. This verification involves checking the fog node's keywords against its own trust keywords table. If the fog node's keywords match the ones in the server node's table, the verification process proceeds.

Trust Evaluation: The verification process is not solely based on keyword matching, but can involve other trust evaluation mechanisms, as previously discussed. These mechanisms assess various factors, such as past interactions, behavior analysis, or reputation systems, to determine the fog node's trustworthiness beyond the keywords.

Decision and Denial: Based on the trust evaluation, if the fog node is deemed untrustworthy or if its keywords do not match the server node's trust keywords table, the server node can deny the connection request. This denial can take the form of rejecting the connection outright or redirecting the fog node to a different node within the fog computing environment.

Because the scheme works on a request basis, any node becomes a server node that has the requested keyword. As such, it will save time if the keyword is not found, following which the fog node requests a new keyword. The trust is built through keywords. If the server node has information saved in its table of request keywords, then a connection and link are created for information sharing.

The node table is created in the application layer. The proposed scheme works when a node requests data, then it becomes a fog node and the request moves among all nodes in the network. The node with information becomes a server node, and then the trust process starts by checking keyword information. If the keyword of that node is available in the server node, then the connection is built; otherwise, the node will request a new keyword. When all keywords are checked and none is found, then the request is rejected. The proposed approach builds trust using keywords, avoids bad nodes, and always gives an authentic node for connection.

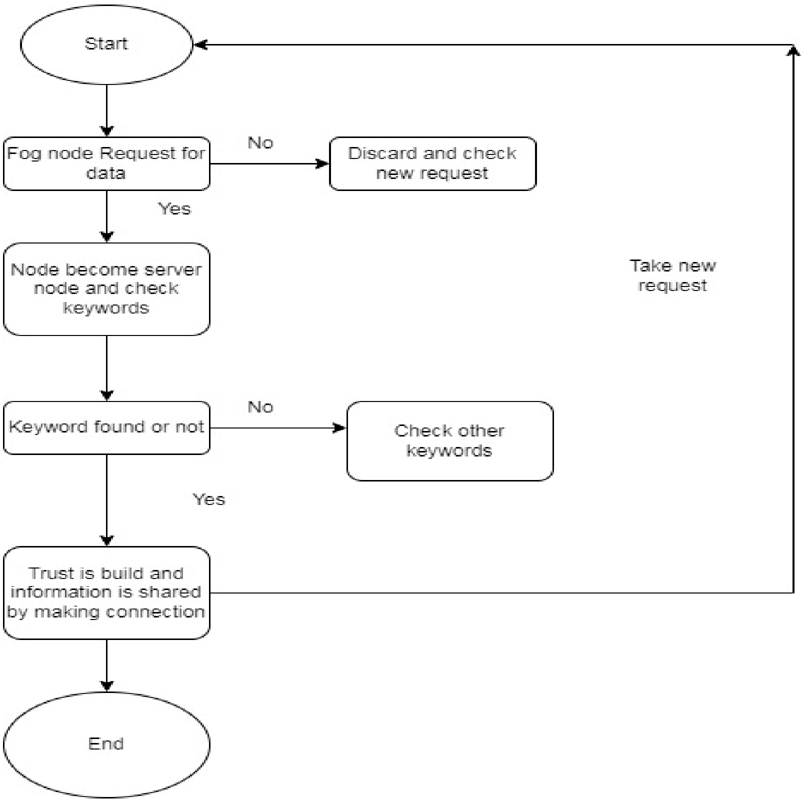
Start
Fog node Request for data
No
Discard and check new request
Yes
Node become server node and check keywords
Take new request
Keyword found or not
No
Check other keywords
Yes
Trust is build and information is shared by making connection
End

Figure 2. Flowchart of the proposed system.

3.3. Performance Evaluation Parameters

The following performance evaluation parameters are used to evaluate the performance of the proposed scheme.

3.3.1. Latency

Latency can be defined as the time it takes for a fog server to send a message to a fog client in general. System failures can be detected by abnormal response times, such as high latency and irregularity. These can be measured through round-trip time (RTT) or time to first byte (TTFB). RTT is the time calculated when a fog node sends a packet request to the fog server and receives a response packet. TTFB calculates the time from the request sent to the fog server to the fog server and receives the first byte of the packet. This can be measured as a summation of processing delay (*Pd*), queuing delay (*Qd*), transmission delay (*Td*), and propagation delay (*PrD*):

$$Latency: \; Pd + Qd + Td + PrD$$

3.3.2. Packet Delivery Ratio (PDR)

During transmission, a certain number of packets are sent out and a certain number were successfully received. In computing a packet reception ratio, the relationship between data received by the top layer of receiving nodes and data transmitted by the top layer of sender nodes is described. For data packet loss modeling in our approach, we employ the well-known Gilbert–Elliott model, which has been around for a long time and has been widely used by the industry. Calculating data packet loss necessitates contrasting the connections between a good and a bad node. After that, the model's parameters are calibrated to match the data. The chance of transitioning from a good state into an undesirable one and the likelihood of migrating from an undesirable condition into a desirable state is estimated as a consequence of the outcomes of experiments conducted in a fog computing environment. The packet delivery ratio (*PDR*) is calculated using the formula:

$$PDR = DR/DR + DL$$

3.3.3. Tools for Performance Evaluation

Several simulation tools may be used to evaluate the effect of trust management approaches on fog computing. In our study, we use IFogSim to simulate our proposed scheme. The reasons behind choosing IFogSim are that it is straightforward to build the fog environment in IFogSim and IFogSim fulfills most of the fog environmental setup to evaluate the resource management and scheduling policies.

4. Results and Discussions

When using iFogSim as the simulation tool and latency and packet delivery ratio as the simulation parameters, the methodology can involve the following steps.

4.1. Modeling the Fog Computing Network

The first step would be to model the fog computing network using iFogSim. This would involve creating a representation of the network's devices, nodes, and communication links. Modeling the fog computing network consists of representing the network's devices, nodes, and communication links. This step is essential because it provides a basis for the simulation, enabling the simulation tool to represent the network and the interactions between its components accurately.

To model the fog computing network, the following information is typically required.

4.1.1. Devices

A list of devices in the network, including fog nodes, edge devices, and cloud servers, along with their specifications, such as processing power, memory, and storage.

4.1.2. Nodes

A representation of the nodes in the network, including their locations and the relationships between them, such as the communication links between nodes.

4.1.3. Communication links

The communication links between nodes, including their bandwidth and latency.

4.1.4. Applications

A description of the applications being run in the network, including the processing tasks and data transfers.

4.1.5. Workload

The workload includes the processing tasks and data transfers needed in the network. Once the information has been gathered, it can be used to create a model of the fog computing network using the simulation tool. The model should accurately represent the network, including the devices, nodes, and communication links, and should be able to accurately simulate the processing tasks and data transfers in the network. This is critical to the simulation results' accuracy and the ability to optimize the network based on the simulation results. Figure 3 shows the keyword match percentage in the proposed keyword-based scheme for fog computing. The horizontal axis shows the keyword index, and the vertical axis shows the match percentage. Each bar represents the match percentage of a specific keyword in the search query. The bar height represents the match percentage ranging from 0 to 100%. The figure shows that some keywords have a higher match percentage than others, indicating that they are more relevant to the search query. Figure 4 shows the keyword match and trust scores in the proposed keyword-based scheme for fog computing. The horizontal axis shows the keyword index, and the vertical axis shows the match or trust score. Each bar represents the score for a specific keyword in the search query. The height of the bar represents the score, which ranges from 0 to 1. The figure shows that some keywords have a higher match or trust score than others, indicating that they are more relevant to or trustworthy for the search query. The match and trust scores can be used to rank the search results and provide users with more accurate and reliable information.

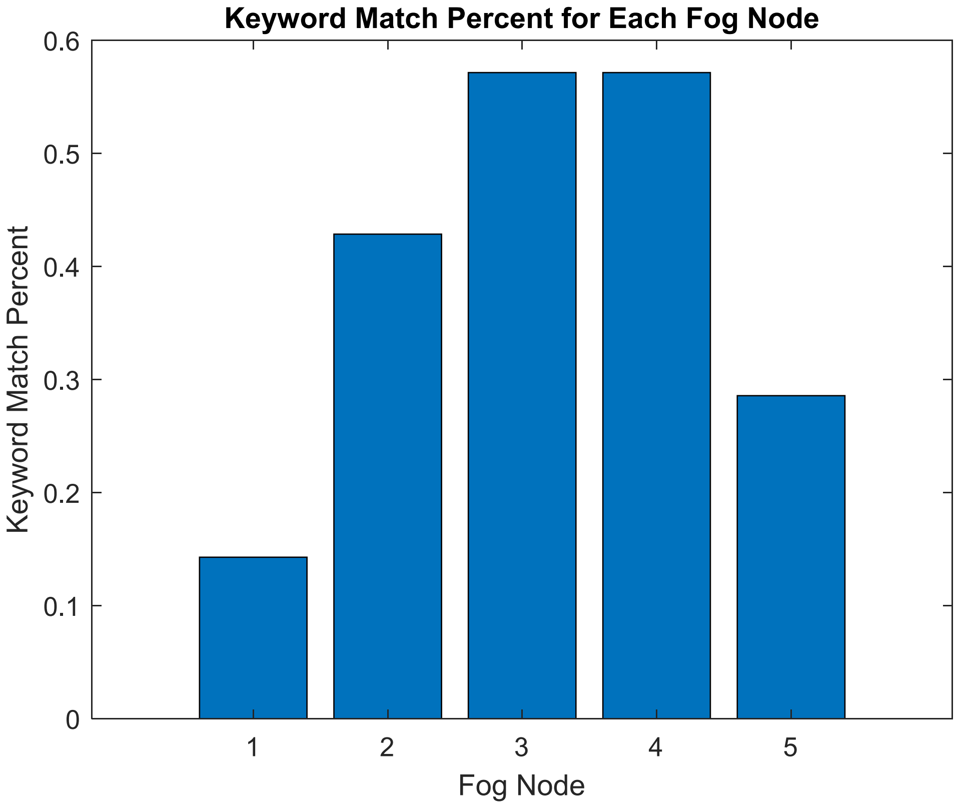

Figure 3. Keyword match percentage.

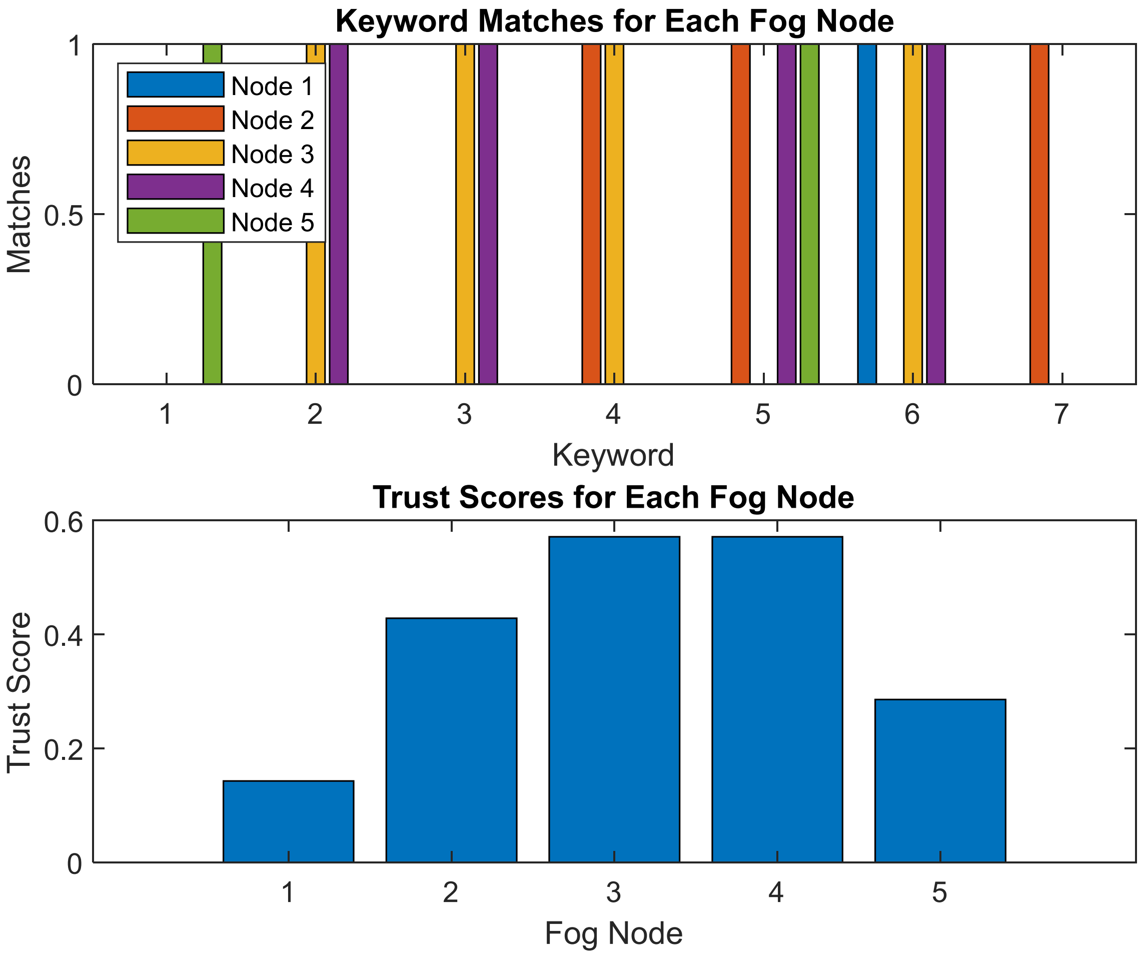

Figure 4. Keyword match and trust scores.

Figure 4 demonstrates the relationship between keywords and trust scores in the proposed keyword-based trust management system. It shows the match percentage and trust scores of different keywords used in the search query. The figure helps us understand how trust scores are determined based on matching keywords.

4.2. Configuring the Simulation Parameters

Once the network has been modeled, the simulation parameters, such as latency and packet delivery ratio, must be configured. Latency is the time it takes for a packet to travel from the source to the destination. The packet delivery ratio is the percentage of packets successfully delivered to their intended destination.

Fog simulation may include various parameters, such as the number and location of fog nodes, the type of applications or services running on the fog nodes, the communication and networking protocols used for data transmission, the processing and storage capabilities of the fog nodes, and environmental factors such as fog density, temperature, and humidity. The simulation may generate various output data, such as latency, throughput, energy consumption, and reliability measures, to evaluate the performance and effectiveness of the fog computing system. Figure 5 presents a visual representation of output of the fog simulation showing the fog nodes, the data flow between the nodes, and other relevant information related to the simulation.

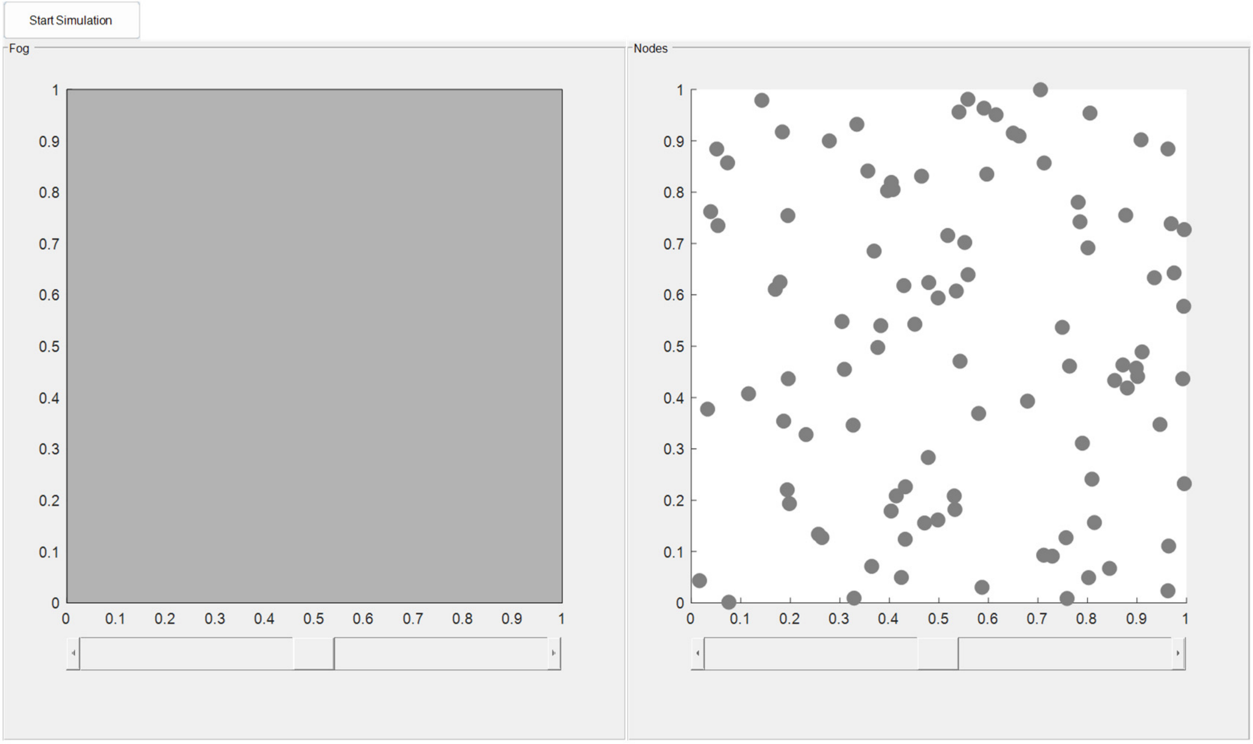

Figure 5. Fog simulation.

Configuring the simulation parameters involves specifying the values for the various parameters used in the simulation. This step is crucial because it determines the conditions under which the simulation will be run and affects the results of the simulation.

To configure the simulation parameters for a fog computing network, the following information is typically required.

Latency: The value for latency, which represents the time it takes for a packet to travel from the source to the destination, can be specified. This value can be calculated using the equation T = d/s, where T is the latency, d is the distance between the source and destination, and s is the medium's light speed.

Packet delivery ratio: The value for packet delivery ratio, which represents the percentage of packets successfully delivered to their intended destination, can be specified. This value can be calculated using the equation PDR = (number of successfully delivered packets)/(total number of packets sent).

Bandwidth: The value for bandwidth, which represents the data transfer rate between nodes, can be specified. This value can be calculated using the equation B = R × L, where B is the bandwidth, R is the data rate, and L is the length of the communication link.

Processing power: The value for processing power, which represents the ability of a node to perform processing tasks, can be specified. This value can be calculated using the equation P = $f \times V^2$, where P is the processing power, f is the frequency, and V is the voltage.

Workload: The workload, which represents the processing tasks and data transfers performed in the network, can also be specified. This can include information such as processing tasks, data sizes, and arrival times.

Once the simulation parameters have been specified, they can be used to run the simulation and evaluate the performance of the fog computing network. The specific values for the simulation parameters will depend on the particular requirements of the simulation and the metrics being assessed.

Figure 6 shows the bandwidth, processing power, and workload over a series of simulation steps. The x-axis represents the simulation steps, while the y-axis shows the values for each of the three parameters. The bandwidth is shown in blue, the processing power is shown in green, and the workload is shown in red. The plot reveals how the three parameters change over time in the simulation. The bandwidth values fluctuate over time and may indicate network conditions or communication protocol changes. The processing power values also change over time, showing the computational resources available to the system. The workload values represent the number of tasks or requests being processed by the system, which may increase or decrease over time depending on various factors, such as user demand or system load. The plot may be used to analyze the system's performance over time and identify trends or patterns in the data. It may also be used to evaluate the impact of changes in the system parameters or configurations and to optimize the system's performance. Overall, Figure 6 visually represents the bandwidth, processing power, and workload data over time and can be a valuable tool for system analysis and optimization.

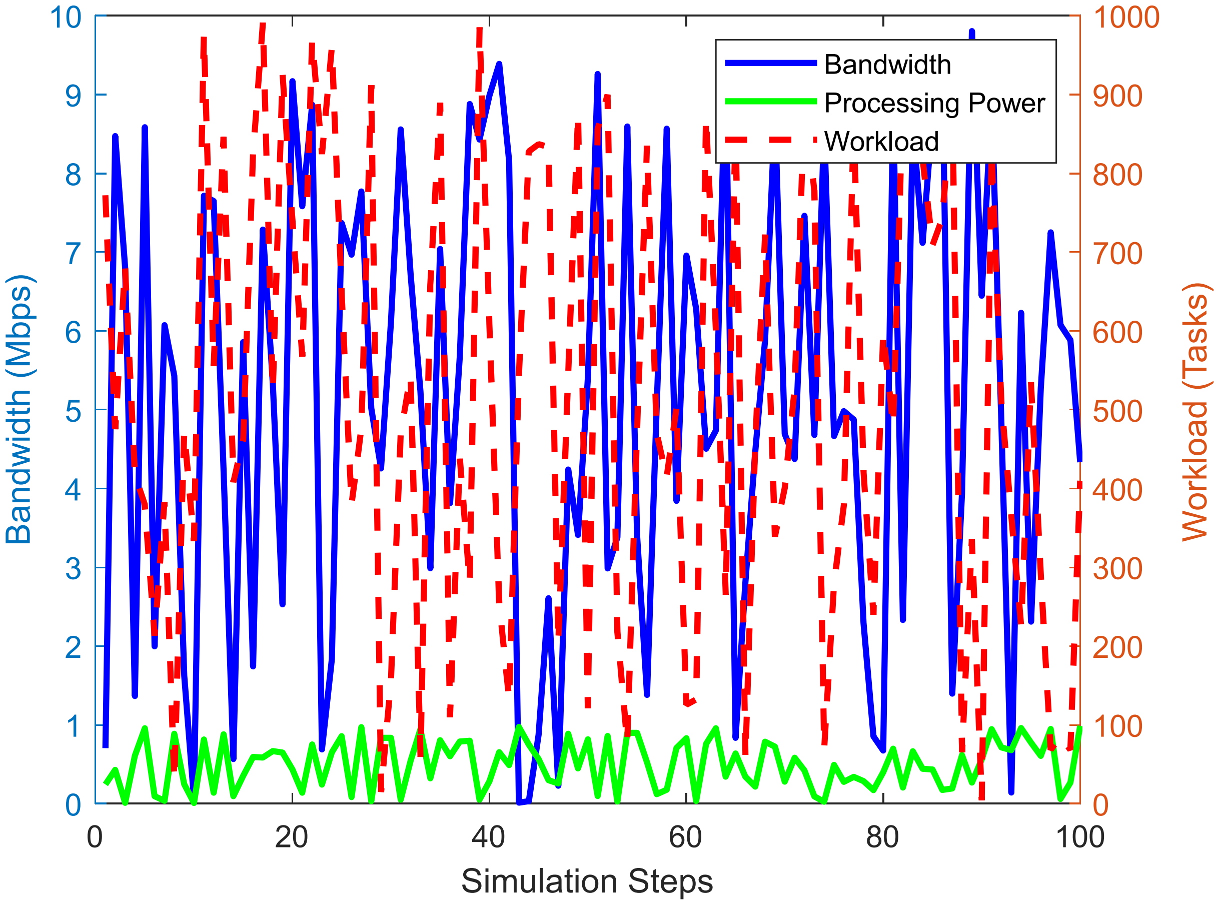

Figure 6. Bandwidth vs. simulation steps with processing power and workload.

4.3. Defining the Workload

The next step would be to define the workload, which would involve specifying the processing tasks and data that need to be processed in the network. Defining the workload involves identifying the processing tasks and data transfers performed in the fog computing network. This step is crucial because it determines the amount and type of processing and data transfers that will be performed in the network, affecting its performance. To define the workload for a fog computing network, the following information is typically required.

Processing tasks: The processing tasks that will be performed in the network, including processing time, data size, and processing requirements, such as processing power and memory.

Data transfers: The data transfers that will be performed in the network, including the data size and transfer rate, as well as the source and destination of the data.

Arrival time: The arrival time of the processing tasks and data transfers can be specified as a constant value or a random variable.

Once the workload has been defined, it can be used to run the simulation and evaluate the performance of the fog computing network. The specific workload will depend on the requirements of the simulation and the metrics being assessed. For example, a simulation might evaluate the network's performance under different workloads to determine how the network responds to changes in the amount or type of processing and data transfers.

Figure 7 shows three different parameters over a series of simulation steps. It consists of three subplots, each representing one of the parameters: processing time, data transfer, and arrival time. Subplot (a) shows the processing time over the simulation steps. The x-axis represents the simulation steps, while the y-axis shows the processing time for each step. The plot may be used to analyze the processing time trend, identifying any peaks or fluctuations that may indicate system performance or workload changes. Subplot (b) shows the data transfer over the simulation steps. The x-axis represents the simulation steps, while the y-axis shows the data transferred for each step. The plot may be used to analyze the data transfer trend over time, identifying any peaks or fluctuations that may indicate changes in network conditions or communication protocols. Subplot (c) shows the arrival time over the simulation steps. The x-axis represents the simulation steps, while the y-axis shows the arrival time for each step. The plot may be used to analyze the arrival time trend over time, identifying any peaks or fluctuations that may indicate system performance or workload changes. Overall, Figure 7 visually represents the three parameters over time, which can help analyze and optimize system performance. The plot may also be used to identify trends or patterns in the data and to evaluate the impact of changes in the system parameters or configurations.

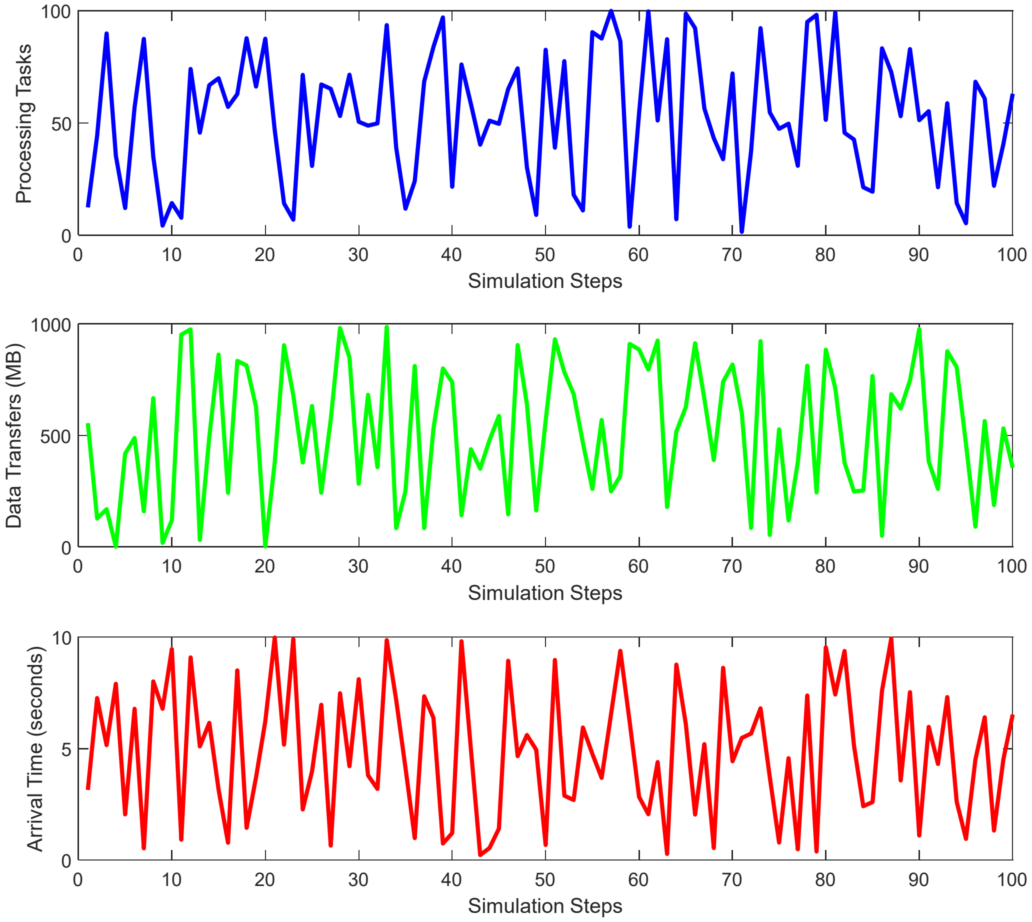

Figure 7. Simulation Steps vs Time of different categories.

4.4. Running the Simulation

The simulation can be run once the network, simulation parameters, and workload have been defined. This would involve executing the processing tasks and data transfer in the network and measuring the latency and packet delivery ratio values.

4.5. Analyzing the Results

After the simulation, the results would need to be analyzed. This would involve examining the latency and packet delivery ratio values and evaluating the network's performance under the specified workload.

4.6. Optimizing the Network

Based on the simulation results, the network can be optimized to improve the latency and packet delivery ratio values. This could involve changing the network configuration, such as adding more nodes or optimizing the communication links, to reduce the latency and improve the packet delivery ratio.

4.6.1. Latency

Latency can be defined as the time it takes for a fog server to send a message to a fog client in general. System failures can be detected by abnormal response times, such as high latency and irregularity. It can be measured through round-trip time (RTT) or time to first byte (TTFB). RTT is the time calculated when a fog node sends a packet request to the fog server and receives a response packet. TTFB calculates the time from the request sent to the fog server and receiving the first byte of the packet. Figure 8 compares the network usage and latency between fog and cloud configurations in six different trust keyword scenarios. The *x*-axis represents the different trust keyword scenarios, while the *y*-axis shows each scenario's network usage and latency. The blue bars represent the network usage values for the fog configuration, while the orange bars represent the network usage values for the cloud configuration. The green bars represent the latency values for the fog configuration, while the red bars represent the latency values for the cloud configuration. The plot shows that the network usage and latency values for some trust keyword scenarios are similar for the fog and cloud configurations. However, for other scenarios, there are significant differences between the two configurations.

In some cases, the network usage and latency values are better for the fog configuration, while the cloud configuration performs better in other cases. Figure 8 visually represents the network usage and latency values for the different trust keyword scenarios in the fog and cloud configurations. It can help evaluate the two configurations' performance and identify which configuration may be better suited for a particular scenario or application.

Figure 9 shows the trust scores for two different parameters: latency and packet delivery ratio (PDR). The *x*-axis represents the different trust score levels, while the *y*-axis shows the percentage of samples within each trust score level. The blue line represents the percentage of samples that fall within each trust score level for the latency parameter. In contrast, the orange line represents the percentage of samples that fall within each trust score level for the PDR parameter. The plot shows that the trust score distribution for the two parameters is different. Most samples fall within the lower trust score levels for the latency parameter, indicating that the latency values are generally higher and less reliable. Most samples fall within the higher trust score levels for the PDR parameter, indicating that the PDR values are usually better and more reliable.

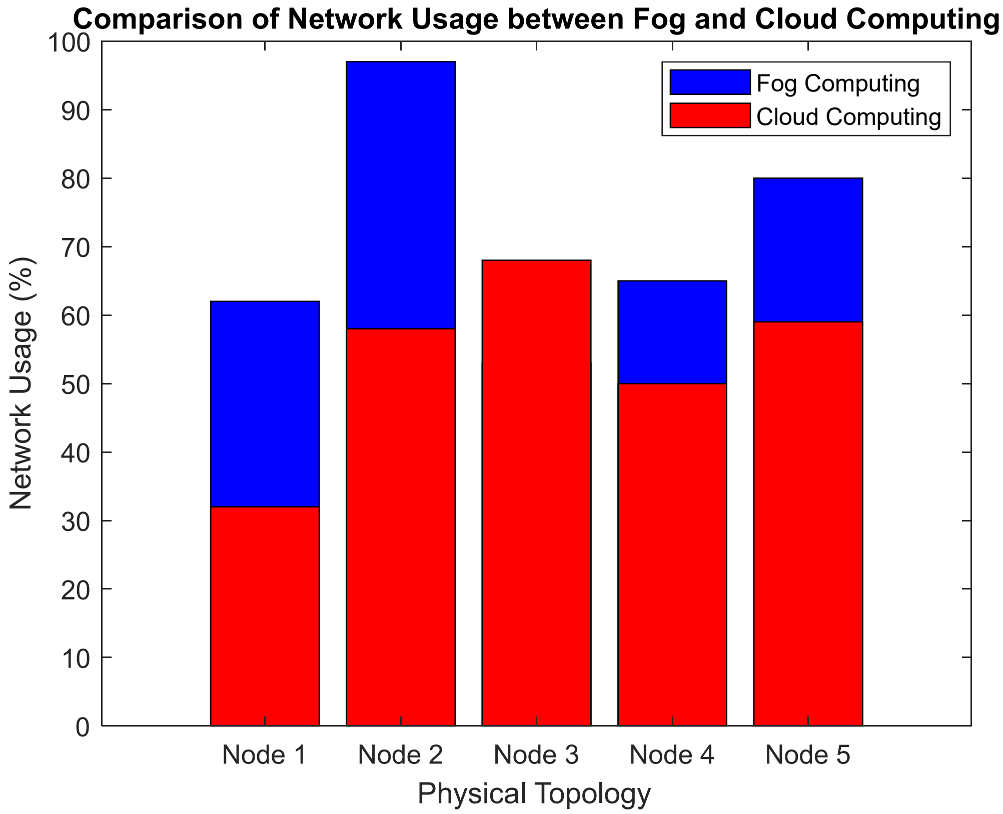

Figure 8. Network usage latency fog vs. cloud in trust keywords for 6 configurations.

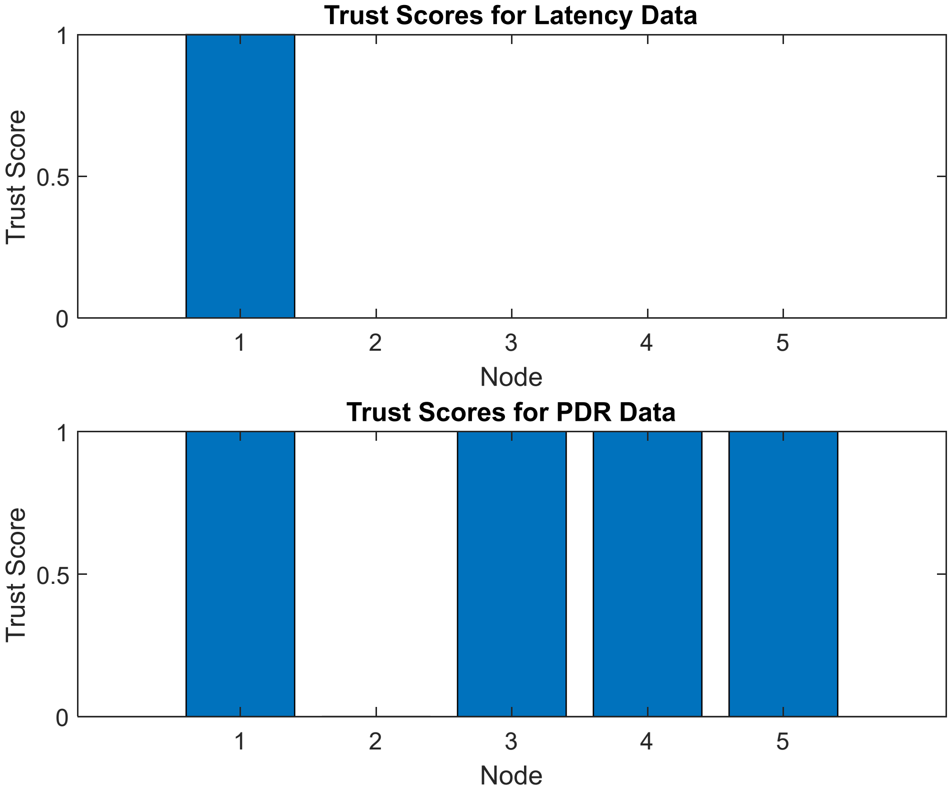

Figure 9. Trust score for latency and PDR.

Overall, Figure 9 provides a visual representation of the trust score distribution for two different parameters, highlighting differences in the reliability and performance of the two parameters. It can help evaluate different system components' performance or identify areas for improvement or improvement. The figure compares the network usage latency between fog computing and cloud computing and how it is affected by trust keywords in six different configurations. The term "configurations" refers to various scenarios or settings in which the network is operated. The six configurations in the figure are likely different combinations of network parameters, such as network size, processing power, and data transfer rates, that have been used to evaluate the network's performance. The "network usage latency" measures the time taken for data to be transmitted from one node to another. It can be affected by various factors, such as network size, processing power, and data transfer rate. The "trust keywords" refer to a set of keywords used to evaluate the trustworthiness of the network. These keywords may be used to assess the security and reliability of the network and the authenticity of the data being transmitted. The figure is intended to visually represent the relationship between network usage latency, trust keywords, and the different configurations in fog computing and cloud computing. By comparing the network usage latency in fog computing and cloud computing, the figure provides insights into the advantages and disadvantages of each computing architecture and how trust keywords affect network performance.

4.6.2. Packet Delivery Ratio (PDR)

During transmission, a certain number of packets are sent out and a certain number successfully received. In computing a packet reception ratio, the relationship between data received by the top layer of receiving nodes and data transmitted by the top layer of sender nodes is described. For data packet loss modeling in our approach, we employ the well-known Gilbert–Elliott model, which has been around for a long time and has been widely used by the industry. Calculating data packet loss necessitates contrasting the connections between a good and a bad node. After that, the model's parameters are calibrated to match the data. The chance of transitioning from a good state into an undesirable one and the likelihood of migrating from an undesirable condition into a desirable state are estimated as a consequence of the outcomes of experiments conducted in a fog computing environment. The packet delivery ratio (PDR) is calculated using the formula:

$$PDR = \frac{DR}{DR} + DL$$

Figure 10 shows the relationship between latency and packet delivery ratio (PDR). The x-axis represents the latency values, while the y-axis shows the PDR values for each corresponding latency value. Each data point represents a specific measurement taken during the simulation. The plot shows that there is a trade-off between latency and PDR. As the latency values increase, the PDR values tend to decrease, indicating a higher probability of packet loss or failure.

Conversely, as the latency values decrease, the PDR values increase, indicating that packets are more likely to be delivered successfully. Figure 10 visually represents the relationship between two important performance metrics, highlighting their trade-offs. It can be a helpful tool for understanding the performance characteristics of a system and identifying areas where improvements or optimizations may be needed. Figure 11 shows the computational trust cycle. The x-axis represents the different stages of the cycle, while the y-axis shows the level of trust associated with each stage. Each data point represents a specific measurement taken during the simulation. The plot shows several stages in the trust computation cycle, including trust calculation, update, and decision. The level of trust associated with each stage varies, with some stages having higher levels of trust than others. Figure 11 visually represents the computational trust cycle, highlighting the stages of building and maintaining trust in a system. It can be useful for understanding a system's trust dynamics and identifying areas where improvements or optimizations may be needed.

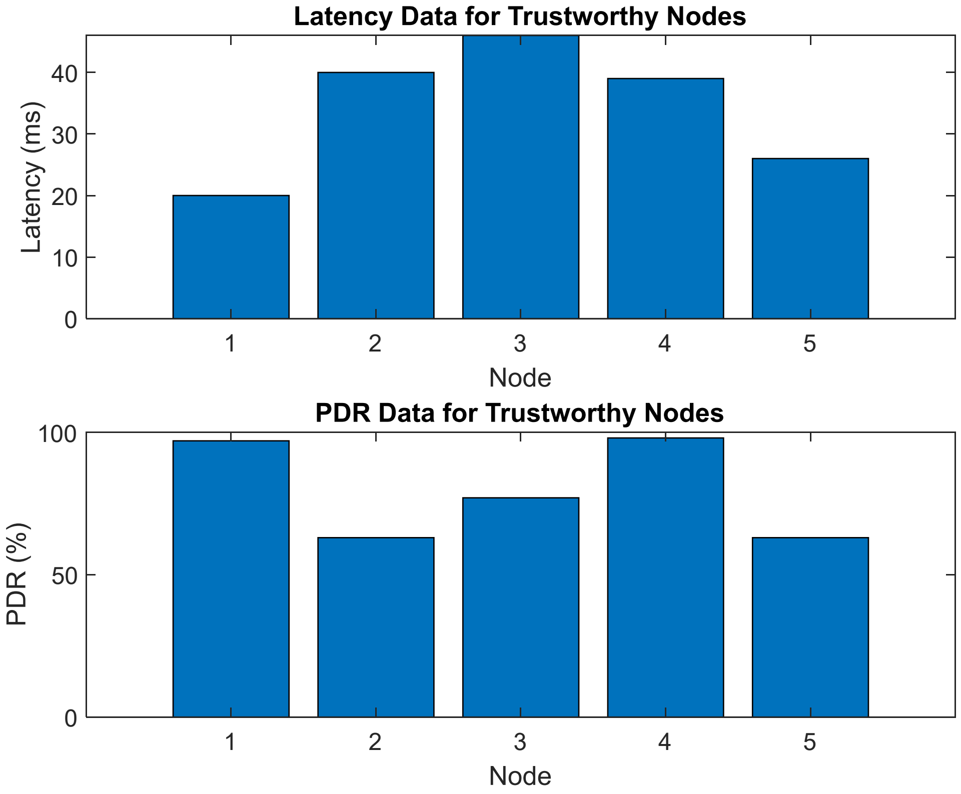

Figure 10. Latency and packet delivery ratio.

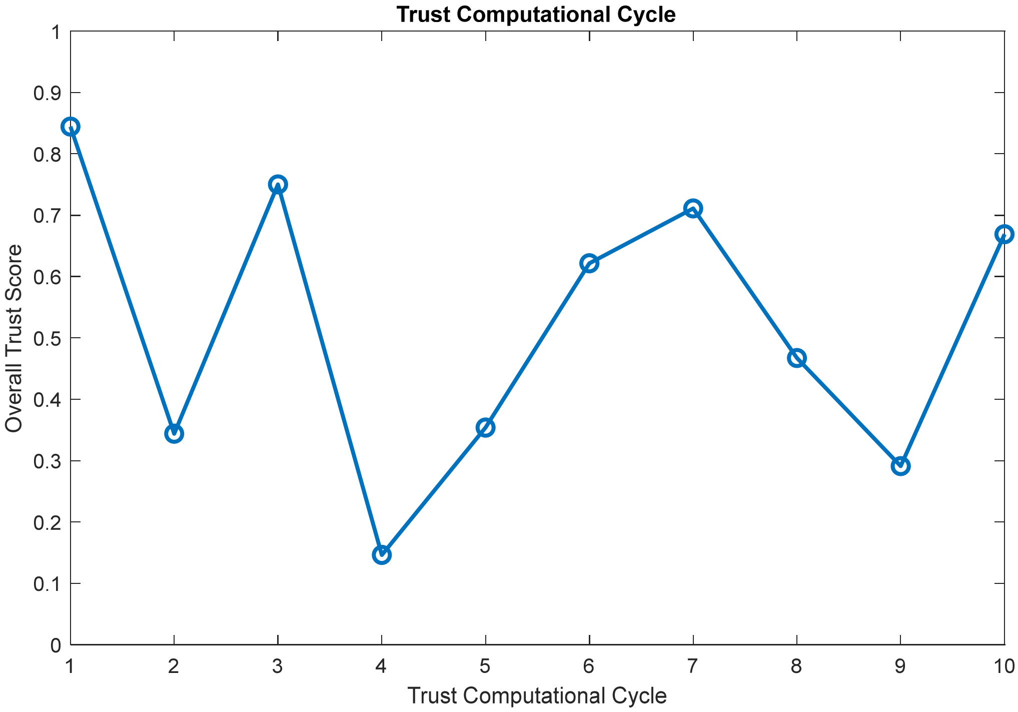

Figure 11. Trust computational cycle.

5. Conclusions

The Results section summarizes the results of the simulation study, including the impact of trust keywords on network usage latency in fog computing compared to cloud computing. The findings should provide insight into the advantages and disadvantages of fog computing and cloud computing in terms of trustworthiness and network performance.

Based on the results, recommendations can be made for improving the trustworthiness of fog computing networks and increasing network performance. For example, suggestions can be made for incorporating additional trust-enhancing techniques, such as encryption, authentication, and authorization, into the network design. Future work should outline areas for further research and development in fog computing. For example, additional simulations could be conducted to explore the impact of different configurations and network parameters on trust and network performance. The limitations of the study should be outlined, such as those in the simulation parameters or the data used in the study, as well as any assumptions or regulations regarding the current state of fog computing research, such as the assumption that fog computing is a relatively new field with ongoing development and refinement. In conclusion, this study provides valuable insights into the impact of trust keywords on network usage latency in fog computing and cloud computing. The findings can guide future research and development in fog computing to improve network performance and trustworthiness.

Funding: This research was funded by the Sensor Networks and Cellular Systems Research Center at the University of Tabuk (grant 154-1444 H).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data used to support the findings of this study are included within the article.

Acknowledgments: This study was supported by the Sensor Networks and Cellular Systems Research Center and the Deanship of Scientific Research, University of Tabuk.

Conflicts of Interest: The author has no conflict of interest to report regarding the present study.

References

1. Xia, Y.; Zhou, M.; Luo, X.; Zhu, Q.; Li, J.; Huang, Y. Stochastic modeling and quality evaluation of infrastructure-as-a-service clouds. *IEEE Trans. Autom. Sci. Eng.* **2015**, *12*, 162–170. [CrossRef]
2. Ghahramani, M.H.; Zhou, M.; Hon, C.T. Toward cloud computing QoS architecture: Analysis of cloud systems and cloud services. *IEEE/CAA J. Autom. Sin.* **2017**, *4*, 6–18. [CrossRef]
3. Ometov, A.; Molua, O.L.; Komarov, M.; Nurmi, J. A Survey of Security in Cloud, Edge, and Fog Computing. *Sensors* **2022**, *22*, 927. [CrossRef]
4. Wang, K.; Du, M.; Yang, D.; Zhu, C.; Shen, J.; Zhang, Y. Game-theory-based active defense for intrusion detection in cyber-physical embedded systems. *ACM Trans. Embed. Comput. Syst.* **2016**, *16*, 1–21. [CrossRef]
5. Shi, W.; Zhang, L.; Wu, C.; Li, Z.; Lau, F.C.M. An online auction framework for dynamic resource provisioning in cloud computing. *Perform. Eval. Rev.* **2014**, *42*, 71–83. [CrossRef]
6. Ma, F.; Luo, X.; Litvinov, E. Cloud Computing for Power System Simulations at ISO New England—Experiences and Challenges. *IEEE Trans. Smart Grid* **2016**, *7*, 2596–2603. [CrossRef]
7. Chen, X.; Jiao, L.; Li, W.; Fu, X. Efficient Multi-User Computation Offloading for Mobile-Edge Cloud Computing. *IEEE/ACM Trans. Netw.* **2016**, *24*, 2795–2808. [CrossRef]
8. Mahmud, R.; Kotagiri, R.; Buyya, R. Fog Computing: A taxonomy, survey and future directions. In *Internet of Everything: Algorithms, Methodologies, Technologies and Perspectives*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 103–130. [CrossRef]
9. Chen, S.; Irving, S.; Peng, L. Operational Cost Optimization for Cloud Computing Data Centers Using Renewable Energy. *IEEE Syst. J.* **2016**, *10*, 1447–1458. [CrossRef]
10. Zeng, D.; Gu, L.; Guo, S.; Cheng, Z.; Yu, S. Joint Optimization of Task Scheduling and Image Placement in Fog Computing Supported Software-Defined Embedded System. *IEEE Trans. Comput.* **2016**, *65*, 3702–3712. [CrossRef]
11. Bonomi, F.; Milito, R.; Zhu, J.; Addepalli, S. Fog computing and its role in the internet of things. In Proceedings of the MCC '12: Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing, Helsinki, Finland, 17 August 2012; pp. 13–15. [CrossRef]
12. Manzoor, A.; Shah, M.A.; Khattak, H.A.; Din, I.U.; Khan, M.K. Multi-tier authentication schemes for fog computing: Architecture, security perspective, and challenges. *Commun. Syst.* **2022**, *35*, e4033. [CrossRef]
13. Alam, S.; Shuaib, M.; Ahmad, S.; Jayakody, D.N.K.; Muthanna, A.; Bharany, S.; Elgendy, I.A. Blockchain-Based Solutions Supporting Reliable Healthcare for Fog Computing and Internet of Medical Things (IoMT) Integration. *Sustainability* **2022**, *14*, 15312. [CrossRef]

14. Stojmenovic, I.; Wen, S. The Fog computing paradigm: Scenarios and security issues. In Proceedings of the 2014 Federated Conference on Computer Science and Information Systems, Warsaw, Poland, 7–10 September 2014; Volume 2, pp. 1–8. [CrossRef]
15. Khan, S.; Parkinson, S.; Qin, Y. Fog computing security: A review of current applications and security solutions. *J. Cloud Comput.* **2017**, *6*, 19. [CrossRef]
16. Ali, A.; Ahmed, M.; Imran, M.; Khattak, H.A. Security and privacy issues in fog computing. In *Fog Computing: Theory and Practice*; Wiley: Hoboken, NJ, USA, 2020; pp. 105–137. [CrossRef]
17. Bessis, N.; Dobre, C. *Big Data and Internet of Things: A Roadmap for Smart Environments*; Springer: Berlin/Heidelberg, Germany, 2014; Volume 546, pp. 1–19. [CrossRef]
18. Tang, B.; Chen, Z.; Hefferman, G.; Pei, S.; Wei, T.; He, H.; Yang, Q. Incorporating Intelligence in Fog Computing for Big Data Analysis in Smart Cities. *IEEE Trans. Ind. Inform.* **2017**, *13*, 2140–2150. [CrossRef]
19. Molina, B.; Palau, C.E.; Fortino, G.; Guerrieri, A.; Savaglio, C. Empowering smart cities through interoperable sensor network enablers. In Proceedings of the 2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC), San Diego, CA, USA, 5–8 October 2014; pp. 7–12. [CrossRef]
20. Su, H.; Jung, C. Perceptual enhancement of low light images based on two-step noise suppression. *IEEE Access* **2018**, *6*, 7005–7018. [CrossRef]
21. Kang, J.; Yu, R.; Huang, X.; Zhang, Y. Privacy-Preserved pseudonym scheme for fog computing supported internet of vehicles. *IEEE Trans. Intell. Transp. Syst.* **2018**, *19*, 2627–2637. [CrossRef]
22. Wang, T.; Zhang, G.; Bhuiyan, M.Z.A.; Liu, A.; Jia, W.; Xie, M. A novel trust mechanism based on Fog Computing in Sensor–Cloud System. *Future Gener. Comput. Syst.* **2020**, *109*, 573–582. [CrossRef]
23. Fortino, G.; Guerrieri, A.; Russo, W.; Savaglio, C. Integration of agent-based and Cloud Computing for the smart objects-oriented IoT. In Proceedings of the 2014 IEEE 18th International Conference on Computer Supported Cooperative Work in Design (CSCWD), Hsinchu, Taiwan, 21–23 May 2014; pp. 493–498. [CrossRef]
24. Sarrab, M.; Alshohoumi, F. Assisted Fog Computing Approach for Data Privacy Preservation in IoT-Based Healthcare. In *Healthcare. Security and Privacy Preserving for IoT and 5G Networks: Techniques, Challenges, and New Directions*; Springer International Publishing: Berlin/Heidelberg, Germany, 2022; Volume 95. [CrossRef]
25. Moosavi, S.R.; Gia, T.N.; Nigussie, E.; Rahmani, A.M.; Virtanen, S.; Tenhunen, H.; Isoaho, J. End-to-end security scheme for mobility enabled healthcare Internet of Things. *Future Gener. Comput. Syst.* **2016**, *64*, 108–124. [CrossRef]
26. Al-Otaiby, N.; Alhindi, A.; Kurdi, H. AntTrust: An Ant-Inspired Trust Management System for Peer-to-Peer Networks. *Sensors* **2022**, *22*, 533. [CrossRef]
27. Liu, B. A Survey on Trust Modeling from a Bayesian Perspective. *Wirel. Pers. Commun.* **2020**, *112*, 1205–1227. [CrossRef]
28. Yan, Z.; Zhang, P.; Vasilakos, A.V. A survey on trust management for Internet of Things. *J. Netw. Comput. Appl.* **2014**, *42*, 120–134. [CrossRef]
29. Cho, J.H.; Swami, A.; Chen, I.R. A survey on trust management for mobile ad hoc networks. *IEEE Commun. Surv. Tutor.* **2011**, *13*, 562–583. [CrossRef]
30. Guo, J.; Chen, I.R.; Tsai, J.J.P. A survey of trust computation models for service management in internet of things systems. *Comput. Commun.* **2017**, *97*, 1–14. [CrossRef]
31. Okafor, K.C.; Achumba, I.E.; Chukwudebe, G.A.; Ononiwu, G.C. Leveraging Fog Computing for Scalable IoT Datacenter Using Spine-Leaf Network Topology. *J. Electr. Comput. Eng.* **2017**, *2017*, 2363240. [CrossRef]
32. Chen, I.R.; Guo, J.; Bao, F.; Cho, J.H. Integrated social and quality of service trust management of mobile groups in ad hoc networks. In Proceedings of the 2013 9th International Conference on Information, Communications & Signal Processing, Tainan, Taiwan, 10–13 December 2013. [CrossRef]
33. Ni, J.; Zhang, K.; Lin, X.; Shen, X.S. Securing Fog Computing for Internet of Things Applications: Challenges and Solutions. *IEEE Commun. Surv. Tutor.* **2018**, *20*, 601–628. [CrossRef]
34. Alzoubi, Y.I.; Osmanaj, V.H.; Jaradat, A.; Al-Ahmad, A. Fog computing security and privacy for the Internet of Thing applications: State-of-the-art. *Secur. Priv.* **2021**, *4*, e145. [CrossRef]
35. Marín-Tordera, E.; Masip-Bruin, X.; García-Almiñana, J.; Jukan, A.; Ren, G.J.; Zhu, J. Do we all really know what a fog node is? Current trends towards an open definition. *Comput. Commun.* **2017**, *109*, 117–130. [CrossRef]
36. Fang, W.; Zhang, W.; Chen, W.; Liu, Y.; Tang, C. TMSRS: Trust management-based secure routing scheme in industrial wireless sensor network with fog computing. *Wirel. Netw.* **2020**, *26*, 3169–3182. [CrossRef]
37. Rathee, G.; Sandhu, R.; Saini, H.; Sivaram, M.; Dhasarathan, V. A trust computed framework for IoT devices and fog computing environment. *Wirel. Netw.* **2020**, *26*, 2339–2351. [CrossRef]
38. Afzali, M.; Pourmohammadi, H.; Samani, A.M.V. An efficient framework for trust evaluation of secure service selection in fog computing based on QoS, reputation, and social criteria. *Computing* **2022**, *104*, 1643–1675. [CrossRef]
39. Chen, D.; Chang, G.; Sun, D.; Li, J.; Jia, J.; Wang, X. TRM-IoT: A trust management model based on fuzzy reputation for internet of things. *Comput. Sci. Inf. Syst.* **2011**, *8*, 1207–1228. [CrossRef]
40. Chen, I.R.; Guo, J.; Bao, F. Trust management for service composition in SOA-based IoT systems. In Proceedings of the 2014 IEEE Wireless Communications and Networking Conference (WCNC), Istanbul, Turkey, 6–9 April 2014; pp. 3444–3449. [CrossRef]
41. Bao, F.; Chen, I.-R. Dynamic Trust Management for the Internet of Things Applications. In Proceedings of the 2012 International Workshop on Self-Aware Internet of Things, New York, NY, USA, 17 September 2012; pp. 1–30.

42. Jayasinghe, U.; Truong, N.B.; Lee, G.M. RpR: A Trust Computation Model for Social Internet of Things. In Proceedings of the2016 Intl IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld), Toulouse, France, 18–21 July 2016. [CrossRef]
43. Hajizadeh, R.; Navimipour, N.J. A method for trust evaluation in the cloud environments using a behavior graph and services grouping. *Kybernetes* **2017**, *46*, 1245–1261. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.