Exploring Trust Management in Fog Computing: A Comprehensive Review and Future Challenges in Task Offloading

Liu Feng, Suhaidi Hassan, Mohammed Alsamman
School of Computing (SoC), University Utara Malaysia (UUM), 06010 Sintok, Kedah, Malaysia

***Abstract*—With the proliferation of data-driven services and latency-sensitive applications, fog computing has emerged as a pivotal extension of cloud infrastructure, enabling data processing and resource allocation at the network edge. However, the trustworthiness of task offloading in such decentralized and heterogeneous environments remains insufficiently explored, posing significant concerns related to system reliability, security, and performance. This review aims to address this gap by providing a comprehensive and systematic analysis of current research on trust-based task offloading in fog computing. The study investigates various trust evaluation mechanisms, categorizing them into three major paradigms: Direct Trust-based, Recommended Trust-based, and Comprehensive Trust. Through this classification, the study identifies and examines key trust-related metrics that influence offloading decisions, including task execution accuracy, trust evaluation accuracy, and evaluation latency. A critical assessment of the strengths and limitations of existing approaches reveals ongoing challenges such as dynamic trust management, scalability in large-scale networks, interoperability among diverse nodes, and resilience against malicious behaviours. Based on these insights, the study highlights pressing research opportunities and recommends the development of lightweight, adaptive, and context-aware trust frameworks capable of supporting real-time decision-making in dynamic fog environments. By synthesizing fragmented research and offering a forward-looking perspective, this review contributes a foundational reference for scholars and practitioners seeking to enhance the reliability and security of task offloading in fog computing, thereby supporting the evolution of more robust and efficient edge-based computing infrastructures.**

Keywords*—Trust management; fog computing; cloud computing; task offloading; heterogeneous networks; trust evaluation; task completion time

I. Introduction

Recently, the swift evolution of cloud computing, along with the rise of mobile smart devices and wireless networking technologies, has prompted a reassessment of computing architecture to better support mobile terminals. This adaptation aims to reduce system latency and enhance the user experience [1]. Therefore, Cisco introduced the concept of fog computing, defining it as a highly virtualized platform that shifts computational tasks from centralized cloud data centre to edge devices of the network [2]. This approach addresses the inherent limitations of the cloud computing architecture, which centralizes computing far from data sources, resulting in increased latency, congestion, reduced reliability, and augmented security vulnerabilities. Around a decade after the emergence of cloud computing, fog computing garnered extensive attention as an optimization of the existing cloud computing infrastructure [3].

Fog computing, in essence, represents a form of cloud computing with a local-oriented nature. It acts as an addition to the conventional cloud computing paradigm. In this regard, when cloud computing is paralleled to Wide Area Network (WAN) - based computing, fog computing can be compared to Local Area Network (LAN) - based computing. In a similar vein, just as the Content Delivery Network (CDN) deals with the problem of local caching within the TCP/IP framework, fog computing endeavors to solve the intrinsically local computing difficulties presented by cloud computing [4]. A pivotal moment in the development of fog computing occurred in November 2015, when industry giants such as Cisco, ARM, Dell, Intel, Microsoft, and Princeton University Edge Lab collaborated to establish the OpenFog Consortium [5]. The primary objective of this consortium was to promote and expedite the adoption of open fog computing, thereby facilitating advancements in the Internet of Things (IoT) domain. The fog computing market is anticipated to experience robust growth, with a projected Compound Annual Growth Rate (CAGR) of 67.90% during the forecast period of 2021-2026 [6]. Despite being a relatively new domain in the technological landscape, fog computing has rapidly established itself as a significant field.

Fog computing, introduced by Cisco, has emerged from the rapid evolution of cloud computing and mobile technologies. It shifts computational tasks from centralized data centres to edge devices, addressing latency, congestion, and security issues. Acting as a local extension of cloud computing, it parallels LAN to WAN. The OpenFog Consortium, formed in 2015 by industry leaders, promotes this technology and supports the Internet of Things (IoT). The fog computing market is expected to grow significantly, with a CAGR of 67.90% from 2021 to 2026, highlighting its growing importance in technology.

II. Fog Computing Concept

The architecture of fog computing is structured into a three-tier system extending from the network core to the periphery. This system comprises the cloud layer, the fog layer, and the terminal layer, as illustrated in Fig. 1. Compared to traditional computing paradigms, fog computing offers several advantages. These advantages include significantly reduced latency, conserved backbone bandwidth, enhanced support for

high mobility, extensive geographical distribution, interoperability, and reduced energy consumption [7, 8].

With the development of grid computing, cloud computing, and fog computing as a complement to cloud computing came into being. Fog computing cannot replace cloud computing and needs to work together with cloud computing. Fog computing, as an emerging computing paradigm focused on providing services close to the user side, and it doesn't require all data to be sent to the cloud computing centre. It builds an infrastructure for IoT-oriented distributed computing and provides various services, including computing, storage and network connectivity between interconnected devices at the edge of the network and cloud computing platforms in the core of the network. By extending the capabilities of cloud computing to the edge of the network, users can analyse and manage data locally.

All in all, fog computing is an innovative computing model that concentrates on providing services such as computing, communication and storage for users on network edge devices close to them. Specifically, it utilizes network devices to realize services like computing, storage and network communication between cloud servers and mobile terminal devices. By expanding the fog layer with storage and computing capabilities, the data processing and computing processes are made closer to the terminal devices, and then three main functions, namely data caching, localized computing and wireless access, are provided for users. This can not only effectively relieve the computing and storage pressure on the cloud but also significantly improve the response speed of the entire application system. Moreover, it can better fit the characteristics of high mobility of mobile terminal devices and fully meet the strict requirements of mobile applications for high traffic transmission and low-latency response.

A. Cloud, Fog and Edge Computing

Unlike cloud computing, the main goals of fog and edge computing are similar in some ways; both enable services to be closer to the user and provide lower latency [10]. Fog computing performs as a bridge between cloud computing and mobile terminal devices, building a platform with remarkable performance in computing, digital storage and network services. Therefore, cloud computing and fog computing are not independent entities; on the contrary, they interact and influence each other. In practical applications, there are several main differences among cloud computing, fog computing and edge computing. As shown in Table I, these differences span over various aspects, including latency characteristics, network characteristics, location awareness and target users [11].

B. Fog Computing Application Scenarios

Since the inception of fog computing, a wide range of scientific research institutions, internet companies, and scholars have conducted extensive studies on the subject. This collaborative effort has led to the emergence of various fog computing platforms, including OpenFog, LocalGrid, and PrismTech Vortex. These platforms facilitate the deployment of fog computing solutions across diverse sectors. Typical applications of fog computing encompass automotive networking, where vehicles communicate and share data; wireless sensor networks, which enable real-time monitoring; and intelligent building control systems that optimize energy use. Additionally, fog computing plays a crucial role in the IoT, augmented reality experiences, mobile communications, and Software Defined Networking (SDN) applications. This wide array of applications illustrates the versatility and potential of fog computing in enhancing network efficiency and performance [12].

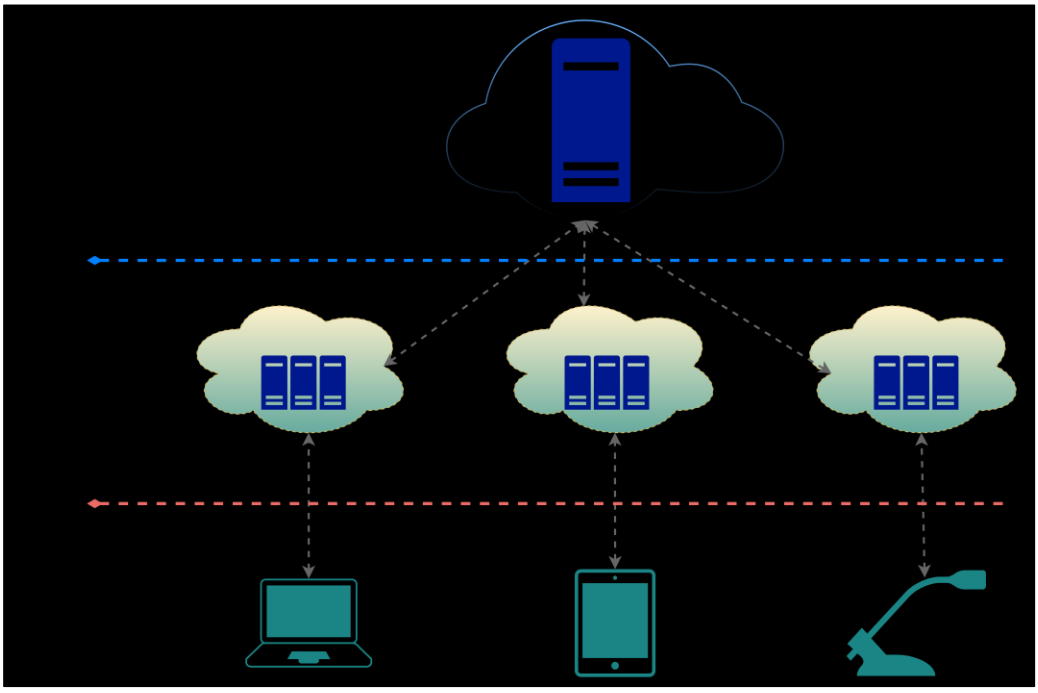

Fig. 1. Fog computing system architecture [9].

TABLE I COMPARISON ON DIFFERENT COMPUTING PARADIGMS

Attributes	Cloud Computing	Edge Computing	Fog Computing
Expected Task Execution Time	High	High-Medium Often	Low
Provided Services	Universal services	Often uses mobile networks	Vital for a particular domain and distributed
Security	Centralized (guaranteed by the Cloud provider)	Centralized (guaranteed by the Cellular operator)	Mixed (depending on the implementation)
Energy Consumption	High	Low	Varying but higher than for Edge
Identifying location	No	Yes	Yes
Main Providers	Amazon and Google	Cellular network providers	Proprietary
Interaction in Real-Time	Available	Available	Available
Latency	High	Low	Varying but higher than for Edge
Bandwidth Cost	High	Low	Low
Storage capacity and Computation	High	Very limited	Varying
Scalability	Average	High	High
Overall usage	Computation distribution for huge data (Google MapReduce), Apps virtualization, Storage of data scalability	Control of traffic, data caching, wearable applications	CCTV surveillance, imaging of subsurface in real-time, IoT, Smart city, Vehicle-to-Vehicle (V2X)

Fog computing as much as possible to the user's needs of the content and application services close to the end user, through this way, fog computing with support for mobile performance, low latency and high scalability and other advantages. Fog computing can be widely used in augmented reality (Augmented Reality, AR) and health monitoring and other real-time services; smart grid and local content distribution and other data dissemination areas; mobile big data processing and computing offloading and other distributed data areas; shopping centres and public transport and other temporary storage and other areas. In distributed data scenarios, fog computing efficiently handles user requests by leveraging the computing and storage capabilities of fog nodes. This approach alleviates the pressure on centralized cloud data processing and minimizes the response time for end-user data requests. Additionally, it reduces both the data traffic sent to the cloud and the storage requirements within it. Ultimately, this leads to a more balanced distribution of data processing demands across the cloud, fog, and terminal layers.

III. MOTIVATION

With the development of the Internet, fog computing has gradually become a new type of Internet infrastructure that has attracted much attention, but its research in the field of trusties is still insufficient [14]. Trust evaluation and management for fog nodes is an important means to improve the reliability of the network, which can effectively mitigate the security risks brought by the geographical dispersion of nodes and differences in the network environment. Although fog computing is more secure than cloud computing due to temporary data storage and local processing [15], inter-node collaboration may still trigger malicious behaviors, such as theft or manipulation of private data by malicious nodes. Traditional encryption solutions cannot effectively counter internal attacks from authenticated malicious fog nodes [16].

Trust management in fog computing faces critical challenges, particularly delayed trust value updates caused by computational overhead. While mechanisms like fuzzy logic trust evaluation, and multi-source feedback aggregation enhance trust and accuracy, they increase computational complexity, leading to high resource consumption and delayed updates in dynamic environments. Hierarchical trust models and reputation-based schemes, such as Gu et al. [17], improve reliability but introduce significant computational burdens. Similarly, adaptive mechanisms like those proposed by Almas et al. [18] struggle with timely updates due to overhead, and methods relying on recommendation information are vulnerable to malicious nodes. Fog computing faces several challenges in the process of trust management, particularly in balancing trust management and system performance. The complexity of trust models, such as reputation-based or subjective logic models, introduces significant computational overhead, making real-time decision-making difficult in large-scale environments. For instance, Alemneh et al. [19] highlight that while two-way trust management enhances security, it places a heavy computational burden on resource-constrained fog nodes. Moreover, approaches like the one proposed by Atwa et al. [20] experience difficulties in adapting to dynamic environments, such as Vehicular Ad Hoc Networks (VANETs), while models that aggregate multiple feedback sources, as discussed by Liang et al. [21], are prone to implementation complexity and require robust infrastructure. Furthermore, techniques like fuzzy logic and reputation-based mechanisms, employed by Bukhari et al. [22], introduce increased computational complexity that can hinder real-time performance. These challenges are compounded by the need for frequent updates in highly dynamic environments, which may lead to delays in trust evaluation and, consequently, suboptimal node selection.

The fog computing environment faces significant challenges related to reliability, primarily due to the participation of various vendors, including private cloud providers and multiple Internet Service Providers (ISPs). A fog network comprises numerous fog nodes, and ensuring their reliability is crucial during the task offloading. If the tasks are sent towards the malicious fog nodes, it can result in serious issues such as denial of service, jamming, spamming, impersonation, and data tampering [23, 24].

IV. CONTRIBUTION

This study advances research on trust-based task offloading in fog computing through several key contributions.

It first provides a systematic classification of trust evaluation mechanisms into three paradigms—Direct Trust-based, Recommended Trust-based, and Comprehensive Trust—consolidating fragmented research to clarify their principles and application scenarios, facilitating targeted comparisons for specific fog use cases.

Second, the work identifies and analyses critical trust-related metrics (task execution accuracy, trust evaluation accuracy, evaluation latency) and their interdependencies, offering a holistic reference for designing multi-objective trust frameworks that balance security and efficiency, critical for latency-sensitive fog applications.

Third, it critically assesses existing approaches, delineating core challenges including dynamic trust management in volatile networks, scalability in large-scale deployments, interoperability barriers, and vulnerability to malicious behaviours. This analysis transcends summarization to guide researchers toward high-impact problems.

Fourth, the review highlights emerging opportunities and recommends lightweight, adaptive, context-aware trust frameworks for real-time decision-making in dynamic fog environments, bridging research and practical implementation with actionable roadmaps.

Finally, by synthesizing knowledge into a unified perspective, it serves as a foundational reference for academia (guiding state-of-the-art advancements) and industry (supporting robust system deployment), advancing fog computing as a pivotal cloud extension.

V. ORGANIZATION

The organization of the study is designed to provide an in-depth exploration of fog computing, beginning with an Introduction that establishes the topic's significance in the context of modern technology, as shown in Fig. 2.

This section sets the stage for understanding how fog computing enhances traditional computing frameworks. Following this, the study delves into the Fog Computing Concept, where it outlines the core principles that define fog computing and its advantages over conventional cloud computing. Next, the study differentiates among Cloud, Fog, and Edge Computing, providing clarity on their unique roles and how they interact within the broader computing ecosystem. This distinction is crucial for understanding the specific contributions of fog computing. The section on Fog Computing Application Scenarios highlights various real-world implementations, showcasing its transformation influence across several industries that includes smart cities, healthcare, and industrial automation [13].

The Motivation section articulates the pressing challenges that fog computing aims to address, such as latency issues and bandwidth constraints, emphasizing its relevance in today's data-driven world. Following this, the Contribution section details the novel insights and findings that the study presents, adding to the existing body of knowledge on fog computing. An overview of the study's structure is provided in the Organization section, guiding the reader through the subsequent discussions. The main body focuses on Trust in Fog Computing, which is critical for ensuring secure operations in distributed environments. This includes an examination of Trust Management strategies, the design of a Trusted Task Offloading Architecture, and the various methods involved in the Trust Evaluation Process. The Task Offloading Process is also explained, detailing how tasks are managed within a fog framework.

The study further reviews specific techniques for Trust Evaluation in Fog Computing and presents a Comparative Analysis of Calculation Methods, evaluating different approaches to trust assessment. In addition to these discussions, the section on Classical Algorithms for Task Offloading examines traditional algorithms that facilitate efficient task management in fog environments, with a focused look at the particle swarm optimization algorithm and its relevance. A comprehensive Related Work section reviews existing literature, identifying gaps in current research and highlighting contributions to the field. The study concludes with a discussion on Future Challenges, addressing potential obstacles that may arise as fog computing evolves. Finally, the Conclusion summarizes the key findings and emphasizes the critical roles of trust and task offloading in the advancement of fog computing, reinforcing its importance in the current technological landscape.

Fig. 2. Organization of the study.

VI. Trust in Fog Computing

Trust is essential in fog computing due to its decentralized architecture and proximity to end-users, requiring robust mechanisms to ensure data integrity and reliable service delivery. This is achieved through several interconnected processes:

1) Trust management: Establishes and maintains trust relationships between nodes.

2) Trusted task offloading architecture: Ensures tasks are only delegated to trustworthy nodes.

3) Trust evaluation process: Defines methods for assessing the trustworthiness of nodes.

4) Task offloading process: Integrates trust verification prior to data transfer.

5) Dynamic trust assessment: Employs specific techniques to evaluate trust levels in real time.

6) Comparative analysis of calculation methods: Assesses the effectiveness of different trust calculation approaches.

Together, these components create a comprehensive framework that fosters a secure and reliable fog computing environment.

C. Trust Management

Fog nodes in fog computing networks are capable of executing data processing tasks near the data sources, which enhances support for real-time applications and reduces dependency on remote cloud resources. However, security and privacy concerns remain prominent within these networks.

Although fog computing is often perceived as more secure than traditional cloud computing due to the localized storage and analysis of data, vulnerabilities still exist. For instance, when fog nodes collaborate and share data to complete tasks, they may expose themselves to risks. Offloading tasks to untrustworthy fog nodes can lead to unauthorized access or manipulation of sensitive user information, raising significant privacy concerns.

To address these challenges, effective management of trust within fog nodes is vital. Implementing robust trust frameworks can help ensure that only reliable nodes are used for data processing. This includes assessing the trustworthiness of nodes, establishing secure communication channels, and continuously monitoring node behavior to prevent malicious activities. By prioritizing trustworthy management, fog computing can enhance its security posture, safeguarding user data and maintaining the integrity of services. Sunilkumar S. Manvi and others [25] explored various aspects of trust management within the realm of fog computing. Trust is a critical factor in the contemporary networked landscape, significantly influencing users' willingness to adopt fog computing solutions. In their research, they emphasize that trust management systems can build user confidence by ensuring secure and transparent data processing and storage. Effective trust management mitigates risks related to data integrity and privacy while enhancing the reliability of fog networks. By implementing clear trust protocols and assessment mechanisms, users can feel assured that their information is handled appropriately, which is crucial for the widespread adoption and success of fog computing technologies.

D. Trusted Task Offloading Architecture

As depicted in Fig. 3, this solution primarily focuses on the task offloading mechanism based on trust values within fog networks. In this architecture, every fog node has the capability to directly share tasks with other fog nodes. Meanwhile, it is assumed that fog nodes can reach each other from any location within the same fog domain. In real-world task flow scenarios, the task arrival rate can vary significantly depending on the location of the fog nodes [26].

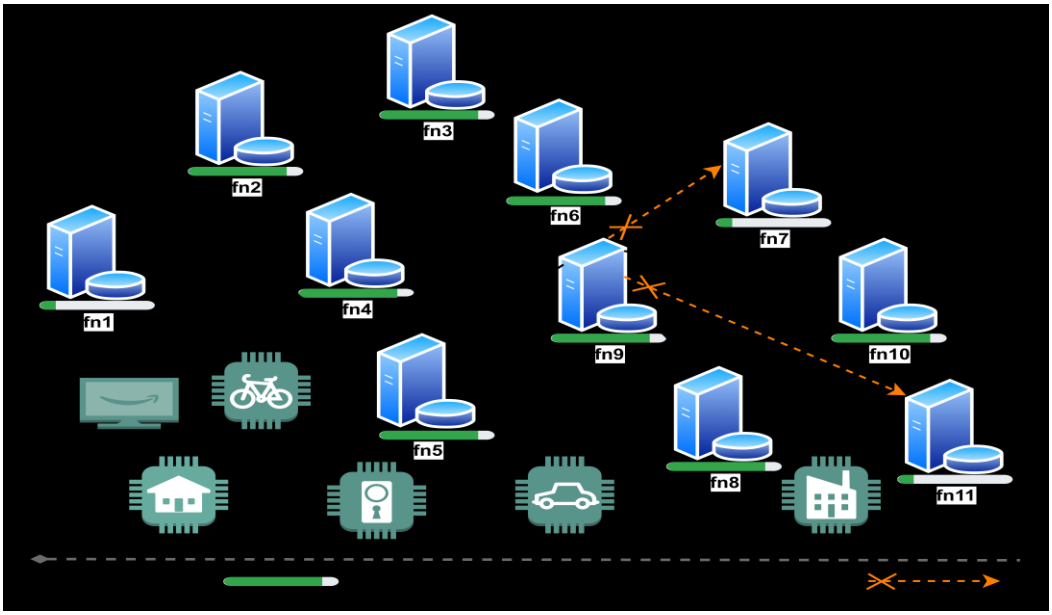

Fig. 3. Trusted task offloading architecture.

E. Trust Evaluation Process

Each fog node is equipped with a trust evaluation module that regularly assesses the trustworthiness of other fog nodes. Additionally, each node can function as both a task initiator and receiver, enabling task offloading to other fog nodes when necessary. When offloading a task, the node selects the most suitable fog node based on trust evaluation results and additional metrics such as completion time and latency [27].

Fog nodes can obtain each other's network load lightness information through local communication mechanisms. For example, network load information sharing: fog nodes can periodically exchange information about their own resource usage and task load among themselves. Such information includes CPU utilization, memory usage, task rank length, etc. The communication can be carried out through a neighbor-based communication policy. Trust evaluation is done by the fog nodes themselves. Each fog node can initiate computational detective tasks and select other fog nodes to handle these tasks based on the trust evaluation results. Fig. 4 shows the flowchart of trust evaluation, and the steps of trust evaluation can be represented as:

a) Initialization: All fog nodes set the same initial trust value.

b) Trust evaluation: Adjusting the trust value according to the completion of tasks by other fog nodes. When the network load is light, it is performed by sending some probing tasks with less data volume to the fog nodes with free resources. When the network load is large, i.e., exceeds the pre-set load threshold, the frequency of trust evaluation is reduced, or the trust value of the collaborators is evaluated by the fog nodes during the task collaboration.

c) When a fog node needs to offload a task, it identifies the most appropriate node by considering trust evaluation results and time delays. This strategy guarantees that tasks are directed to dependable fog nodes that offer optimal performance. As a result, overall efficiency and security are significantly improved.

F. Task Offloading Process

As shown in Fig. 2, there are two types of fog nodes present in the fog layer: trusted fog nodes and untrusted fog nodes. At the time of offloading, according to the different number of task packets and priority of the tasks (indicated by different colors in the figure), which are distributed to different fog nodes. Each fog node maintains a task processing rank for tasks waiting to be processed, and each task has its arrival time, processing time and deadline. The relationship between the task processing rank and the time delay is as follows: the length of the task processing rank, which represents the number of tasks in the rank, may affect the waiting time of the tasks. The longer the rank, the longer the waiting time may be, which leads to an increase in the total time delay of the task. When a fog node fn_9 initiates a task offloading interaction, it selects a neighbouring fog node to interact with. Neighboring node fn_7 has a short task rank but a low trust value and hence does not interact with node fn_7. Whereas, neighboring node fn_{10} has both low latency and high trust and hence chooses to interact with node fn_{10} on tasks.

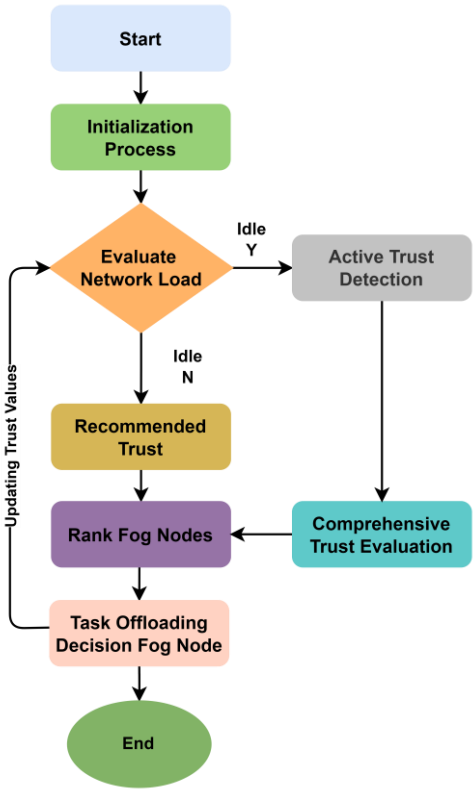

Fig. 4. Trust evaluation process [28].

G. Fog Computing Concept

Trust management in direct trust, recommended trust, and comprehensive trust:

a) Direct trust: In the fog-to-fog collaboration model, a fog node evaluates its interactions with other fog nodes, such as during task offloading or resource sharing. It updates its trust levels based on the outcomes of these interactions. For instance, the trust value of a fog node that successfully completes a task increases, while the value for a node that fails or produces subpar results decreases. The results of these trust calculations are recorded locally to assess the trustworthiness of each node.

b) Recommended trust: Fog nodes can request trustworthiness recommendations about third-party fog nodes from other fog nodes. This recommendation trust information helps fog nodes to understand the trust status of nodes in the entire network and thus make more informed task offloading decisions. However, the recommendation trust information can be affected by malicious nodes spreading false information. However, the recommended trust information may be affected by malicious nodes spreading false information. To ensure the reliability of the recommendation trust information, the trustworthiness of the recommender and the similarity of the recommendation information (recommender's trustworthiness) are calculated. Similarity (weight of recommendation trust) of the recommendation information. In this way, after obtaining direct trust and recommendation trust information about other fog nodes.

c) Comprehensive trust: The comprehensive trust value is a metric that thoroughly assesses the reliability and trustworthiness of fog nodes. It combines two key dimensions: direct trust value and recommended trust value. This integrated approach offers a solid foundation for making task offloading decisions in fog computing. By providing a clearer picture of fog nodes' performance during task allocation, the comprehensive trust value helps in selecting the most suitable nodes for task assignments.

H. Comparative Analysis of Calculation Methods

Each trust value calculation method has its unique advantages and disadvantages. Choosing a suitable calculation method and improving it with optimization suggestions can better address the challenges in specific application scenarios and improve the accuracy and efficiency of trust value calculation [20]. In Table II, combining the advantages and weaknesses of the seven algorithms mentioned above, along with their application scenarios, suggests that the fuzzy logic algorithm is particularly well-suited for real-time task offloading needs.

TABLE II MAINSTREAM WAYS OF TRUST EVALUATION

Techniques	Advantages	Weaknesses	Applicable Scenarios
Weighted Sum Method	Simple to implement and flexible	Dependent on weight settings, linear assumptions	Used in scenarios where there are fewer indicators and the weights are relatively easy to determine
Fuzzy Logic	Flexible rule definition for complex environments	Too many rules may lead to computational inefficiency	Fuzzy logic is suitable for scenarios dealing with uncertainty and ambiguity
Bayesian Inference	Dynamically updated and adaptable	Higher computational complexity	Used in scenarios where trust needs to be updated dynamically
Trust Chain Model	Highly flexible	Over-reliance on network structure	Distributed network environments for multi-hop trust delivery
Machine Learning Methods	Ability to handle complex multi-dimensional data and non-linear relationships	Requires large amounts of high-quality training data	Ideal for scenarios dealing with complex, multi-dimensional data
Graph-Based Trust Calculation	Ability to globally evaluate the trust value of nodes in the network	Slower response to dynamic changes, which may lead to lags	Suitable for social networks, P2P networks and other scenarios
Reputation-Based Trust Calculation	Flexibility to adjust the final trust value according to different evaluation sources and weights	Vulnerable to malicious ratings or subjective bias	Used in scenarios where reputation needs to be assessed and built up over time

VII. CLASSICAL ALGORITHMS FOR TASK OFFLOADING

The task of offloading problems in fog computing is typically considered as either a single-objective or multi-objective optimization problem, subject to diverse constraints. These constraints include task completion latency, the quantity of tasks fulfilled within the deadline, the degree of load balancing, the energy consumption of the fog computing system, and the overall service quality. This section outlines several commonly used algorithms for addressing task offloading challenges. Among the algorithms discussed are heuristic approaches such as the PSO algorithm and the Ant Colony (AC) algorithm, as well as the Min-Min algorithm. These methods provide crucial insights and form the basis for the research explored in this study.

A. Particle Swarm Optimization Algorithm

The PSO originated from scholars' research on the feeding behavior of bird flocks and was first proposed by Kennedy and Eberhart [29]. The particle swarm algorithm has become a hot research topic, and has been used by scholars to produce a large number of research results in various fields, and has been continuously improved to produce a number of improved PSO algorithms, such as the binary PSO algorithm, the heterogeneous PSO algorithm, the adaptive PSO algorithm, the cooperative PSO algorithm, and the discrete PSO algorithm, etc, [30]. Compared with other intelligent optimization algorithms, particle swarm optimization algorithms are easy to implement and efficient for solving combinatorial optimization problems.

The main process of the standard particle swarm optimization algorithm is as follows:

1) Initialisation: Initialise the particles in the population by assigning random initial values to their positions and velocities. The positions and velocities of the particles can be initialised according to the following equation, as shown in Eq. (1) and Eq. (2):

$$x_i = r(X_{max} - X_{min}) + X_{min} \quad (1)$$

$$v_i = r(V_{max} - V_{min}) + V_{min} \quad (2)$$

In the above equations, "r" represents a random number between 0 to 1, "X_{max}" and "X_{min}", respectively, denote the maximum and minimum positions of the particle, and "V_{max}" and "V_{min}", respectively, denote the maximum and minimum velocities of the particle. The velocity and position of the particle cannot exceed the specified maximum and minimum ranges.

The fitness values of particles within a population are determined by setting a fitness function. This fitness function should be established based on the specific objective function that requires optimization.

The fitness value of each particle's current position within the population is compared with the fitness value of its historical best position. Through this comparison, the best fitness value for each individual particle can be obtained.

The fitness values of the current positions of all particles in the population are compared with the fitness value of the global best position. As a result, the global best fitness value of the population can be determined.

Update the velocity and position of particles. For the $(t+1)$ generation iteration, update each particle's velocity and position [see Eq. (3) and Eq. (4)]:

$$v_{id}(t+1) = v_{id}(t) + c_1 r_1(t)[p_{id}(t) - x_{id}(t)] + c_2 r_2[p_{gd}(t) - x_{id}(t)] \quad (3)$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \quad (4)$$

where, r_1 and r_2 denote random numbers between (0,1), p_{id} and p_{gd} are the optimal positions of the particle and the population, respectively, c_1 and c_2 are the acceleration constants. One of the common optimization methods for particle swarm algorithms is to add inertia weights to control the current velocity of the inherited particles when updating their positions. ω includes a check to see if the maximum number of iterations has been reached, or if the desired fitness value has been reached. If the termination condition is met, the individual with the highest fitness value is output.

B. Ant Colony Algorithm

The Ant Colony Optimization (ACO) algorithm is a swarm intelligence approach inspired by the foraging behavior of ants and driven by positive feedback. When ants search for food, they release pheromones to communicate with the colony, naturally favoring shorter paths that lead to the target more quickly. As ants traverse these paths, pheromone accumulation increases, making those routes more attractive to others. Over time, pheromones evaporate, ensuring the colony gravitates towards the path with the highest concentration, ultimately leading to the most efficient solution.

Similar to the particle swarm optimization algorithm, ACO is a highly effective parallel search method capable of solving complex combinatorial optimization problems [31]. The typical process of an ACO algorithm follows these steps:

Initialize parameters: Initialize parameter information, such as colony size and pheromone.

Place ants: Randomly place ants in the colony on the path.

Path selection: The ants choose the path according to the difference of pheromone concentration on the path, and the probability of choosing the path (i,j) is Eq. (5):

$$p_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}(t)]^\beta}{\sum_{s \in J_k(i)} [\tau_{is}(t)]^\alpha \cdot [\eta_{is}(t)]^\beta}, & j \in J_k(i) \\ 0, & \text{others} \end{cases} \quad (5)$$

where, $(J_k(i) = \{1,2,\dots,n\} - tabu_k)$ is the set of next paths that the ant can choose, $tabu_k$ represents the set of paths that the ant has already travelled, and the paths that have already been chosen will be added to the taboo list for no further choices, and the ant completes a path selection when all paths have been added to the taboo list; τ_{ij} is the pheromone concentration of the path (i,j) at this point in time. Pheromone concentration; η_{ij} is the heuristic factor with $\eta_{ij} = 1/b_{ij}$, the expected degree of ant κ choosing the path (i,j). Where, α is the pheromone heuristic factor, indicating the degree of importance of the pheromone, and β indicates the degree of importance of the heuristic factor. Generally, α takes a constant value between 1 and 4, with its value positively correlated to the influence of the pheromone. To find the optimal path, once all the ants have completed their path selection, the best path among all the ants is selected. The pheromone is then updated. In this iteration, the pheromone on the path remains constant, and the update occurs according to the following Eq. (6) and Eq. (7):

$$\tau_{ij}(t+n) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij} \quad (6)$$

$$\Delta\tau_{ij} = \sum_{k=1}^{m} \Delta\tau_{ij}^k \quad (7)$$

where, $\rho(0 < \rho < 1)$denotes the evaporation coefficient of the pheromone on the path, $\Delta\tau_{ij}$ denotes the increment of pheromone in this iteration. $\Delta\tau_{ij}^k$ denotes the pheromone produced by ant k on path (i, j).

C. Min-Min Algorithm

The Min-Min algorithm was originally proposed to solve the task offloading problem in Grid computing [32]. The idea is first to map tasks with small computational resource requirements and schedule these small tasks to servers with high computational power, i.e., fast execution speed. The typical execution steps of the Min-Min algorithm are: Compute the expected completion time of every task on each available server. Then, identify the earliest completion time for each task and its corresponding server. Locate the task with the shortest earliest completion time from the task list. Dispatch this task to the server according to the earliest time calculated in Step (I).

Update the task list on the server with the earliest time. After the scheduling, update the expected completion times of other tasks in the task list on the available server, and remove this task from the task set. Repeat the above steps until all the tasks in the task list have been scheduled.

The Min-Min algorithm is simple and easy to operate, but there are some problems. On one hand, it maps many small tasks to servers with high computing power, resulting in load imbalance, and on the other hand, it causes large tasks with high computational resource demand to have no free servers to process them, so the tasks cannot be processed as quickly as possible to meet the latency requirements.

VIII. Related Works

In the increasingly complex landscape of fog computing, trust management has emerged as a critical concern, with various mechanism developed to ensure secure and reliable interactions between IoT devices and fog nodes. Mahmood et al. [33] introduced a trust-based architecture that leverages digital certificates signed by fog nodes acting as certification authorities. This approach significantly enhances the security and privacy of communications within a trusted domain, but it faces challenges related to the complexity and cost of deploying

multiple fog nodes and managing revoked certificates. Building on this, Atwa et al. [20] developed a reputation-based model for VANETs, which integrates fog computing to reduce the need for extensive cloud communication by aggregating trust evaluations at the fog nodes. Although this model reduces message transmission overhead, it encounters difficulties in adapting to the highly dynamic nature of VANET environments.

Liang et al. [21] proposed a reliable trust computing mechanism that aggregates multisource feedback within a Social Sensor Cloud (SSC) environment. By using a fusion algorithm, this approach improves the accuracy of trust evaluations and enhances the detection of malicious nodes. However, the integration of multiple feedback sources introduces significant implementation complexity and requires robust infrastructure. Meanwhile, Mahmoud et al. [34] applied the Fuzzy Analytic Hierarchy Process (Fuzzy-AHP) to prioritize multiple trust criteria, such as Quality of Service (QoS) and Quality of Security (QoSec), to deliver more accurate and reliable trust assessments. Despite its comprehensiveness, this method is limited by the complexity of implementation and the potential for bias due to reliance on expert judgment.

Alemneh et al. [19] proposed a two-way trust management system using subjective logic, which allows both service requesters and providers to evaluate each other's trustworthiness. This approach enhances the overall security of the fog network by combining direct and indirect trust, but the computational burden introduced by subjective logic, particularly in processing large volumes of recommendation information, poses challenges for resource-constrained fog nodes. In addressing cybersecurity issues within Sensor-Cloud Systems (SCS), Wang et al. [35] introduced a hierarchical trust mechanism that employs real-time monitoring and behaviour analysis at the fog layer to quickly identify and isolate malicious nodes. While effective in threat detection, this system may experience significant computational and communication overheads in large-scale networks due to the multi-layered trust computations required.

Almas et al. [18] contributed to the field by proposing a context-based adaptive trust mechanism tailored for smart healthcare systems. By utilizing a Bayesian approach and similarity measures, the system dynamically adjusts trust weights to respond to environmental changes, effectively mitigating trust-related attacks. However, the system may face challenges in maintaining timely updates to trust values in highly dynamic network environments, potentially leading to misjudgments. Bukhari et al. [22] introduced the Fog Node Selection Engine (FNSE), an AI-driven framework that employs Fuzzy Logic, Logistic Regression, and Deep Neural Networks to predict the trustworthiness of fog nodes. This approach demonstrated superior accuracy but also increased computational complexity, which could impact the system's real-time performance.

Yadav and Baranwal [36] developed a trust management mechanism based on feedback credibility evaluation to combat malicious feedback in fog computing. The system enhances the accuracy of trust evaluations by using a checkers-based method to prioritize recent interactions. However, in highly dynamic environments, the computation of trust values may not reflect the latest state of nodes, leading to delays in decision-making. Gu et al. [15] proposed a reputation-based resource allocation scheme in community-based fog computing, which matches user tasks with fog nodes that meet specific reputation criteria, thereby improving service reliability. This approach, however, introduces significant computational overhead due to the complexity of reputation computations across multiple layers of communities and nodes.

Kochovski et al. [37] integrated blockchain technology into fog computing trust management by utilizing smart contracts and trustless oracles to manage trust relationships in real-time. While this approach enhances transparency and trustworthiness, it also increases computational complexity and latency, particularly in handling a large number of transactions. Zhang et al. [38] introduced a fog-based detection system (FDS) for Sensor-Cloud Systems, focusing on hierarchical trust evaluation to detect hidden data attacks. Although effective in detecting malicious activity, the system may struggle with timely updates to trust values in dynamic networks, impacting decision accuracy.

Rehman et al. [39] proposed FogTrust, a lightweight trust management mechanism for IoT that uses fuzzy logic and trust aggregation to detect and mitigate various IoT attacks, such as on-off good-mouthing and bad-mouthing attacks. Despite its effectiveness, the multiple trust calculation processes involved, particularly with fuzzy logic, may affect the real-time performance and responsiveness of the system.

Collectively, these studies illustrate the diverse mechanisms employed to manage trust in fog computing environments. They highlight the ongoing trade-offs between enhancing security, maintaining system performance, and managing computational complexity. As the field continues to evolve, future research must address the challenges of scalability and real-time application, ensuring that trust mechanisms remain robust and efficient even in large-scale, dynamic fog computing environments.

Continuing from the previous detailed exploration, additional studies have further expanded on trust management in fog computing, focusing on various aspects such as trust evaluations, feedback credibility, and resource allocation.

Muhtadi et al. [40] presented a trust model for fog computing that leverages subjective logic to evaluate trustworthiness among fog nodes. This model considers belief, disbelief, and uncertainty in its calculations, combining direct experiences with recommendations from other nodes to assess trust. While effective in identifying and mitigating malicious nodes, the model is susceptible to bias if the network contains a high number of malicious nodes. Moreover, the reliance on recommendation information could lead to inaccuracies if the sources of these recommendations are compromised.

Mazumdar et al. [41] introduced a trust-based load-offloading protocol specifically designed to optimize service delivery in fog computing for IoT applications. This protocol uses a distributed scheme where fog nodes collaborate to share the load based on trust evaluations. Trust is established through

a Fog Registration Center (FRC) that authenticates fog nodes using Shamir's secret sharing scheme, ensuring that only trusted nodes participate in load-sharing. Although this approach minimizes service delays and ensures timely task completion, it adds complexity, especially in high-load or emergency situations where the scheme's security mechanisms might introduce additional computational burdens.

Yadav and Baranwal [42] proposed a trust evaluation method that combines Quality of Service (QoS) attributes with social relationships among fog nodes. This multi-layered approach enhances the security of fog computing environments by minimizing the risk of malicious nodes disrupting operations. However, the approach's complexity, particularly in large-scale networks, may lead to significant computational overheads and delays, as multiple layers of trust computations and data analyses are required to accurately assess trustworthiness.

Rathee et al. [43] introduced a trust-based security framework designed to enhance communication security between fog nodes and IoT devices. This framework includes a Trust Manager (TM) that evaluates the legitimacy of fog nodes and IoT devices by calculating trust values and factors based on past interactions. The TM records these metrics in a lookup table, using optimization algorithms to ensure reliable communication paths. Although the framework effectively reduces the risk of malicious activities, the need to compute multiple levels of trust values introduces high computational complexity, potentially impacting the system's real-time performance.

Ben Daoud et al. [44] developed the TACRM model, which integrates access control with trust and resource management in fog computing environments. The model computes trust based on user behaviors and assigns trust levels to users, which are then used for access control decisions. This approach enhances security by monitoring user activities and ensuring that resources are allocated securely and efficiently. However, the continuous monitoring required by the system raises potential privacy concerns, particularly in sensitive or personal data contexts.

Hamza et al. [45] proposed a bi-directional trust management system for fog computing, which uses both direct and indirect trust computations to establish trust between Service Requesters (SR) and Service Providers (SP). The system employs fuzzy logic to aggregate trust scores, with a decay function modelling the influence of past interactions on current trust assessments. While this approach effectively mitigates common trust-related attacks, such as ballot stuffing and self-promotion, the introduction of Bayesian inference and centrality into the trust evaluation process increases computational complexity, which may affect real-time performance in large-scale networks.

Ogundoyin and Kamil [16] introduced a trust management system that also uses fuzzy logic to evaluate trust in fog computing environments. This system integrates direct trust from self-observation with indirect trust from past reputation and recommendations, considering multiple criteria such as QoS and Quality of Security (QoSec). While the system combines various factors to provide a nuanced trust assessment, the use of a trust decay function in highly dynamic environments may not update trust values in a timely manner, potentially affecting the accuracy of trust-based decisions.

Finally, Rahman et al. [46] introduced a trust management system that uses a fuzzy control system to evaluate the security, reputation, and availability of vehicular fog nodes (v-fogs). This system is particularly adaptable to dynamic environments, as it handles uncertainty and variability in trust evaluation processes. The use of a fuzzy control system enables the processing of multiple inputs and the evaluation of numerous rules to generate a final trust score. However, the computational overhead introduced by this method can be challenging in real-time applications, especially in large-scale networks where rapid decision-making is crucial.

Collectively, the above literature related to trust in fog computing from 2019 to 2024 illustrates various approaches to trust management in fog computing, each addressing specific challenges related to security, reliability, and performance. In Table III, the ongoing development of these mechanisms highlights the necessity of balancing robustness with efficiency, particularly as fog computing environments scale in size and complexity. Future research in this area is likely to focus on optimizing these trust management policies to ensure that they meet the needs of real-time applications while maintaining a high level of security and scalability in heterogeneous computing environments.

TABLE III COMPARATIVE ANALYSIS OF TRUST MANAGEMENT APPROACHES IN FOG COMPUTING LITERATURE (2019-2024)

Author	Method	Advantage	Weakness	How to Evaluate
Shahid Mahmood, et al. [26]	Based on digital certificate trust management	Enhances security and privacy of data communication between IoT devices through digital certificates.	Complex and costly deployment of numerous fog nodes as certification authorities. Cumbersome management and distribution of revoked certificates across devices.	Measure latency reductions. Assess security enhancements (e.g., fewer successful cyber-attacks). Evaluate resource utilization efficiency.
Rasha Jamal Atwa,et al. [20]	Task-based Experience Reputation (TER) : combines traditional Experience-based Reputation Models and Recommendation-based Reputation Models	Significantly reduces message transmission overhead and vehicle workload compared to experience-based models.	Traditional reputation computation methods struggle to adapt to the high mobility and transient nature of vehicles in VANETs.	Performance evaluated through MATLAB simulations. Comparison against experience-based trust models. Assessment of message overhead reduction and accuracy of trust assessments via task-specific reputation values.

Junbin Liang,et al. [21]	Multi-source feedback trust computation, combining both direct and recommended trust methods	Improves detection of malicious feedback nodes, enhancing the overall trustworthiness of the SSC.	Complicated implementation due to the integration of multiple feedback sources and the need for robust infrastructure.	Performance evaluated through theoretical analysis and simulation. Simulation results demonstrate higher trust accuracy with the multisource feedback mechanism.
Ogundoyin, et al. [34]	Based on Fuzzy-AHP	Enables comprehensive evaluation of trust by simultaneously considering multiple criteria, resulting in more accurate and reliable trust assessments.	Complex implementation due to the need for expert opinions and mathematical computations for prioritizing criteria. Subjectivity in expert judgment can lead to biased results, especially with inconsistent or inaccurate opinions.	Utilizes Fuzzy-AHP to prioritize trust criteria and calculate their weights. Assesses the impact on overall trust evaluations. Effectiveness demonstrated through nuanced and accurate trust assessments, enhancing reliability and security of fog computing services.
E. Alemneh,et al. [19]	Based on Subjective Logic: Direct Trust, Indirect Trust, Trust Tuple	Ensures trustworthiness of both service providers and requesters, enhancing overall security and reliability of the fog network.	Involves complex computational operations (e.g., discounting and consensus) that may burden resource-constrained fog nodes, particularly with large amounts of recommendation information.	Performance evaluated through simulations. Focuses on metrics such as accuracy, convergence, and resistance to trust-based attacks.
T. Wang, et al. [35]	Based on Hierarchical trust: Direct Trust, Recommendation Trust, Comprehensive Trust	Quickly identifies and isolates malicious nodes through real-time monitoring.	Involves multiple levels of trust computation and data analysis, leading to high computational complexity and potential overhead in large-scale networks.	Evaluated through simulations measuring performance metrics such as energy consumption, malicious node detection speed, and recovery of misjudged nodes.
A. Almas, et al. [18]	Based on Bayesian approach: Direct Trust, Indirect Trust, Total Trust	Dynamically adjusts trust weights to respond to environmental changes, effectively mitigating trust-related attacks.	Timeliness of trust value updates may be insufficient in highly dynamic network environments, potentially leading to misjudgement.	Evaluated through simulations using Contiki-NG and Cooja in a smart healthcare scenario. Analysis includes the impact of different similarity measures and comparisons between static and adaptive weighting methods on trust scores.
S. Mahmood, et al. [33]	Based on multi-source trust evaluation: Direct Trust, Recommendation Trust, Context-Aware Feedback, Content-Based Trust	Enhances reliability and accuracy of trust assessments by incorporating multiple sources of trust data.	Involves multiple levels of trust computation and data analysis, leading to high computational complexity and potential overhead in large-scale networks.	Evaluated through extensive simulations demonstrating effectiveness and reliability in assessing user trustworthiness. Analysis includes the impact of weight factors and the significance of monitor mode.
A. A. Bukhari, et al. [22]	Based on Fuzzy Logic: Fuzzification, Rule Setup, Fuzzy Inference, Defuzzification	The FL-based FNSE approach demonstrated superior performance with the highest accuracy, precision, recall, and F1 score in predicting fog node trustworthiness.	The fuzzy logic approach involves multiple steps (fuzzification, rule inference, and defuzzification), leading to high computational complexity that may impact real-time system performance.	Conducted experiments comparing three models (FL, LR, DNN) based on metrics such as accuracy, precision, recall, F1 score, and execution time.
R. Yadav, G. Baranwal [42]	Based on Feedback Credibility Evaluation: Direct Trust, Indirect Trust, Feedback Credibility Evaluation, Final Trust Score	Filters out malicious feedback before trust evaluation, enhancing the accuracy of trust scores assigned to fog nodes.	In highly dynamic network environments, trust value computations may not reflect the latest state of nodes in a timely manner, leading to delayed decision-making.	Evaluated through simulations using synthetic data to measure effectiveness in filtering malicious feedback and maintaining accurate trust scores. Performance metrics included detection rate, false rate, and deviation of trust scores from an ideal baseline.
Ke Gu, et al. [17]	Based on Reputation Mechanism: Internal Reputation, Indirect Reputation, Overall Reputation Value	Effectively matches user tasks with fog nodes that meet reputation requirements, improving service reliability.	Involves multiple levels of reputation computation and data analysis, leading to high computational complexity, especially with many communities and nodes, which can cause overheads and delays.	Evaluated using simulations on the CloudSim platform. Focused on metrics such as reputation changes, resource allocation success rate, average completion delay of tasks, and average distance between users and service providers.
J. Al. Muhtadi, et al. [40]	Based on Subjective Logic: Direct trust, Recommended Trust, trust convergence	Effectively identifies and mitigates malicious nodes using subjective logic that incorporates uncertainty and indirect evidence.	Heavily relies on recommendation information, which can bias trust calculations in networks with a high number of malicious nodes.	Evaluated through simulations assessing accuracy, convergence, and resilience against various trust-based attacks.

N. Mazumdar, et al. [41]	Based on Direct Trust, Indirect Trust, Comprehensive trust evaluation	Minimizes service delays and ensures timely task completion by offloading tasks to trusted nodes.	Shamir's secret sharing scheme adds complexity and burden, particularly in high-load or emergency situations.	Evaluated through simulations focusing on average latency, service response rate, and task completion success.
R. Yadav and G. Baranwal, [36]	Based on multilevel trust: Trust Factor Calculation, Overall, Trust Evaluation	Enhances security in fog computing by evaluating trust through QoS and social relationships, minimizing risks from malicious nodes.	Involves multi-layered trust computation and data analysis, leading to high computational complexity and potential overheads in large-scale networks.	Considers historical interactions and reputation among fog nodes to assess trustworthiness.
P. Kochovski, et al. [37]	Based on blockchain: Smart Contracts, Trustless Smart Oracles, Markov	Supports real-time monitoring and trust updates for quick responses to changes in node behavior.	Use of blockchain and smart contracts increases computational complexity and latency, especially with numerous transactions.	Evaluated through simulations focusing on the ability to detect and handle malicious nodes while maintaining performance. Metrics included accuracy in detecting trust anomalies and system overhead.
G. Rathee, et al. [43]	Based on Tidal Trust Algorithm	Ensures only trusted nodes communicate, effectively reducing the risk of malicious activities.	Requires computation of multiple trust values and uses complex algorithms, leading to high computational complexity that may impact real-time performance.	Examines performance in scenarios with malicious versus trusted fog nodes, focusing on managing security for handoff IoT devices (HEU) and mobile HEU (MHEU).
W. Ben Daoud, et al. [44]	Based on TACRM framework	Monitors and controls access based on trust levels, reducing the risk of unauthorized access and attacks.	Continuous monitoring of user behaviour may raise privacy concerns.	Evaluated through simulations of various network scenarios to measure the impact on security breach incidents.
M. Hamza, et al. [45]	based on a Bayesian inference model: Direct Trust, Reputation Function, Degree Centrality, Service Score, Final Trust	Detects and mitigates common trust-related attacks like ballot stuffing and self-promotion.	Comprehensive trust evaluation and the use of Bayesian inference increase computational complexity, potentially affecting real-time performance in large-scale networks.	Measures the accuracy of trust assessments through modelling.
S. O. Ogundoyin and I. A. Kamil [16]	Based on fuzzy logic: Direct Trust, Indirect Trust, trust convergence	Combines QoS, QoSec, and social interactions for a nuanced trust assessment.	Trust decay function may not update values timely in highly dynamic environments, affecting decision accuracy.	Evaluated through simulations measuring the system's ability to mitigate trust-related attacks and the accuracy of trust assessments.
G. Zhang, et al. [38]	Based on Hierarchical Trust Evaluation: Direct Trust, Indirect Trust, Hierarchical Trust Evaluation	Efficiently detects hidden data attacks by analysing sensor data correlations and historical trust states.	Layered architecture may lead to untimely updates of trust values in dynamic networks, affecting decision-making accuracy.	Evaluated through MATLAB simulations testing detection accuracy of the FDS under various scenarios, including different malicious sensor ratios and group sizes.
A. Rehman, et al. [39]	Based on Fuzzy Logic: Direct Trust, Indirect Trust, Trust Encryption and Aggregation	Effectively detects and mitigates various IoT attacks, including on-off, good-mouthing, and bad-mouthing attacks through encryption and trust aggregation.	Involves complex trust calculations, particularly with fuzzy logic, leading to high computational complexity that may affect real-time responsiveness.	Evaluated through simulations testing FogTrust's effectiveness against various attacks.
Q. Zhang, et al. [47]	a semantic-based trust management system integrated: Semantic Analysis, Direct Trust, Indirect Trust	Enables faster and more accurate emergency responses through semantic analysis, reducing failure risks in critical situations.	Scaling the system may challenge trust evaluations and semantic data processing, impacting performance.	Fog nodes assess trustworthiness based on historical interactions, data freshness, and recommendations, filtering out untrustworthy content.
T. Wang, et al. [35]	based on both direct trust and indirect trust	Utilizes both direct and indirect trust to adapt to network changes, including node mobility and varying trust levels.	Growing network size may complicate trust value calculations and data management, impacting scalability.	Final trust value is computed by combining direct and indirect trust values using a weighted sum.
F. H. Rahman, et al. [46]	Based on a fuzzy control system: Security, Reputation, Availability	Fuzzy control system effectively handles uncertainty and variability in trust evaluation, adapting to dynamic environments.	Introduces computational overhead from processing multiple inputs and evaluating numerous rules, challenging real-time applications.	Trust value is calculated using a fuzzy control system that processes security, reputation, and availability metrics to determine vehicle cluster membership.

IX. Challenges and Future Prospects

As the landscape of fog computing continues to shift, new challenges emerge that must be addressed to ensure effective trust management and task offloading. The increasing complexity and dynamism of network environments require innovative solutions to enhance the reliability and security of these systems. This section outlines several key challenges and

research directions that researchers and practitioners will need to tackle in the pursuit of more robust trust management frameworks.

A. AI and Machine Learning Integration

Integrating AI and machine learning techniques can significantly enhance trust assessments by leveraging historical data and real-time interactions. Research can explore supervised and unsupervised learning algorithms to model node behavior and predict trustworthiness, allowing for more accurate and adaptive trust evaluations that evolve with the network.

B. Hybrid Trust Models

Hybrid models that combine various trust evaluation techniques can provide more resilient systems. Research can explore the integration of reputation systems, subjective logic, and Bayesian approaches to create comprehensive models that account for both direct experiences and indirect recommendations, enhancing the accuracy of trust assessments.

C. Context-Aware Trust Management

Context-aware mechanisms can adjust trust evaluations based on environmental factors and user behaviors. Future research can focus on developing systems that dynamically modify trust metrics according to contextual information, ensuring that trust assessments are relevant and responsive to the current state of the network.

D. Decentralized Trust Management Systems

Decentralized approaches reduce reliance on central authorities and enhance resilience. Research should explore the design of decentralized trust management frameworks that empower nodes to independently evaluate trust, using peer-to-peer communication protocols to share and validate trust information among nodes.

E. Cross Domain Trust Management

Managing trust across different domains (e.g., fog and cloud computing) is vital for seamless interactions. Future studies should investigate methods for establishing trust relationships that span multiple environments, ensuring that resources can be shared securely and efficiently without compromising the integrity of trust evaluations.

F. Performance Optimization Techniques

Optimizing computational processes involved in trust evaluations can minimize overhead. Research should focus on developing algorithms that streamline trust computations, utilizing techniques such as caching trust values, batching evaluations, or employing approximation methods to reduce computational complexity without sacrificing accuracy.

G. User Centric Trust Models

Developing user-centric trust management systems can enhance user experiences by prioritizing individual preferences and interactions. Research should explore personalized trust models that adapt to user behaviors and preferences, allowing for more relevant interactions and trust assessments that reflect user needs.

By addressing these challenges and pursuing the outlined research directions, the field of trust management in fog computing can evolve to meet the demands of increasingly complex and dynamic environments, ensuring both security and performance.

X. Conclusion

This study provides a comprehensive exploration of the critical role of trust management in fog computing, particularly in ensuring secure and efficient task offloading among diverse IoT devices. It highlights significant gaps in existing methodologies and emphasizes the urgent need for advanced trust assessment frameworks. By establishing key metrics for evaluating trust management systems, we lay the groundwork for measuring their effectiveness, which is essential for fostering reliable interactions in dynamic environments. The integration of AI and machine learning techniques is examined as a promising approach to enhancing trust evaluations. These technologies allow systems to adapt to real-time data and evolving network conditions. However, several challenges persist, including scalability, the management of diverse IoT devices, the achievement of real-time trust evaluations, and the need to address security and privacy concerns. Moreover, addressing these challenges through innovative research and practical implementations is crucial for advancing fog computing technologies. This work aims to provide valuable insights that will facilitate the development of more secure and efficient applications, ensuring that trust management evolves alongside the rapidly changing landscape of IoT and fog computing.

References

[1] Y. Meng, M. A. Naeem, A. O. Almagrabi, R. Ali, and H. S. Kim, "Advancing the state of the fog computing to enable 5g network technologies," Sensors, vol. 20, no. 6, p. 1754, 2020.

[2] M. A. Al-Shareeda, A. A. Alsadhan, H. H. Qasim, and S. Manickam, "The fog computing for internet of things: review, characteristics and challenges, and open issues," Bulletin of Electrical Engineering and Informatics, vol. 13, no. 2, pp. 1080-1089, 2024.

[3] R. Das and M. M. Inuwa, "A review on fog computing: Issues, characteristics, challenges, and potential applications," Telematics and Informatics Reports, vol. 10, p. 100049, 2023.

[4] M. A. Al-Tarawneh, "Bi-objective optimization of application placement in fog computing environments," Journal of Ambient Intelligence and Humanized Computing, vol. 13, no. 1, pp. 445-468, 2022.

[5] C. Arivazhagan and V. Natarajan, "A Survey on Fog computing paradigms, Challenges and Opportunities in IoT," in 2020 international conference on communication and signal processing (ICCSP), 2020, pp. 0385-0389: IEEE.

[6] G. K. Walia, M. Kumar, and S. S. Gill, "AI-empowered fog/edge resource management for IoT applications: A comprehensive review, research challenges, and future perspectives," IEEE Communications Surveys & Tutorials, vol. 26, no. 1, pp. 619-669, 2023.

[7] M. Muneeb, K.-M. Ko, and Y.-H. Park, "A fog computing architecture with multi-layer for computing-intensive iot applications," Applied Sciences, vol. 11, no. 24, p. 11585, 2021.

[8] Hassan, S., & Alsamman, M. (2021, January). Energy-aware multipath forwarding mechanism for named data network in wireless environment. In Proc. 6th Int. Conf. Internet Appl. Protoc. Serv. Endur. Internet's Use Growth Pandemic Era (pp. 43-49).

[9] K. M. Sadique, R. Rahmani, and P. Johannesson, "Fog computing for trust in the Internet of Things (IoT): A systematic literature review," in 2020 International Conference on Computer Science, Engineering and Applications (ICCSEA), 2020, pp. 1-6: IEEE.

[10] S. U. Khan, "The curious case of distributed systems and continuous computing," IT Professional, vol. 18, no. 2, pp. 4-7, 2016.

[11] A. Ometov, O. L. Molua, M. Komarov, and J. Nurmi, "A survey of security in cloud, edge, and fog computing," Sensors, vol. 22, no. 3, p. 927, 2022.

[12] P. Hu, S. Dhelim, H. Ning, and T. Qiu, "Survey on fog computing: architecture, key technologies, applications and open issues," Journal of network and computer applications, vol. 98, pp. 27-42, 2017.

[13] M. Alsamman, Y. Fazea, F. Mohammed and M. A. M. Kehail, "Fog Computing in Smart Cities: A Systematic Review," 2023 3rd International Conference on Emerging Smart Technologies and Applications (eSmarTA), Taiz, Yemen, 2023, pp. 1-8, doi: 10.1109/eSmarTA59349.2023.10293505.

[14] Z. Ashi, M. Al-Fawa'reh, and M. Al-Fayoumi, "Fog computing: security challenges and countermeasures," Int. J. Comput. Appl, vol. 175, no. 15, pp. 30-36, 2020.

[15] S. Parikh, D. Dave, R. Patel, and N. Doshi, "Security and privacy issues in cloud, fog and edge computing," Procedia Computer Science, vol. 160, pp. 734-739, 2019.

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

[17] K. Gu, L. Tang, J. Jiang, and W. Jia, "Resource allocation scheme for community-based fog computing based on reputation mechanism," IEEE Transactions on Computational Social Systems, vol. 7, no. 5, pp. 1246-1263, 2020.

[18] A. Almas, W. Iqbal, A. Altaf, K. Saleem, S. Mussiraliyeva, and M. W. Iqbal, "Context-based adaptive Fog computing trust solution for time-critical smart healthcare systems," IEEE Internet of Things Journal, vol. 10, no. 12, pp. 10575-10586, 2023.

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

[20] R. J. Atwa, P. Flocchini, and A. Nayak, "A fog-based reputation evaluation model for VANETs," in 2021 International Symposium on Networks, Computers and Communications (ISNCC), 2021, pp. 1-7: IEEE.

[21] J. Liang, M. Zhang, and V. C. Leung, "A reliable trust computing mechanism based on multisource feedback and fog computing in social sensor cloud," IEEE Internet of Things Journal, vol. 7, no. 6, pp. 5481-5490, 2020.

[22] A. A. Bukhari and F. K. Hussain, "Fuzzy logic trust-based fog node selection," Internet of Things, vol. 27, p. 101293, 2024.

[23] V. Meena, M. Gorripatti, and T. Suriya Praba, "Trust enforced computational offloading for health care applications in fog computing," Wireless Personal Communications, vol. 119, no. 2, pp. 1369-1386, 2021.

[24] B. Premalatha and P. Prakasam, "TwI-FTM: Two-way IoT-FoG trust management scheme for task offloading in IoT-FoG networks," Results in Engineering, vol. 22, p. 102197, 2024.

[25] S. S. Manvi and N. C. Gowda, "Trust management in fog computing: a survey," in Applying integration techniques and methods in distributed systems and technologies: IGI global, 2019, pp. 34-48.

[26] A. Shakarami, A. Shahidinejad, and M. Ghobaei-Arani, "A review on the computation offloading approaches in mobile edge computing: A game-theoretic perspective," Software: Practice and Experience, vol. 50, no. 9, pp. 1719-1759, 2020.

[27] M. Yu, A. Liu, N. N. Xiong, and T. Wang, "An intelligent game-based offloading scheme for maximizing benefits of IoT-edge-cloud ecosystems," IEEE Internet of Things Journal, vol. 9, no. 8, pp. 5600-5616, 2020.

[28] W. Mo, T. Wang, S. Zhang, and J. Zhang, "An active and verifiable trust evaluation approach for edge computing," J. Cloud Comput., vol. 9, no. 1, 2020, doi: 10.1186/s13677-020-00202-w.

[29] J. Kennedy and R. Eberhart, "Particle swarm optimization," in Proceedings of ICNN'95-international conference on neural networks, 1995, vol. 4, pp. 1942-1948: ieee.

[30] T. M. Shami, A. A. El-Saleh, M. Alswaitti, Q. Al-Tashi, M. A. Summakieh, and S. Mirjalili, "Particle swarm optimization: A comprehensive survey," Ieee Access, vol. 10, pp. 10031-10061, 2022.

[31] Prado-Rodríguez, R., González, P., Banga, J. R., & Doallo, R. (2024). Improved cooperative Ant Colony Optimization for the solution of binary combinatorial optimization applications. Expert Systems, 41(8), e13554.

[32] Liu, F., & Guo, W. (2019). Optimized Min-Min Dynamic Task Scheduling Algorithm in Grid Computing. Advances in Intelligent Systems and Computing. https://doi.org/10.1007/978-3-030-25128-4_92.

[33] S. Mahmood, A. Ullah, and A. K. Kayani, "Fog computing trust based architecture for internet of things devices," International Journal of Computing and Communication Networks, vol. 1, no. 1, pp. 18-25, 2019.

[34] S. O. Ogundoyin and I. A. Kamil, "A Fuzzy-AHP based prioritization of trust criteria in fog computing services," Applied Soft Computing, vol. 97, p. 106789, 2020.

[35] T. Wang, G. Zhang, M. Z. A. Bhuiyan, A. Liu, W. Jia, and M. Xie, "A novel trust mechanism based on fog computing in sensor–cloud system," Future Generation Computer Systems, vol. 109, pp. 573-582, 2020.

[36] R. Yadav and G. Baranwal, "An efficient trust management using feedback credibility evaluation method in fog computing," Simulation Modelling Practice and Theory, vol. 120, p. 102610, 2022.

[37] P. Kochovski, S. Gec, V. Stankovski, M. Bajec, and P. D. Drobintsev, "Trust management in a blockchain based fog computing platform with trustless smart oracles," Future Generation Computer Systems, vol. 101, pp. 747-759, 2019.

[38] T. Li, G. Huang, S. Zhang, and Z. Zeng, "NTSC: a novel trust-based service computing scheme in social internet of things," Peer-to-Peer Networking and Applications, vol. 14, pp. 3431-3451, 2021.

[39] A. Rehman, K. A. Awan, I. Ud Din, A. Almogren, and M. Alabdulkareem, "FogTrust: fog-integrated multi-leveled trust management mechanism for internet of things," Technologies, vol. 11, no. 1, p. 27, 2023.

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

[41] N. Mazumdar, A. Nag, and J. P. Singh, "Trust-based load-offloading protocol to reduce service delays in fog-computing-empowered IoT," Computers & Electrical Engineering, vol. 93, p. 107223, 2021.

[42] R. Yadav and G. Baranwal, "Trust-aware framework for application placement in fog computing," in 2019 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS), 2019, pp. 1-6: IEEE.

[43] G. Rathee, R. Sandhu, H. Saini, M. Sivaram, and V. Dhasarathan, "A trust computed framework for IoT devices and fog computing environment," Wireless Networks, vol. 26, pp. 2339-2351, 2020.

[44] W. B. Daoud, M. S. Obaidat, A. Meddeb-Makhlouf, F. Zarai, and K.-F. Hsiao, "TACRM: trust access control and resource management mechanism in fog computing," Human-centric Computing and Information Sciences, vol. 9, pp. 1-18, 2019.

[45] M. Hamza, W. Iqbal, A. Ahmad, M. Babar, and S. Khan, "A social qualitative trust framework for Fog computing," Computers and Electrical Engineering, vol. 102, p. 108195, 2022.

[46] F. H. Rahman, S. S. Newaz, T. W. Au, W. S. Suhaili, and G. M. Lee, "Off-street vehicular fog for catering applications in 5G/B5G: A trust-based task mapping solution and open research issues," IEEE Access, vol. 8, pp. 117218-117235, 2020.

[47] Q. Zhang, J. Wu, M. Zanella, W. Yang, A. K. Bashir, and W. Fornaciari, "Sema-IIoVT: Emergent semantic-based trustworthy information-centric fog system and testbed for intelligent internet of vehicles," IEEE Consumer Electronics Magazine, vol. 12, no. 1, pp. 70-79, 2021.