technologies

MDPI

Article

FogTrust: Fog-Integrated Multi-Leveled Trust Management Mechanism for Internet of Things

Abdul Rehman [1], Kamran Ahmad Awan [1], Ikram Ud Din [1,*], Ahmad Almogren [2,*] and Mohammed Alabdulkareem [2]

[1] Department of Information Technology, The University of Haripur, Haripur 22620, Pakistan
[2] Chair of Cyber Security, Department of Computer Science, College of Computer and Information Sciences, King Saud University, Riyadh 11633, Saudi Arabiaa
* Correspondence: ikramuddin205@yahoo.com (I.U.D.); ahalmogren@ksu.edu.sa (A.A.)

Abstract: The Internet of Things (IoT) is widely used to reduce human dependence. It is a network of interconnected smart devices with internet connectivity that can send and receive data. However, the rapid growth of IoT devices has raised security and privacy concerns, with the identification and removal of compromised and malicious nodes being a major challenge. To overcome this, a lightweight trust management mechanism called FogTrust is proposed. It has a multi-layer architecture that includes edge nodes, a trusted agent, and a fog layer. The trust agent acts as an intermediary authority, communicating with both IoT nodes and the fog layer for computation. This reduces the burden on nodes and ensures a trustworthy environment. The trust agent calculates the trust degree and transmits it to the fog layer, which uses encryption to maintain integrity. The encrypted value is shared with the trust agent for aggregation to improve the trust degree's accuracy. The performance of the FogTrust approach was evaluated against various potential attacks, including On-off, Good-mouthing, and Bad-mouthing. The simulation results demonstrate that it effectively assigns low trust degrees to malicious nodes in different scenarios, even with varying percentages of malicious nodes in the network.

Keywords: Internet of Thing; fog-computing; trust management; security; privacy preservation; trustworthiness

check for updates

Citation: Rehman, A.; Awan, K.A.; Ud Din, I.; Almogren, A.; Alabdulkareem, M. FogTrust: Fog-Integrated Multi-Leveled Trust Management Mechanism for Internet of Things. *Technologies* **2023**, *11*, 27. https://doi.org/10.3390/technologies11010027

Academic Editors: Manoj Gupta, Eugene Wong and Gwanggil Jeon

Received: 21 December 2022
Revised: 30 January 2023
Accepted: 2 February 2023
Published: 7 February 2023

1. Introduction

The concept of the Internet of Things (IoT) [1] has become more prevalent, though the idea of connected devices dates back to the 1970s. The term "Internet of Things" was introduced in 1999 by Kevin Ashton [2]. IoT is a technology that connects devices and machines to communicate with each other [3]. It is used in various fields, such as smart homes [4], wearable technology [5], and smart agriculture [6]. Despite its wide range of applications, IoT faces several challenges, including security, connectivity, privacy, interoperability, and energy consumption [7–9]. In 2018, over 23 billion devices were connected, which is twice the population [10]. The future projection is that the number of IoT devices will increase to a minimum of 80 billion [11]. The main goal of IoT is to make all devices autonomous through the power of the internet. However, the vast number of devices presents major privacy and security challenges [12]. These are critical issues that IoT companies must address for a promising future. The major security challenges include authentication [13], access control [14], policy enforcement [15], mobile security [16], secure middleware, confidentiality, and latency [8]. Security is a crucial concern for organizations, governments, and individuals, as they become increasingly digital-centric [17]. With the growing complexity of IoT attacks, it is important to detect, defend against, and respond to these threats. Hackers now have additional access points that can affect the real world [18].

Fog computing promotes IoT innovation through an open architecture [19]. It is a decentralized form of computing, where applications and data storage are located between the data source and cloud [20]. Fog computing performs computation, storage, and communication from edge devices, which control the flow of data between two networks such as routers, switches, access devices, gateways, hubs, etc. Fog operates in a DIST network environment [21] that is closely connected to the cloud and IoT/edge devices. It processes selected data locally before sending it to the cloud, reducing bandwidth [22] and latency [23] needs. An important benefit of fog computing is improved security, as it provides computing security locally rather than remotely.

In this article, a mechanism named FogTrust is proposed to detect and eliminate compromised and malicious nodes. The proposed system uses the fog to provide data integrity, which helps to prevent potential IoT attacks such as on-off, good-mouthing, and bad-mouthing attacks. To ensure security and integrity, a lightweight mechanism is implemented to maintain the integrity and aggregate the computed trustworthiness data (TD) for aggregation purposes. The TD of IoT nodes will reduce the impact of malicious and compromised nodes in good and bad-mouthing attacks. The use of a trust agent as an intermediary between the fog and IoT nodes performs a trust evaluation, reducing the computational burden on less capable nodes to improve security and reduce vulnerabilities caused by such nodes. The proposed approach can be summarized as:

1. The proposed mechanism, FogTrust, uses a multi-layer trust management (TM) architecture with central authorities to maintain a secure environment by detecting and eliminating malicious and compromised nodes with low trust.
2. The fog is integrated into the architecture to encrypt and maintain the integrity of the trust degree (TD) computed by trust agents.
3. The proposed system aggregates the current trust (CT) with previous trust (PT) to form the aggregated TD of a node, providing robustness against potential IoT attacks.

The structure of the rest of the paper is as follows:

Section 2 summarizes the existing literature and provides a comparative analysis to highlight the limitations. Section 3 explains the working of the PM, including the proposed architecture, trust parameters and computations, direct trust computation, indirect trust computation, trust development, and decision-making. Section 4 presents simulation results and discusses the performance of FogTrust compared to existing literature. Section 5 concludes the paper.

2. Related Work

To manage the IoT trust, various TM mechanisms have been proposed, including DIST, and CENT. DIST relies on nodes to manage trust between nodes, while IoT nodes in CENT depend on a CA for trust management. Despite several techniques for addressing trust management, the privacy and security challenges in fog computing remain significant, as sensitive information is transmitted between IoT devices or the edge layer and fog layer. Identifying malicious nodes and protecting data from attacks in fog computing is a major issue, but no notable solution has been proposed to address these security challenges in data sharing in fog computing.

A novel context-based trust management model is proposed for the Social IoT [24]. The proposed "ConTrust" approach uses a novel combination of parameters (satisfaction, commitment, and capability) to increase system efficiency. ConTrust measures job characteristics, honesty, job capability, and behavior of malicious nodes. Its architecture includes three components: job requester, trust management, and prospective provider. When a job requester requests a service, the trust evaluation process starts by computing trust parameters. If a node is trustworthy, the prospective provider will provide services to the requester. ConTrust is limited to covering IoT-related potential attacks. A multi-dimensional trust management model based on SLA is also proposed for Fog computing [25]. It contributes to applications, peers, and fog editors for fog service providers and measures their trustworthiness. The architecture comprises five components: smart application client, fog auditors, SLA agent, service providers, and smart applications. The model works when a service provider advertises their services and the application interacts with them for the first time.

In [26], a lightweight trust mechanism is presented that uses trust agents to manage communication certificates, which identify the trustworthiness of nodes using parameters. The mechanism employs a statistical probabilistic model to compute the degree of trust with high precision and adaptability. Its main role is to provide a solid and reliable mechanism for edge device information exchange [27]. The system can be enhanced with a hybrid approach to detect malicious nodes and network attacks. Ref. [28] presents research focusing on privacy and security in fog computing with IoT applications. It uses a subjective logic-based (SL-B) trust approach to improve IoT security and address challenges related to data transmission protection and protection against compromised attacks. The proposed system maintains the trustworthiness of each node in the network, calculates and updates their trust values, and stores them in a local list with a node ID.

In the field of cloud computing, industry TM is a recurring research trend [29]. It is expected that similar problems will arise in the emerging fog realm. Although the fog and cloud are similar, evaluating the trust in fog is more challenging than evaluating the trust in the cloud due to its mobility, distributed nature, and proximity to the end-user [18]. Unlike clouds, fog has little to no human involvement and is not redundant, meaning that disruptions may occur at any time, making it difficult to trust. These unique characteristics can be used as metrics to assess fog trust, along with existing features. In [30], a fuzzy logic approach was proposed to evaluate trust in fog and identify configurations that can alter its trust value. A campus scenario was presented as an example application, where various fog resources (FRs) were evaluated for reliability using the proposed metrics. The scenario discussed the FRs and attributes used to assess their trustworthiness, and an adopted fuzzy-logic approach was used to handle the complex trust values. The approach follows the steps of a fuzzy inference system, first evaluating the attributes of distance, latency, and reliability, then using the AND operator rather than the OR operator in the second step.

In [31], a TM framework is proposed that uses the MAPE-K feedback control loop to evaluate the trust levels. The framework includes trust agents and a consumer layer of TMS nodes that interact with clients. The cloud filters trust parameters into an adaptive trust parameters pool and assists with trust evaluation via the MAPE-K loop. The input framework takes into account the previous history to standardize the effect of anomalies. False decisions caused by malicious information decrease the effectiveness of the MAPE-K loop.

In [32], a TM scheme called COMITMENT is presented for fog computing. It uses the fog node reputation to construct a global reputation language and provides secure and trusted environments for information exchange. The DIST fog topology is considered, with nodes connected through communication protocols and a unique identity. Each node computes the trust evaluation of its nearest nodes to create a list of trusted nodes. The COMITMENT is a set of protocols installed on fog nodes that select trusted nodes for information sharing and provide a secure environment for resource sharing and information exchange. The goal is to build trust between parties to facilitate sensitive information exchange. The approach requires a central trust authority for trust level evaluation.

In [33], a two-way trust management system (TMS) is presented that allows both the service requester (SR) and service provider (SP) to evaluate each other's trustworthiness. The TMS aggregates trust using subjective logic theory, which is useful when uncertainty and proposition are involved. Clients request services from fog servers and the fog server evaluates the trustworthiness of the clients through direct observation and consultation with the nearest fog server. The clients also consult the nearest server to determine the trust level of the fog server. Both clients and servers in the fog share information about other clients and servers. The system must simultaneously calculate the trustworthiness of both the SR and SP. In a recent study, Trust2Vec [34], a trust management system for large-scale IoT systems, is proposed. The system has the ability to manage trust relationships in large IoT systems and mitigate attacks from malicious devices. It uses a network structure to build trust relationships among devices and has a key phase to detect malicious nodes through determining device communities, generating random walk algorithms, and leveraging trust relationships in clusters. The proposed system has an overall detection rate of 94% for malicious devices or nodes, and its key contribution is the use of a random-walk algorithm for navigating trust relationships and a parallelization method for attack detection.

In [35], a TM model is proposed to improve security, social relationships, and services in fog computing. The model evaluates trust through direct trust, recommendations, and reputation, and uses fuzzy logic to aggregate trust and handle uncertainty in mobile fog computing. The detection and mitigation rate is approximately 71%, with 70% of clients and fogs being malicious and 74% of attacks detected. However, like other TMS in fog computing, it assumes fog nodes are static, making it challenging to handle dynamic nodes. The contributions and limitations of the existing approaches are provided in Table 1.

Table 1. Comparative Analysis of Existing Literature.

Ref.	Contribution	Limitation
[24]	A trust management model is presented in social IoT that is context-dependent to compute the trust.	Need to check the proposed system against potential attacks that are related to trust.
[25]	A multi-dimensional trust management system is presented to check the trustworthiness of FSP.	Need to evaluate the malicious behavior of applications that enter in fog environment.
[26]	Utilizes a lightweight mechanism that manages trust in IIoT-Edge nodes.	Requires improved prediction capabilities to increase performance.
[28]	Establishes a secure environment for fog applications by using a TM.	A hybrid technique is required to ensure robust network security.
[30]	Utilizes a fuzzy approach to evaluate trust in fog computing.	Requires a broker that acts as a fog TM.
[31]	Utilizes a MAPE-K feedback control loop for evaluation of trust level.	Requires trust to be calculated before the fog layer and data to be protected in the fog layer.
[32]	Utilizes the COMITMENT approach for security in fog computing.	Requires a CA that evaluates trust before the fog layer.
[33]	TW-TMS evaluates the trust level of SP and then checks the TD of SD.	Requires the trustworthiness of the SP and SR to be calculated at the same time.
[34]	Utilizes a random-walk algorithm for the navigation of trust relationships and parallelization method for attack detection.	The work can be extended by including the TM of data entities.
[35]	Utilizes fuzzy logic for trust aggregation to handle uncertainty in fog computing.	Static nodes handling is difficult.

3. Proposed FogTrust Mechanism

The proposed model will use the fog computing to ensure data integrity, which will reduce the possibility of various IoT attacks, including on-off attacks, good-mouthing attacks, and bad-mouthing attacks. The system proposes a lightweight encryption method to protect the TD and aggregate its evaluation to maintain integrity. The encrypted TD from IoT nodes reduces the impact of malicious and compromised nodes in good and bad-mouthing attacks. A trust agent, acting as an intermediary between the fog and IoT nodes, performs trust evaluation, reducing the computational burden on less capable nodes, thereby improving security and reducing vulnerabilities posed by such nodes.

3.1. Proposed Architecture of FogTrust

The proposed architecture of FogTrust consists of three layers: community layer, trust agent layer, and fog layer. The working of the proposed architecture is shown in Figure 1; the FogTrust includes communities separated into different domains, each of which has nodes that can connect with one another to complete specific tasks. The IoT nodes have a unique identity, and the message file includes their identification, community, and domain information. When a node (TE) requests communication from another node (TR), TR provides material to the trust agents for trust evaluation. The community layer, edge layer, or IoT node layer consists of IoT devices such as smart cameras, smartwatches, smartphones, smart laptops, sensors, and other IoT-related devices that can generate and transmit data or information autonomously. Before the data are transmitted to the fog, the trust agent in the trust agent layer evaluates its trustworthiness, determining whether the information is trustworthy or not.

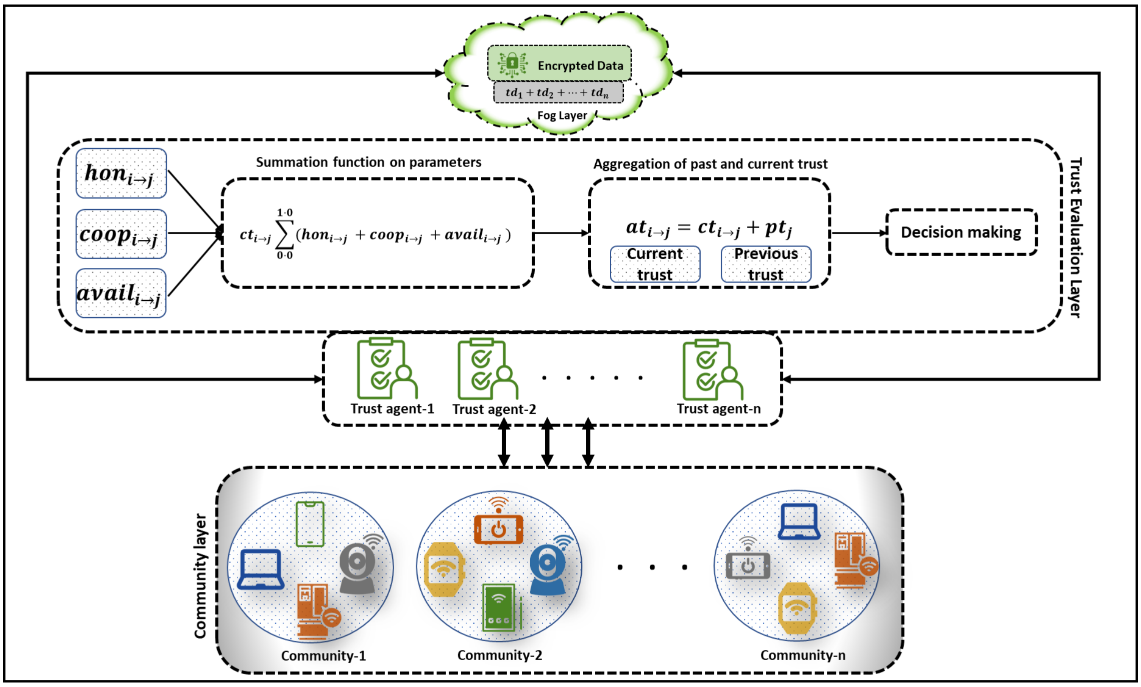

Figure 1. The proposed FogTrust Architecture.

The proposed system in FogTrust performs trust evaluation using a combination of three trust parameters: honesty, cooperativeness, and availability. The evaluation combines current trust and previous trust values to compute the aggregated trust values, which are used to make trust decisions. The trustworthiness of a device is determined by comparing its trust data (TD) with a threshold value that ranges from 0.0 to 1.0, with 0.0 being the minimum trust and 1.0 being the maximum trust. Newly joined nodes are assigned a default trust value of 0.5.

3.2. Trust Parameters and Computation

The trust evaluation combines three parameters: availability, honesty, and cooperativeness to enhance the reliability and security in the IoT network. Availability refers to the accessibility of resources to end-users, while cooperativeness reflects a node's ability to collaborate with others. Honesty is determined based on the observations of one node (i) towards another node (j). The cooperativeness is measured by analyzing response time and calculated as the ratio of prompt responses to the total number of responses. The evaluation considers the previous and current trust values to make the final trust decision. The threshold for trust ranges from 0.0 to 1.0, with 0.0 being the minimum and 1.0 being the maximum trust. New nodes are assigned a default trust value of 0.5.

3.3. Direct Trust Computations

The evaluation of the direct trust procedure begins with the *TE* being identified using their unique ID. The Algorithm 1 represents a direct trust observation procedure that takes place when a TR needs to evaluate the *TD*. TE requests services from TR during the joining of the network.

Algorithm 1 DOB-Trust Computation

```
 1: procedure TRUST EVALUATION($i \to j$)
 2:     $j_{i_d}$                                                   ▷ Identification of TE
 3:     $j_{req} \to i$                                             ▷ Request TE towards TR
 4:     $pt_{o_b}: I \to J\ hon_{i\to j}, coop_{i\to j}, avail_{i\to j}$
 5:     if ($pt_{ob}: i \to j == Yes$) then
 6:         $GotoStep - 9;$
 7:     else
 8:         $GotoAlgorithm - 2;$
 9:     $\mathcal{E}va_{Trust}: i \to j[hon_{i\to j}, coop_{i\to j}, avail_{i\to j}]$
10:     $\sum_{0.0}^{1.0} ct_{i\to j}^{direct} = \sum(hon_{i\to j} + coop_{i\to j} + avail_{i\to j})$
11:     $at_{i\to j} = ct_{i\to j} + pt_j$                          ▷ Aggregated Trust Formulation
12:     if ($at_{i\to j} \geq threshold$) then
13:         $ProvideServices;$
14:     else
15:         $Decline;$
16:     Exit.
```

The j_{id} shows the identification of the TE that requested to gather the services from the *TR*. Where *j* represents the TE and *id* is the identification, it is initialized when a *TR* receives a request from the *TE* for services. In j_{req}, *j* represents the *TE*, *req* is the request, and *i* demonstrates *TR*. This is where the *TE* requests services from the *TR*.

$$pt_{ob}: I \to J[hon_{i\to j}, coop_{i\to j}, avail_{i\to j}] \tag{1}$$

The evaluation process starts with determining the trust level of the TE node using the trust parameters of honesty, cooperativeness, and availability, as described in Equation (1). The TE is identified and initialized into the network. In Equation (1), *pt* represents past trust, *ob* represents observation, *I* and *J* represent TR and TE, respectively, *hon* represents honesty, *coop* represents cooperativeness, and *avail* represents availability.

$$If(pt_{ob}: \ \to j == Yes) \tag{2}$$

The Equation (2) represents the observations that the TE (*j*) must gather for the TR (*i*) before service can be provided. TR (*i*) will evaluate TE (*j*) only if the required observations are equal to "yes".

$$\mathcal{E}va_{Trust}: i \to j[hon_{i\to j}, coop_{i\to j}, avail_{i\to j}] \tag{3}$$

In Equation (3), the evaluation process of TE *j* through TR *i*. Here, *εva* represents the trust evaluation.

$$\sum_{0.0}^{1.0} ct_{i\to j}^{direct} = \sum(hon_{i\to j} + coop_{i\to j} + avail_{i\to j}) \tag{4}$$

In Equation (4), the current trust (*ct*) is evaluated by combining the direct trust evaluation (*direct*) between the TR (*i*) and TE (*j*).

$$at_{i\to j} = ct_{i\to j} + pt_j \tag{5}$$

In Equation (5), *at* represents the aggregated trust which is calculated as the mean of the current trust (*ct*) and previous trust (*pt*) of node *j* and *i* and *j*, which are, respectively, the TE and the TR. The $i \to j$ symbol represents the trust of TE towards TR. The final TD is formulated by aggregating the past and current trust values.

$$If(at_{i\to j} \geq threshold) \tag{6}$$

In Equation (6), *at* represents the aggregated trust, and *threshold* is the predetermined threshold value, which is compared to the aggregated trust value. If the aggregated trust value is greater than or equal to the threshold value, then the TR (*i*) starts providing services to the TE (*j*) and communication starts.

3.4. Absolute TD Formulation

The evaluation process of the absolute TD formulation starts with identifying the TE using its unique ID. The algorithm referred to as Algorithm 2 represents the procedure of absolute observations that take place when a TR needs to evaluate the degree of trust. The TE requests services from the TR after joining the network, and the Algorithm 2 thoroughly explains the TD formulation procedure.

Algorithm 2 ATD-Formulation

```
1:  procedure TRUST EVALUATION($i \rightarrow j$)
2:      $j_{id}$                                                    ▷ Identification of TE
3:      if (j==new) then                                            ▷ Newly joined node check
4:          $GotoStep-7;$
5:      else
6:          $GotoAlgorithm-3;$
7:      $\mathcal{E}_{Trust}: i \rightarrow j[hon_{i \rightarrow j}, coop_{i \rightarrow j}, avail_{i \rightarrow j}]$
8:      $\sum_{0.0}^{1.0} ct_{i \rightarrow j}^{form} = \sum(hon_{i \rightarrow j} + coop_{i \rightarrow j} + avail_{i \rightarrow j})$      ▷ Direct Trust formulation
9:      $at_{i \rightarrow j} = ct_{i \rightarrow j}^{form} + pt_j$                     ▷ Aggregated Trust Formulation
10:     if ($at_{i \rightarrow j} \geq threshold$) then
11:         $ProvideServices;$
12:     else
13:         $Decline;$
14:     Exit.
```

$$If(j == new) \tag{7}$$

The Equation (7) is used to determine whether the TE is new to the network or not. If the TE *j* is determined to be new, then its trust is evaluated. Otherwise, trust is measured using the Algorithm 3.

$$\mathcal{E}_{Trust}: i \rightarrow j[hon_{i \rightarrow j}, coop_{i \rightarrow j}, avail_{i \rightarrow j}] \tag{8}$$

In Equation (8), the process of the trust value evaluation is illustrated. The variable → represents the evaluation, and *hon*, *coop*, and *avail* represent the honesty, cooperativeness, and availability of the TE, respectively.

$$\sum_{0.0}^{1.0} ct_{i \rightarrow j}^{form} = \sum(hon_{i \rightarrow j} + coop_{i \rightarrow j} + avail_{i \rightarrow j}) \tag{9}$$

In Equation (9), *ct* represents the current trust, while *i* and *j* are the TE and the TR, respectively.

$$at_{i \rightarrow j} = ct_{i \rightarrow j}^{form} + pt_j \tag{10}$$

In Equation (10), *at* represents the aggregated trust which is formulated as the mean of *ct* current trust, where *form* is the formulation of trust and *pt* is the previous trust of node *i* and *j*, respectively.

After formulation of the Algorithm 2, it will further evaluate the honesty, cooperativeness, and availability as described in Algorithm 1 and as elaborated earlier in Equations (1) and (3). The algorithm then formulates the direct overall degree of trust by aggregating the current and previous trust evaluations and checking the final aggregated TD against the threshold value to determine if it can provide the services. The function and description of these Equations (4) and (10) have been explained earlier.

3.5. Recommendations-Based Indirect Trust Evaluation

When direct observation of the TE is not available, the TR must rely on recommendations to evaluate the trust level. The indirect trust evaluation will be conducted by gathering recommendations from nearby nodes based on their knowledge of the TE. If the available observations are insufficient, these algorithms pass a request to Algorithm 3 to compute the indirect trust. If the information shows that the TE is not from the same network, Algorithm 3 will perform an indirect evaluation.

Algorithm 3 RB-Indirect Trust Evaluation

```
procedure TRUST EVALUATION($i \to j$)
    Generating Request to gather Recommendations $\to r_j \to k_{th}$
    $j_{id}$                                                      ▷ TE Identification
    $rec^{check}[i \to j]$
    $rec^{eva}_{j\to k_{th}}$ : $[r_{k_1\to j} + r_{k_2\to j} + \ldots\ldots\ldots + r_{k_n\to j}]$
    $\sum_{0.0}^{1.0} r^{re}_{j\to k_{th}} = r^{je_1}_{j\to k_{th}} + r^{je_2}_{j\to k_{th}} + \ldots + r^{je_n}_{j\to k_{th}}$
    $\sum_{0.0}^{1.0} r^{indirect}_{j\to t} = \sum_{r=0.0}^{r=1.0} r^{re}_{j\to k_{th}}$
    $pt_j = r^{indirect}_{j\to t}$
    $at_{i\to j} = ct_i \to j + pt_j$
    if ($at_{i\to j} > Yes$) then
        $ProvideServices;$
    else
        $Decline;$
    Exit.
```

The Algorithm 3 begins by sending requests for recommendations to nearby nodes. Equation (11) represents the generation of these requests to gather the necessary information for evaluating the TD of a TE.

$$GeneratingRequesttogatherRecommendations \to r_j \to k_{th} \tag{11}$$

In Equation (11), k_{th} represents the nearest nodes (k) and th represents the number of nodes to which a system sends requests for recommendations for a TE evaluation.

$$\sum_{0.0}^{1.0} r^{re}_{j\to k_{th}} = r^{je_1}_{j\to k_{th}} + r^{je_2}_{j\to k_{th}} + \ldots + r^{je_n}_{j\to k_{th}} \tag{12}$$

After gathering the recommendations, they are arranged correctly. In Equation (12), r represents the recommendations, re represents the number of received recommendations, k represents the neighboring node, and th represents the number of generated requests.

$$\sum_{0.0}^{1.0} r^{indirect}_{j\to t} = \sum_{r=0.0}^{r=1.0} r^{re}_{j\to k_{th}} \tag{13}$$

The algorithm evaluates trust by calculating the total degree of trust after gathering the recommendations. The mean value of the recommendations is used to compute the overall degree of trust, which results in a final degree of trust with a value between 0.0 and 1.0.

$$pt_j = r^{indirect}_{j\to t} \tag{14}$$

In Equation (14), the algorithm calculates the indirect trust value by aggregating it with the previous trust (PT) value. pt represents previous trust, r represents the recommendation, *indirect* represents the indirect trust evaluation, and $j \rightarrow t$ indicates that the j TE generates a request to gather recommendations from $k_t h$ nodes.

The rest of the Algorithm 3 operates similarly to what was described earlier. It combines the current trust value with the previous one and compares the aggregated trust value to the threshold value, as outlined in Algorithm 1. If the TE's trust value surpasses the threshold value, the TR offers services. Otherwise, the TR declines and ceases further communication.

3.6. Trust Development

Trust agents can calculate the whole trust value through trust development. They evaluate three separate parameters and use the standard function sigma to obtain the aggregated trust value from the trust parameters' output. The final TD is then formulated and shared with the fog layer. Nodes with low TD are not allowed to share information or communicate. However, nodes with supreme trust or TD higher than the threshold are allowed to communicate further. To determine a node's trustworthiness, the trust evaluation layer computes its trust value and compares it to a predefined threshold. Trust agents can evaluate the aggregated trust value, allowing trust development. They calculate three different parameters and use the sigma function to obtain the aggregated trust value from the trust parameters output.

3.7. Decision Making

The IoT network uses the absolute trust value to make quick decisions for improving system efficiency. The TD of nodes is calculated by evaluating parameters with a comparison to a threshold, with a range of 0.0 to 1.0 and a default trust level of 0.5. A trust value of 0.0 to 0.49 is untrustworthy, 0.51 to 0.79 is moderately trustworthy, and 0.8 to 1.0 is supremely trustworthy. Nodes with trust values above 0.5 are allowed to communicate in the network. The TM must have an effective and reliable technique for determining the absolute trust value.

4. Experimental Simulation and Outcomes

This section presents the simulation results of FogTrust with the existing TM mechanism. The authors have evaluated the trustworthiness of the system in terms of good and bad-mouthing attacks, as well as various on-off attack scenarios. They also compare their proposed approach to existing TM mechanisms such as ConTrust and SLA-Trust. The criteria used for evaluating their work include Aggregation Impact, Good and Bad-mouthing attacks, and On-Off attacks. The simulation results were generated using MATLAB, a multi-paradigm programming language and computing development framework developed by MathWorks. MATLAB is mainly used for matrix operations, data visualization, algorithm implementation, user interface creation, and interfacing with other programming languages. Although symbolic computation is not a primary function of MATLAB, it can be performed through an optional toolbox that uses the MuPAD symbolic engine. Additionally, the Simulink tool provides visual simulation capabilities for dynamic and integrated systems. The data used in the simulation analysis is experimental and is generated when an IoT node joins the network. The proposed approach assigns a pre-defined default trust degree to each node, allowing for communication between nodes.

The simulation setup for the proposed FogTrust mechanism is shown in Table 2. The simulation uses data from the table, which includes the "area" parameter set at 200 square meters and "number of devices" set at 600 randomly distributed. The simulation runs for 100 s, with a data transmission rate of 6 to 8 Mbps. The malicious node detection rate during the simulation is between 50% and 75%.

Table 2. Simulation Environment Implementation Setup.

Parameters	Value
Network area	200 m^2
Number of devices	600
Simulation duration	100 (s)
Degree of trust	0.0∼1.0
Default trust	0.5
Node distribution	Random
Transmission rate	6∼8 Mbps
Malicious nodes percentage	50∼75%

4.1. Analysis of the Trust Aggregation

This section presents the impact of using the aggregation process on the trust degree computation. The comparison is made between using the previous trust with the present computed trust degree and the computation performed without the aggregation process. The use of the aggregation process has a significant impact on the trust degree computation, resulting in more consistent values, as shown in Figure 2.

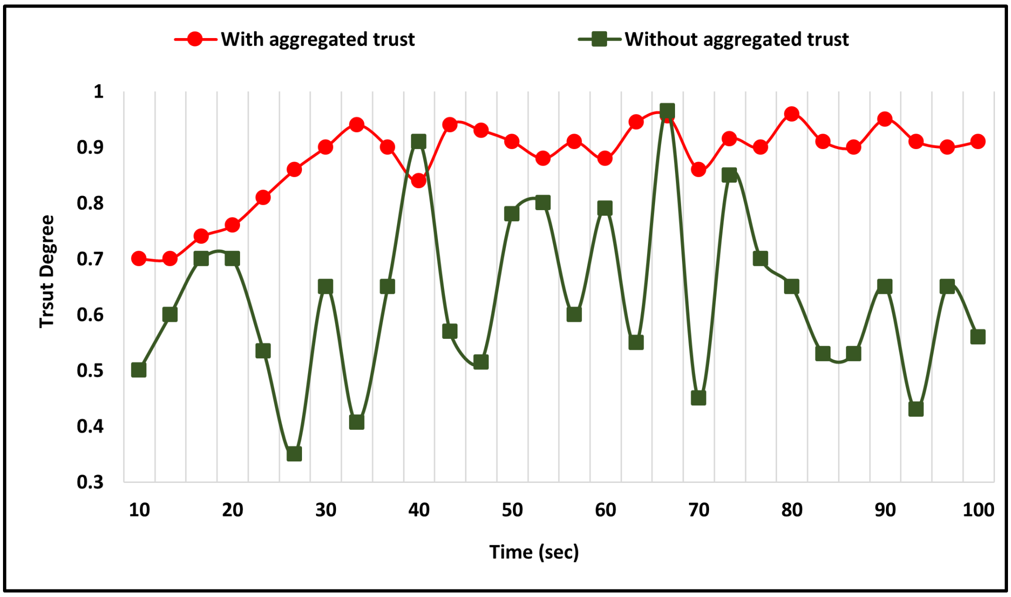

Figure 2. Previous Trust Aggregation Impact on Direct Trust Evaluation.

The comparison shows that the use of aggregation in the trust calculation process results in more consistent trust values and improved reliability compared to the scenario where aggregation is not used. This highlights the importance of considering past trust data in determining the current trust level, which helps to reduce errors and improve the security of the network by accurately identifying malicious nodes.

4.2. Analysis of Detection Rate

The detection rate is a crucial metric for evaluating the performance of any trust management system, as it reflects the system's ability to accurately identify trustworthy entities. Our proposed approach in this article enhances the detection rate by aggregating previous trust degrees with the current computed trust, resulting in more accurate and reliable trust decisions. In this simulation setup, each node has several close neighbors that offer various services over time, while the percentage of malicious and compromised nodes is 70%.

Figure 3 presents the simulation results of the proposed approach in terms of the number of interactions and detection percentage. The results demonstrate that the proposed mechanism has an initial detection rate of 70% and steadily increases over time, reaching over 80% after 25 interactions and exceeding 90% after 45 interactions. This indicates that the proposed approach outperforms other existing mechanisms, such as SLA-Trust, which has a continuous improvement in detection rate, reaching a peak of 81%. While ConTrust has a higher initial detection rate of 80%, it decreases to 66.5% after 20 interactions. Its highest detection rate is 89%, but is still lower than that of FogTrust. The average detection rate of FogTrust is 84.32%, which is higher than that of SLA-Trust (67.89%) and ConTrust (79.66%).

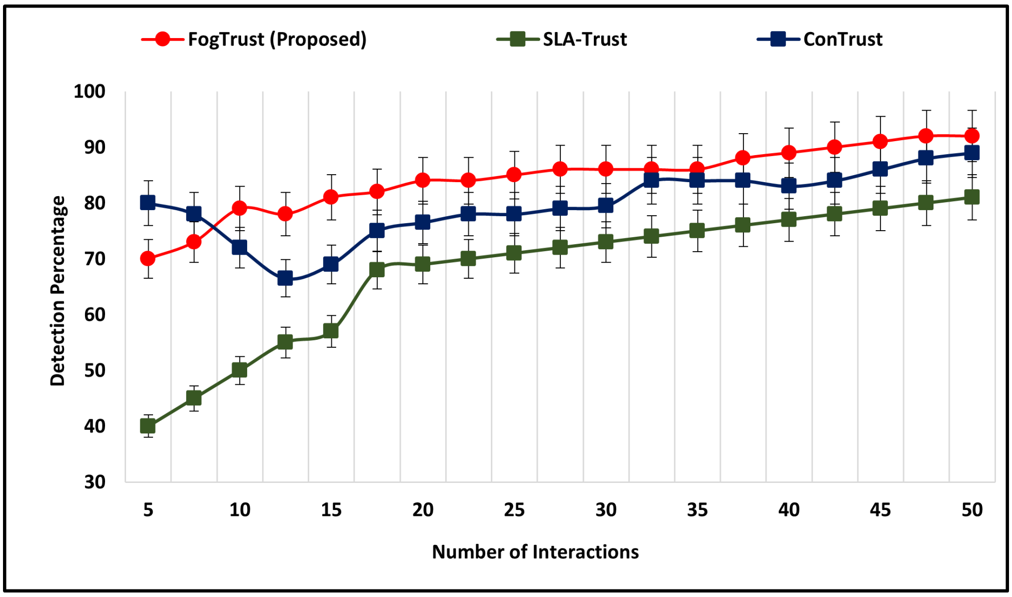

Figure 3. The Detection Rate Comparison of FogTrust with Exiting Approaches.

4.3. On-Off Attack

The simulation results demonstrate that in the occurrence of an on-off attack, the TD of compromised nodes decreases dramatically from 0.5 to 0.2 within seconds. This highlights the effectiveness of the proposed mechanism in detecting and mitigating the impact of such attacks. However, it should be noted that in the case of ConTrust [24], the malicious node may regain its trust after a certain period, which suggests the need for continuous monitoring and updating of trust values.

Figure 4 shows the malicious nodes' level of trust, which decreases and is still unable to regain the highest trust level. In comparison to SLA-Trust [25], the proposed mechanism successfully detects the on-off attack and the malicious node's degree of trust. Furthermore, in the case of ConTrust, the trust value of the malicious node goes down. The malicious node regains its trust to 0.35 in 70 s, but after 70 s it again increases. Similarly, the SLA-Trust value of the malicious node also decreases, which shows that FogTrust can detect the malicious node at a low level of trust.

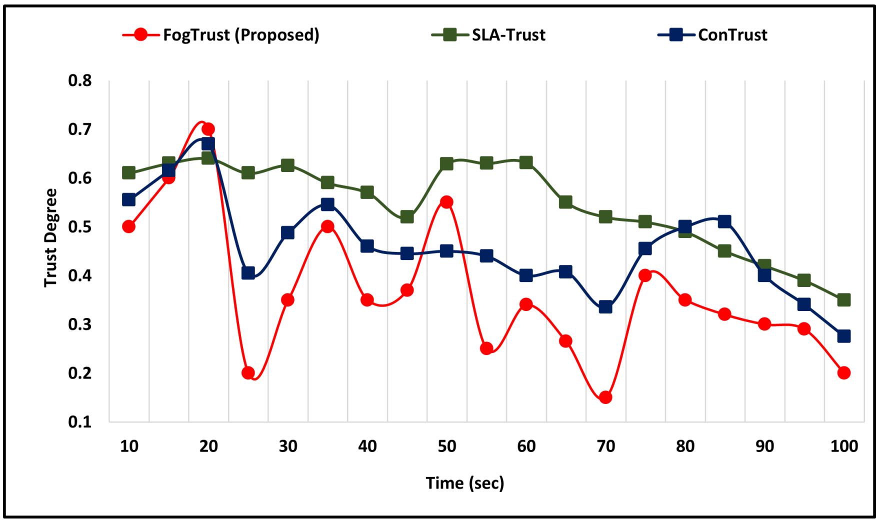

Figure 4. Comparative Analysis of FogTrust Against On-off Attacks.

4.4. Good and Bad Mouthing Attack

This section discusses the comparative simulation outcomes against good and bad-mouthing attacks. The trust value is a predefined threshold value ranging from 0.0 to 1.0. Time (s) is 100 and trust defaults to 0.5. To test the efficiency of the proposed approach against attacks involving goodmouth, we put three trust management models into practice. When the number of negative recommendations grows over time, the level of trust is shown to be declining as shown in Figure 5.

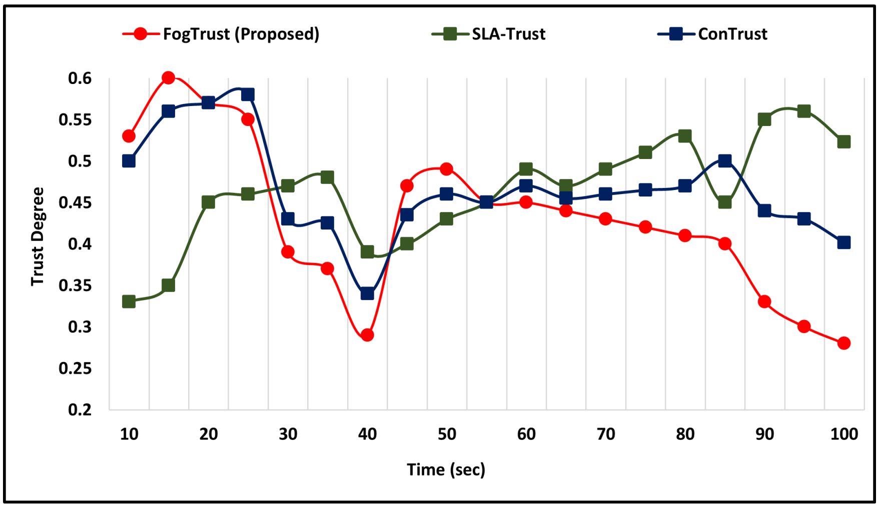

Figure 5. Comparative Analysis Against Good-Mouthing Attacks.

The effectiveness of the PM against bad-mouthing attacks has also been evaluated. The results indicate that the PM is effective in preventing such attacks. Three models were implemented, each with different trust and threshold values. As depicted in Figure 6, as the trust value increases, the detection rate also increases, but if the trust value increases too much, then the detection rate decreases. This indicates that the PM can detect bad-mouthing attacks even when they are at an increasing rate.

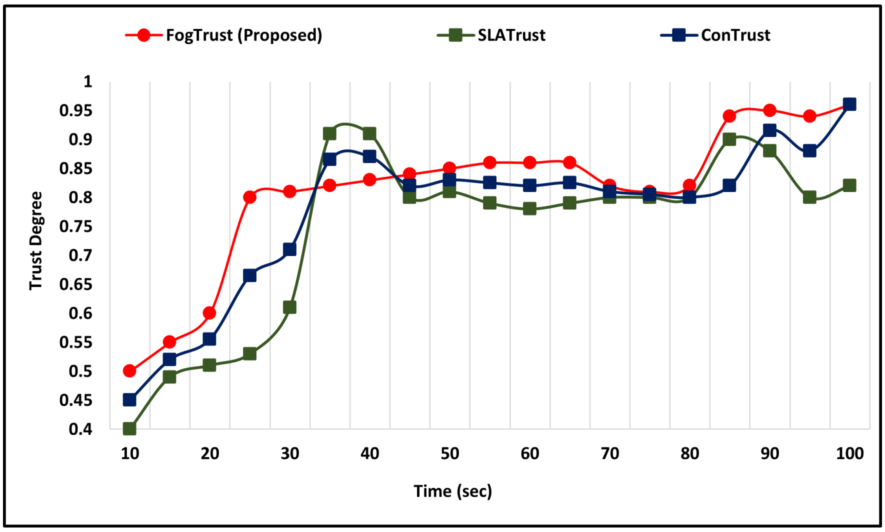

Figure 6. Comparative Analysis Against Bad-Mouthing Attacks.

5. Conclusions

The Internet of Things (IoT) is widely used in various industries, however, IoT nodes often struggle to maintain security on their own, making them susceptible to various attacks. To mitigate these risks, many mechanisms based on privacy and trust management have been proposed. However, current approaches neglect some features of central trust authority communications and the importance of central authority trust management, such as trust agents. The proposed FogTrust is effective in managing trust in the communication of fog computing with IoT devices. Other trust management mechanisms have been proposed, but they ignore the deployment of a centralized trust authority before the fog layer. To enhance the accuracy and reliability of FogTrust, a central authority, i.e., trust agents, is deployed. This central trust authority improves accuracy while reducing the computational weight on IoT nodes, which enhances resistance against attacks, reduces vulnerability, and provides standard security. The overall detection of malicious nodes in the proposed FogTrust mechanism ranges between 50% to 75% when compared with existing approaches. The *PM* can be further enhanced by identity, naming, and certificate allocation, and the security can be increased by encrypting the shared trust degree with the fog.

Author Contributions: Conceptualization, K.A.A. and I.U.D.; methodology, A.A.; software, A.R. and K.A.A.; validation, K.A.A., I.U.D. and A.A.; formal analysis, K.A.A. and M.A.; investigation, I.U.D., A.A. and M.A.; resources, A.A.; data curation, M.A. and A.A.; writing—original draft preparation, A.R.; writing—review and editing, K.A.A., I.U.D. and A.A.; visualization, A.A.; supervision, I.U.D.; project administration, A.A.; funding acquisition, A.A. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Deanship of Scientific Research at King Saud University, Riyadh, Saudi Arabia, through the Vice Deanship of Scientific Research Chairs: Chair of Cyber Security.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

CA	Central Authority
CENT	Centralized
CT	Current Trust
DIST	Distributed
DO	Direct Observation
FS	Fog Server
IO	Indirect Observation
PM	Proposed Mechanism
PT	Previous Trust
SP	Service Provider
SR	Service Requester
TD	Trust Degree
TE	Trustee
TM	Trust Management
TMS	Trust Management System
TR	Trustor

References

1. Koohang, A.; Sargent, C.S.; Nord, J.H.; Paliszkiewicz, J. Internet of Things (IoT): From awareness to continued use. *Int. J. Inf. Manag.* **2022**, *62*, 102442. [CrossRef]
2. Ashton, K. That 'internet of things' thing. *RFID J.* **2009**, *22*, 97–114.
3. Abid, M.A.; Afaqui, N.; Khan, M.A.; Akhtar, M.W.; Malik, A.W.; Munir, A.; Ahmad, J.; Shabir, B. Evolution towards smart and software-defined internet of things. *AI* **2022**, *3*, 100–123. [CrossRef]
4. Babangida, L.; Perumal, T.; Mustapha, N.; Yaakob, R. Internet of Things (IoT) Based Activity Recognition Strategies in Smart Homes: A Review. *IEEE Sens. J.* **2022**, *22*, 8327–8336. [CrossRef]
5. Trovato, V.; Sfameni, S.; Rando, G.; Rosace, G.; Libertino, S.; Ferri, A.; Plutino, M.R. A Review of Stimuli-Responsive Smart Materials for Wearable Technology in Healthcare: Retrospective, Perspective, and Prospective. *Molecules* **2022**, *27*, 5709. [CrossRef]
6. Awan, K.A.; Ud Din, I.; Almogren, A.; Almajed, H. AgriTrust—A trust management approach for smart agriculture in cloud-based internet of agriculture things. *Sensors* **2020**, *20*, 6174. [CrossRef]
7. Mishra, V.K.; Tripathi, R.; Tiwari, R.G.; Misra, A.; Yadav, S.K. Issues, Challenges, and Possibilities in IoT and Cloud Computing. In *Proceedings of the International Conference on Computational Intelligence in Pattern Recognition*; Springer: Singapore, 2022; pp. 326–334.
8. George, A.; Ravindran, A.; Mendieta, M.; Tabkhi, H. Mez: An adaptive messaging system for latency-sensitive multi-camera machine vision at the iot edge. *IEEE Access* **2021**, *9*, 21457–21473. [CrossRef]
9. Bhat, S.A.; Huang, N.F.; Sofi, I.B.; Sultan, M. Agriculture-Food Supply Chain Management Based on Blockchain and IoT: A Narrative on Enterprise Blockchain Interoperability. *Agriculture* **2021**, *12*, 40. [CrossRef]
10. Farhan, L.; Kharel, R.; Kaiwartya, O.; Quiroz-Castellanos, M.; Alissa, A.; Abdulsalam, M. A concise review on Internet of Things (IoT)-problems, challenges and opportunities. In Proceedings of the 2018 11th International Symposium on Communication Systems, Networks & Digital Signal Processing (CSNDSP), Budapest, Hungary, 18–20 July 2018; pp. 1–6.
11. Abiodun, O.I.; Abiodun, E.O.; Alawida, M.; Alkhawaldeh, R.S.; Arshad, H. A review on the security of the internet of things: Challenges and solutions. *Wirel. Pers. Commun.* **2021**, *119*, 2603–2637. [CrossRef]
12. Din, I.U.; Guizani, M.; Hassan, S.; Kim, B.S.; Khan, M.K.; Atiquzzaman, M.; Ahmed, S.H. The Internet of Things: A review of enabled technologies and future challenges. *IEEE Access* **2018**, *7*, 7606–7640. [CrossRef]
13. Zhang, J.; Shen, C.; Su, H.; Arafin, M.T.; Qu, G. Voltage over-scaling-based lightweight authentication for IoT security. *IEEE Trans. Comput.* **2021**, *71*, 323–336. [CrossRef]
14. Ouaddah, A.; Mousannif, H.; Abou Elkalam, A.; Ouahman, A.A. Access control in the Internet of Things: Big challenges and new opportunities. *Comput. Netw.* **2017**, *112*, 237–262. [CrossRef]
15. Sahay, R.; Meng, W.; Estay, D.S.; Jensen, C.D.; Barfod, M.B. CyberShip-IoT: A dynamic and adaptive SDN-based security policy enforcement framework for ships. *Future Gener. Comput. Syst.* **2019**, *100*, 736–750. [CrossRef]
16. Garg, S.; Kaur, K.; Kaddoum, G.; Garigipati, P.; Aujla, G.S. Security in IoT-driven mobile edge computing: New paradigms, challenges, and opportunities. *IEEE Netw.* **2021**, *35*, 298–305. [CrossRef]
17. Tanwar, S.; Gupta, N.; Iwendi, C.; Kumar, K.; Alenezi, M. Next Generation IoT and Blockchain Integration. *J. Sens.* **2022**, *2022*, 9077348. [CrossRef]
18. Mendieta, M.; Neff, C.; Lingerfelt, D.; Beam, C.; George, A.; Rogers, S.; Ravindran, A.; Tabkhi, H. A Novel Application/Infrastructure Co-design Approach for Real-time Edge Video Analytics. In Proceedings of the 2019 SoutheastCon, Huntsville, AL, USA, 11–14 April 2019; pp. 1–7.
19. Haseeb, K.; Alzahrani, F.A.; Siraj, M.; Ullah, Z.; Lloret, J. Energy-Aware Next-Generation Mobile Routing Chains with Fog Computing for Emerging Applications. *Electronics* **2023**, *12*, 574. [CrossRef]

20. Saad, Z.M.; Mhmood, M.R. Fog computing system for internet of things: Survey. *Tex. J. Eng. Technol.* **2023**, *16*, 1–10.
21. Ruan, H.; Gao, H.; Qiu, H.; Gooi, H.B.; Liu, J. Distributed operation optimization of active distribution network with P2P electricity trading in blockchain environment. *Appl. Energy* **2023**, *331*, 120405. [CrossRef]
22. Gupta, P.; Saini, D.K. Introduction to Optimization in Fog Computing. In *Bio-Inspired Optimization in Fog and Edge Computing Environments*; Auerbach Publications: New York, NY, USA, 2023; pp. 1–24.
23. Kar, B.; Yahya, W.; Lin, Y.D.; Ali, A. Offloading using Traditional Optimization and Machine Learning in Federated Cloud-Edge-Fog Systems: A Survey. *IEEE Commun. Surv. Tutor.* **2023**. [CrossRef]
24. Latif, R. ConTrust: A novel context-dependent trust management model in social Internet of Things. *IEEE Access* **2022**, *10*, 46526–46537. [CrossRef]
25. Chang, V.; Sidhu, J.; Singh, S.; Sandhu, R. SLA-based Multi-dimensional Trust Model for Fog Computing Environments. *J. Grid Comput.* **2023**, *21*, 1–19. [CrossRef]
26. Din, I.U.; Bano, A.; Awan, K.A.; Almogren, A.; Altameem, A.; Guizani, M. LightTrust: Lightweight trust management for edge devices in industrial internet of things. *IEEE Internet Things J.* **2021**. [CrossRef]
27. George, A.; Ravindran, A. Scalable approximate computing techniques for latency and bandwidth constrained IoT edge. In *Proceedings of the International Summit Smart City 360°*; Springer: Cham, Switzerland, 2021; pp. 274–292.
28. Al Muhtadi, J.; Alamri, R.A.; Khan, F.A.; Saleem, K. Subjective logic-based trust model for fog computing. *Comput. Commun.* **2021**, *178*, 221–233. [CrossRef]
29. Baghalzadeh Shishehgarkhaneh, M.; Keivani, A.; Moehler, R.C.; Jelodari, N.; Roshdi Laleh, S. Internet of Things (IoT), Building Information Modeling (BIM), and Digital Twin (DT) in Construction Industry: A Review, Bibliometric, and Network Analysis. *Buildings* **2022**, *12*, 1503. [CrossRef]
30. Rahman, F.H.; Au, T.W.; Newaz, S.S.; Suhaili, W.S. Trustworthiness in fog: A fuzzy approach. In Proceedings of the 2017 VI International Conference on Network, Communication and Computing, Kunming, China, 8–10 December 2017; pp. 207–211.
31. Namal, S.; Gamaarachchi, H.; MyoungLee, G.; Um, T.W. Autonomic trust management in cloud-based and highly dynamic IoT applications. In Proceedings of the 2015 ITU Kaleidoscope: Trust in the Information Society (K-2015), Barcelona, Spain, 9–11 December 2015; pp. 1–8.
32. Al-Khafajiy, M.; Baker, T.; Asim, M.; Guo, Z.; Ranjan, R.; Longo, A.; Puthal, D.; Taylor, M. COMITMENT: A fog computing trust management approach. *J. Parallel Distrib. Comput.* **2020**, *137*, 1–16. [CrossRef]
33. Alemneh, E.; Senouci, S.M.; Brunet, P.; Tegegne, T. A two-way trust management system for fog computing. *Future Gener. Comput. Syst.* **2020**, *106*, 206–220. [CrossRef]
34. Dhelim, S.; Kechadi, T.; Aung, N.; Ning, H.; Chen, L.; Lakas, A. Trust2Vec: Large-Scale IoT Trust Management System based on Signed Network Embeddings. *arXiv* **2022**, arXiv:2204.06988.
35. Ogundoyin, S.O.; Kamil, I.A. A trust management system for fog computing services. *Internet Things* **2021**, *14*, 100382. [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.