TTLA: Two-way Trust Between Clients and Fog Servers Using Bayesian Learning Automata

Niloofar Barati Bakhtiari
Department of Computer Engineering, Qom Branch,Islamic azad University, Qom
Masood Rafighi (Rafighi@mut.ac.ir)
Faculty of Electrical & Computer Engineering, Malek Ashtar University of Technology, Tehran
Reza Ahsan
Department of Computer Engineering, Qom Branch,Islamic azad University, Qom

Research Article

Keywords: Fog computing, Trust management, Bayesian Learning Automata

Posted Date: July 6th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1744138/v1

Noname manuscript No.
(will be inserted by the editor)

TTLA: Two-way Trust Between Clients and FogServers Using Bayesian Learning Automata

Niloofar Barati Bakhtiari · Masood Rafighi · Reza Ahsan

Received: date / Accepted: date

Abstract Fog computing (FC) is a promising paradigm to use as an efficient architecture for the Internet of Things applications. Proximity, low latency, flexible resource power, and distributed structure of this architecture are some benefits of it. A huge number of generated data and their requisites to real-time process causes fog nodes offload number of tasks to the others that make trust issues. Here, each clients prefers to offload task to a trusted server, also each server tends to service the trusted clients. This may takes a long especially when we want to consume less energy. In order to encounter this problem,in this paper, we propose a two-way trust management strategy based on Bayesian learning automata. The proposed approach outperforms the other state-of-the-art approaches in terms of the energy consumption, network usage, latency, response time, and trust value.

Keywords Fog computing · Trust management · Bayesian LearningAutomata

1 Introduction

Over the past decade, the ubiquity of the Internet and the growing numberof smart physical devices connected to the Internet has created a new con- cept called the Internet of Things (IoT). IoT devices such as sensors, objects, actuators, and intelligent nodes are associated with mobility challenges, re- source constraints, scalability, and heterogeneity [1]. IoT devices, on the other hand, generate large volumes of data from big data. For example, an electronic

N. Barati and R. Ahsan
Department of Computer Engineering, Qom Branch, Islamic Azad University, Qom, IranE-mail: nil.barati@gmail.com; ahsan.ac.ir@gmail.com

M. Rafighi (✉)
Faculty of Electrical & Computer Engineering, Malek Ashtar University of Technology,Tehran, Iran E-mail: Rafighi@mut.ac.ir

patient health tracking system includes various wearable devices and sensors that generate a multitude of different types of data [2]. Cloud computing is an effective way to process and store this huge amount of data due to its high computing power and storage capability. Some IoT applications may require very short response times and fast handling, such as intelligent transportation systems [3], smart grids [4], smart healthcare [5–9], emergency responses and other delay-sensitive programs. On the other hand, some decisions can be made locally without having to transfer data to the cloud. Even if some decisions need to be made in the cloud, it is not necessary to send all the data to cloud and categorize it. Because not all data will be useful for analysis and decision making. In other words, these challenges, which are driven by the rapid growth of the IoT and are associated with network bandwidth, latency, reliability, and security, cannot be addressed independently of a centralized cloud model. Therefore, cloud computing is not suitable for real-time analysis and decision making [2]. FC is a distributed paradigm that can overcome these challenges and limitations by computing, storage, communication, and network services at the edge of the network and between end devices and cloud data centers to reduce latency, extends bandwidth and reliability [10]. Therefore, it can handle instantaneous data and there is no need to transfer all data from edge devices to the cloud immediately [11]. On the other hand, by integrating edge equipment and cloud resources, it also prevents resource challenges. FC, which actually complements it instead of cloud computing, supports mobility, dynamism, geographic distribution, location awareness, heterogeneity, and interoperability [2]. FC helps cloud computing perform its role more efficiently. Considering the features of cloud computing, traffic safety [12, 13], electronic health [5], web content delivery [14], augmented reality [15], virtual reality [16] and big data analysis [17], including new IoT applications and servicesare suitable for cloud computing. Due to the characteristics of fog such as het- erogeneous environment, flexibility, dynamics and geographical distribution, fog can be in a vulnerable and uncertain situation. On the other hand, fog can be moving and therefore not available. Therefore, at any time, the unforeseen conditions of the fog may be endangered. For example, in a foggy environment,if a moving car is used as a service car, the car may be involved in an accident, or, for example, drones performing edge services may have a technical defect. Therefore, it will be difficult to predict performance indicators such as power, availability, and reliability. Because this is more complex and challenging than the cloud, it is difficult to predict and trust fog [18], and to resist security and privacy threats with the features and flexibility of the fog model [2, 11, 19].

The security and privacy solutions available in cloud computing cannot be used directly for fog computing because the architecture of the two computational podiums is quite different [19, 20]. Because of the centralized components of cloud computing, cloud security is relatively simple compared to distributed architectures such as cloud computing. Therefore, to accelerate the development and adoption of cloud computing in academia and industry, security issues, privacy, and trust in cloud computing are essential issues that need to be addressed. One of the challenges associated with the cloud computing environment, which is highly related to security and privacy, is trust management. Lack of redundancy, dynamism, high mobility support, and low node processing power [21] are other features of FC that add to the complexity of trust management in FC. For these reasons, fog servers are potential threatsto fog clients as well as other fog servers. The same is true for fog customers. Although authentication is a very useful encryption method to start the ini- tial connection between the Internet of Things and fog nodes, it is not enough because over time the devices can malfunction and may be compromised by attackers. In addition, existing cryptographic solutions can not counteract internal attacks such as the attack of a rogue node that has already been authenticated and has become part of the system [1]. Trust plays an important role in the relationship between fog nodes and end devices. Fog nodes are considered to be the most important component of the IoT-based network [22],as it is responsible for ensuring the privacy and anonymity of end users [7].In addition, this component must be trusted to gain agency responsibility, asit must ensure that the node implements the encryption process on the datait publishes, and only launches non-destructive activities. This requires thatall nodes that are part of the fog network have a certain level of trust in each other [11].

Trust management provides the mechanism for deciding whether to trust the entity to be associated. Trust in nodes allows them to predict the behavior of other nodes, and this can help the decision-making process. It also leadsto the diagnosis of damaged or misbehaving nodes [23, 24]. Trust is essentialfor interacting in an uncertain environment and ensures information security and user privacy. Devices may encounter other heterogeneous devices in the network and should be handled with caution as there is uncertainty and instability in their behavior. The purpose of trust management systems in cloud computing is to identify and prevent invalid servers and servers. In addition to establishing trust, it should be noted that the trust management mechanism should not lead to a negative impact on other network parameters, including increasing energy consumption. Therefore, fog servers should be encouraged to reduce energy consumption in order to increase their chances of being selectedto serve fog customers by increasing reliability and reducing energy consumption. In the business world, most organizations are concerned not only with the environment and reducing environmental impacts, but also with their financial interests. By making better use of energy resources, energy costs of organizations are also saved and the desire of managers of organizations to the FC paradigm increases. To reduce energy consumption, various algorithms and structures for cloud computing have been proposed, each of which seeks to reduce energy consumption and better use of resources. Energy consumption is being studied in various fields, for example in the fields of processing, storage and transmission. However, considering the characteristics of the fog network, the most important feature of which is the reduction of latency due to the use of critical and delay-sensitive applications of this network, it is necessary to look for a suitable method for the fog network that reduces latency and security is not compromised [25]. In this research, while presenting the model of mutual trust management between customers and fog servers, we are looking for an intelligent and secure routing that can select the most reliable fog server that also has low energy consumption. There are many routing algorithms in the network, but due to the special conditions of the fog network, the nature of which is to serve real-time and latency-sensitive applications, we should use a light and efficient routing algorithm that tries to select secure nodes with lower power consumption. Help to increase energy efficiency and extend the life of the fog grid as much as possible.

1.1 Motivation of BLA

In the trust management system proposed by this research, the customer sendshis request to the most trusted server, which can be even a server with the lowest energy consumption. In this system, Bayesian learning automaton will be used to learn all the modes and actions in the network in order to achieve the best and most reliable node (and the lowest energy consumption). The selection of BLA has been done by studying the following routing algorithms: heuristic (or innovative / Exploratory) algorithms: Heurists seek a reasonable and acceptable answer to a difficult problem that is not necessarily the best answer to that problem and have no guarantee of an answer, like the greedy algorithm. Meta-heuristic algorithms: algorithms that are used to solve op- timization problems and in cases where they are combined or some of its steps are modified, have the ability to escape the local optimization and reach the global optimization, such as genetic algorithm, ant colony, bee colony, refrigeration simulation algorithm, forbidden search algorithm, and colonial competition [26]. Machine learning algorithms: These algorithms can process large amounts of data and predict or make patterns based on this information.Over time, as the program modifies itself, the predictive model becomes more accurate, such as deep learning, neural network, and reinforcement learning [27].

Due to the nature and features of FC, which has been placed next to the cloud as a supplement to help speed up and reduce delays in critical applications such as intelligent transportation, smart healthcare, emergency response, and other delay-sensitive applications. In the proposed trust management system, a method will be provided that, while increasing the accuracy in calculating trust compared to other trust management systems, has lightweight calculations to increase the speed of delay problem, which is the most important issue for delay-sensitive applications. Since the use of probability in learning algorithms can increase the speed of the algorithm, so among the deep learning algorithms that have the highest accuracy and the learning automata algorithm which has simple and lightweight calculations, because deep learning algorithms and deep learning algorithms. The need for computing power, battery, and high memory is due to the complexity of the algorithm [28] and these requirements are among the limitations of IoT devices, so a combination of learning automata and Bayesian inference based on probability will be used to achieve the best answer in the shortest time.

Our key contributions in this paper are as follows:

1. We provide an intelligent two-way trust management system based on subjective logic using the Bayesian learning automata algorithm. This system allows the clients to verify whether the server can provide a reliable and secure service, and on the other hand, allows the server to check the reliability of clients.
2. System evaluation shows that the proposed method using a lightweight learning algorithm is smart in selecting service providers and is more complete than two-way trust management system in fog systems [2] and fuzzy trust management system [29] with better efficiency in energy consumption, latency, cost, trust and network usage.

The rest of this paper is organized as follows. In Section 2, related works are summarized. The system model and network architecture are presentedin Section 3. In Section. 4, we explain our intelligent trust management algorithms in detail. In Section 5, the evaluation results of the proposed algorithms present and compare with other methods. Finally, in Section 6 the conclusion is discussed and suggestions are made for future work.

2 Related work

In recent years, some researches have been done about trust management in the cloud, fog, and IoT systems. We surveyed these works in the scope oftrust management. In 2020, Rati et al. [30] proposed a routed secure sending mechanism to prevent attack by examining the amount of trust and ranking of each IoT device and fog nodes based on their communication behavior. A trust manager is created between the fog and the IoT layers, which keeps a record of all fog nodes in the search table and identifies IoT nodes and malicious fog nodes. In addition, the fog nodes calculate the requested services layer of the IoT layer and direct the services in the most secure way possible. In this method, a third party is used as the trust manager between the fog and the IoT layers. Trust-based techniques increase security compared to cryptographic techniques without reducing communication overhead and increasing network standards. In 2020, El-Khafaji et al. [31] proposed a trust management approach for cloud computing (COMITMENT) based on a trust recommendation according to quality of service (QoS) and quality of the previous history (QoP) criteria derived from direct interactions and experiences. Previous indirect nodes are used to evaluate and manage the level of trustof nodes in the FC environment. Nodes with a positive track record have a positive effect on the overall trust score and nodes with a negative track record have a negative impact on the overall trust score. In this model, the Bayesian network is used to evaluate direct satisfactory experiences based on direct interactions between fog nodes. In 2020, Yaserhosin et al. [32] proposed a text-aware trust assessment model to assess user reliability in a fog-based IoT (FIoT) environment to identify malicious nodes. The proposed approach uses a multi-resource-aware evaluation system of text-based trust and reputation, which helps to assess user reliability. Text-aware feedback and feedback detector systems have also been used to establish an unbiased trust assessment. Also, the monitoring mode of unreliable and destructive users are considered to monitor the behavior and trust of users.

In 2020, El Amena et al. [2] introduced a two-way trust management system based on peer-to-peer mental logic in an FC. The operation of the system is such that the fog user or the service requester verifies the validity of the fog servers, whether they can provide correct, reliable, and secure services, and conversely, the fog servers before providing the service, the legitimacy, and illegitimacy of the fog users and In fact, the service requester reviews and approves. This system is distributed and event-based trust management and considers the criteria of service quality and social trust to determine trust. Also, the final trust value is calculated by dynamically combining informationobtained from direct and indirect observation (recommendations of neighbor- ing nodes). In 2020, Avan et al. [33] proposed a NeuroTrust trust manage- ment mechanism for the IoT that uses a superimposed learning multilevel perceptron neural network to predict malicious and compromised nodes for safe transmission. An input layer, two hidden layers, and an output layer witha threshold value are used to calculate the binary output. IoT devices include patient health sensors that are connected to patients and can transmit data to health care providers such as hospitals. The proposed method uses trust parameters including reliability, compatibility, and package delivery rate to assess the degree of trust. This method uses a light encryption mechanism for security when publishing data. In the proposed mechanism of NeuroTrust, the destructive behavior of nodes is predicted and data transfer is done only if the receiver is secure and can identify the source node using trust parameters. This method uses a smart home equipped with a dedicated server to perform trust calculations and monitor malicious nodes. Trust is calculated directly and indirectly (recommendations).

In 2020, El Sayed et al. [34] proposed a trust-based framework based on machine learning using decision tree classifier and artificial neural networks for the vehicle network. The network nodes are multifaceted and typically consist of vehicles, roadside units (RSUs), and data centers that use wireless networks to exchange data. To monitor and control large volumes of vehicles, information is exchanged between nodes, and security in this critical network and trust between nodes when exchanging messages is a key element of security. In this model, distance-based criteria such as Euclidean distance are used to assess trust. The proposed trust-based model uses a direct and indi- rect trust assessment strategy (recommendations) to calculate the amount of trust. Maps are assigned to vehicles by RSU based on the behavior of nodesin the vehicle environment, and neural network concepts are implemented in RSU. Nodes with a minimum distance from RSU and a good track record of trust values are selected by RSU as recommended nodes. One of the factors will act as a controlling factor that helps store various computational results and information about the nodes of the vehicle and the RSU.

Farahbakhsh et al. in 2020 [35] used the Bayesian Learning Automata (BLA) to offload context-aware load in Mobile Edge Computing (MEC) with multiple users. The Bayesian automaton learns all the modes and actions in the network and helps to improve the offloading algorithm. Contexts are collected using independent management as a loop of monitoring, analysis, planningand execution in all offload processes. The simulation results show that this method is superior in some criteria such as energy consumption, implementa- tion cost, network usage and delay of local calculations and offloading without considering context-aware algorithms.

Wang et al. in 2020 [36] proposed a framework for collecting data from the WSN to the cloud using moving fog elements that use the remaining distance and energy factors to form a routing map and reduce energy consumption and latency rates. Simulation-based experiments have shown that the proposed framework improves routing performance compared to traditional solutions. However, the proposed framework does not address network threats that maybe dangerous to data transmission and their effects on data privacy and in- tegrity. In addition, routing maps are selected without in-depth analysis of uncontrolled links, which increases response time and communication costs.

In 2020, Elias et al. [37] proposed a three-layer cluster-based wireless sensor network routing protocol to achieve longer network life in terms of energy. They offered this plan with the aim of increasing network life, improving throughput, reducing latency or packet loss, and continuing to work in the face of malicious nodes. The proposed mechanism is a three-layer clustering method with a built-in security mechanism to deal with malicious activity of sensor nodes and blacklist them. It is a center-based clustering protocol, in which the headerand network head are selected by the sink node based on the value of its cost function. In addition, hardware-based link quality estimation is used to evaluate link effectiveness and further improve routing performance. Experimental results show that the performance of this method is based on many of its counterpart protocols such as fuzzy logic based on unequal clustering, ant colony optimization based on hybrid routing, artificial bee colony, three-layer hybrid clustering mechanism and energy-conscious multi-hop routing. The proposed approach is superior to the others in network lifetime, throughput, average power consumption, and packet latency.

In 2020, Sabramanian et al. [38] proposed a lightweight trust mechanism based on the recommendation of neighboring nodes to establish packet routing protocols (PRL) in low-power, high-bandwidth networks. The RPL is a protocol used for routing in IoT networks with limited nodes and high packet loss rates and high-loss links. The RPL provides security features for communications but is often susceptible to routing attacks. In this research, based on the behavior of entities inside the system, the future behavior of nodes is predicted and the level of trust among fog nodes is increased while it does not affect the speed and performance of the network. This algorithm uses the add-on reduction coefficient (AIMD) approach for secure communication and has improved

Table 1: Summary of trusted management researches. (Veh.: Vehicles, Pla: Platform).

Pla.	Properties	Limitations	Trust criteria
Fog [2]	Using subjective logic in assessing trust / Calculating direct and indirect trust / Mutual trust management system between customers and fog servers / Using QoS and social relationships in trust assessment / Accurate calculation of trust and achieving convergence in a very small number of computational cycles / Flexibility against a large number of misbehaving nodes / thwarting trust-based attacks	Negative impact on load distribution due to more customer attraction for servers that have more trust	QoS and Community Relations
Veh. [34]	Decision tree-based trust management and neural network for vehicle network / Calculation of direct and indirect trust / Better detection rate of malicious nodes than compared methods	Use of centralized RSU unitsand control center to store computational results / Ignoring trust-based attacks	QoS
Fog [30]	Secure routing and mobility mechanism to prevent attacks / Build trust manager between fog layer and IoT layer / Ranking and trust of IoT devices with respect to node communication behavior / Focus on identifying malicious nodes and blocking them in future communications / 85% detection accuracy Malicious nodes and removing them as soon as they are detectedUse	Ignoring QoS criteria in trust assessment / Considering limited parameters in trust assessment / Failure to review trust-based attacks	Community Relations
IoT [33]	of dedicated server to perform trust calculations and monitor malicious nodes / Use of multilayer perceptron neural network to assess trust / Use of parameters of reliability, compatibility and package delivery rate to assess trust / Calculate direct and indirect trust / Resistance Equalto trust-based attacks	Use of limited QoS criteria in assessing trust / non-scalability	QoS and Community Relations
Fog [31]	Use of QoS criterion in trust assessment / Calculation of direct and indirect trust / Use of Bayesian network model to assess direct trust / Use of recommendations to assess indirect trust / Identification of malicious attacks up to 66%	Use only the QoS criterion in the assessment of trust / ignore the criteria of social relations between nodes in the assessment of trust	QoS
Fog [32]	Provide a text-based reputation and trust model to identify malicious / indirect trust nodes using user experience records with other users and devices and interactions	Ignoring QoS criteria in trust assessment / not addressing trust parameters / not checking trust-based attacks	Community Relations

the AIMD approach by calculating the trust value based on the recommendations of neighboring nodes. Hasab et al. in 2021 [39] proposed a lightweight and secure fog-based routing protocol (SEFR) to minimize data latency and increase energy management. This method uses multidimensional service quality criteria to select the next hop and facilitates time-sensitive applications with network edges. The proposed protocol also protects real-time data based on two levels of cryptographic security. In the first level, a lightweight confidential scheme for data between cluster heads and fog nodes is proposed, and in the second level, a high-performance asymmetric cryptographic scheme between fog and cloud layers is proposed. Analysis of simulation-based experiments has proven significant results of the proposed protocol compared to existing solutions in terms of routing, security and network management.

The authors in [29] offered a broker-based trust assessment framework for fog service allocation, which focuses on identifying a reliable fog to fulfill user requests. They used fuzzy logic as the basis for the evaluation and designed a fuzzy-based filtering algorithm to match the user's request to one of the predefined sets created and managed by the server. In this plan, only QoS trust criteria are considered and it is one-sided trust.

In this research, we try to model and solve the trust management systemin the Fog. The summary of trusted management researches is categorized by platforms, properties, limitations, and trust criteria in Table 1.

3 System model and architecture

In this section, we present the architecture and system model. The architectureof FC environment including fog clients, fog servers, and device owners is as Fig. 1. In this structure, all fog nodes can offload their task to other trusted devices. The problem is to find the most trusted fog nodes as a destination of offloading.

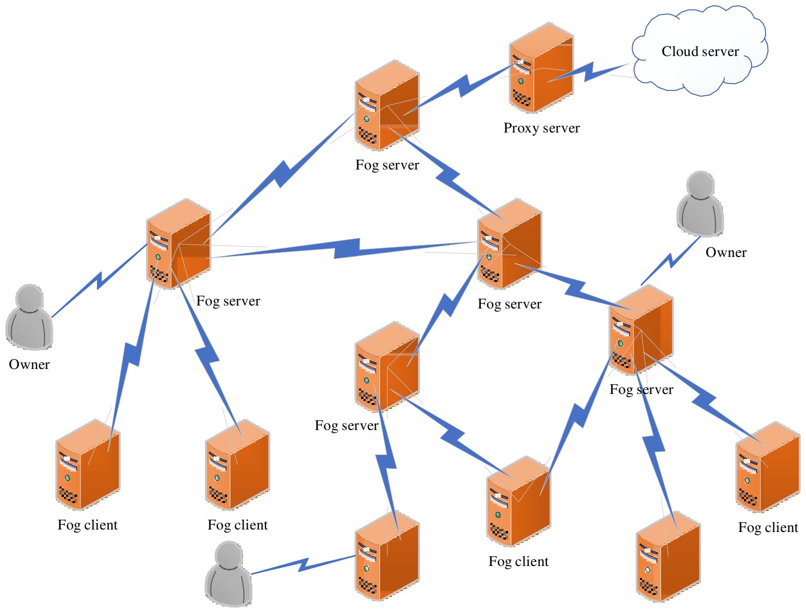

Fig. 1: System architecture including fog clients, fog servers, and device owners

Fig. 2 shows the system model of our proposed system. The fog server can communicate with neighboring fog nodes (i.e. with fog servers and fog clients) for up to one step (1-hop). The customer may be user-portable devices suchas smartphones, laptops and computers, or non-user devices such as smart lamps, smart washing machines, CCTV cameras, and so on. In this research, we mean the first group of devices, i.e. portable devices by the user. Fog clients are moving in a predetermined direction and can communicate with neighboring fog servers. Each node has an owner and an individual may own more than one node.

The fog client, which intends to receive the service, requests a fog server to connect. Using the BLA algorithm, the environment learns to introduce the most trusted and best fog server to the customer. The fog server then wantsto make sure it is connected to a trusted (not malicious) client. Therefore, to determine the legitimacy of the customer, the server, in consultation with neighboring servers and directly observing itself, calculates the amount of trust in the customer. The future is saved. This value is also sent to other servers as a recommendation. The same is true for the customer. The customer, in turn, wants to make sure that the fog server is reliable and can provide the right service. A malicious fog server may provide the wrong service. Fog client consults with neighboring fog servers and the result is combined with direct observation to determine the final trust status of the server. Similar to fog servers, fog clients share their server experience with other nearby servers. Nodes only store and exchange trust information about neighboring nodes for computational performance. In summary, negotiation and interaction are as follows. Using BLA, the most trusted server is connected to the client. The server then evaluates the client's trust, and if a trust threshold is established, the connection between the server and the client is complete, otherwise the server rejects the connection.

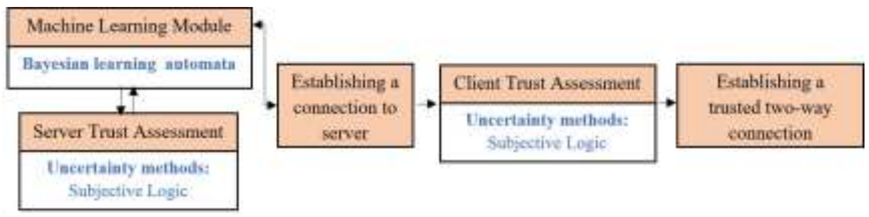

Fig. 2: Relation between machine learning module and trust assessment in mutual trust management system

Due to the characteristics of fog such as heterogeneous environment, dynamism, high mobility support, geographical distribution, lack of redundancy and wireless connectivity in fog environment, security problems in fog are very vital and therefore security as one of the main components of architecture. Fog calculations are considered. Apart from trust management problems, all security issues such as encryption, privacy, authentication, intrusion detection and prevention, etc. are controlled by this component. Trust management systems enable reliable communication between nodes in the fog environment, and trust management algorithms can be used to identify trusted clients and fog servers. The proposed trust management algorithm can be run under applications that require reliable communication. Once trusted nodes are found, fog clients can send their data to servers in secure conditions. Servers also retain clients that are recognized as trusted by a trust management algorithm. Hence, the trust management system helps the nodes to check each other's reliability before the actual service is provided. The symbols used in this paper are defined in Table 2.

Table 2: Symbol definition

Symbol	Definition
$C_1, C_2, \dots, C_n$	Fog clients
$S_1, S_2, \dots, S_n$	Fog servers
$T_{i,j}$	Evaluated trust node i to node j
$T^{x}_{i,j}$	Evaluated trust node i to node j based on x criteria
$T^{EC}_{i,j}$	Energy consumption in communication between nodes i and j
N_{ET}	Number of executed tuple type
N_{RT}	Number of receipt tuple types
$Pmttf$	Average time the service provider has committed to fail
ST_i	Mental trust node i (belief, disbelief, uncertainty, atomic value)
$T^{Friendship}_{i,j\ Honesty}$	Friendship between nodes i and j
$T^{i,j}$	Honesty between nodes i and j
VTP_i	Rate of trust validity
RCR	realized connection requests
$\omega 1, \eta 1, \mu 1, \gamma 1, \rho 1$	Weight factors calculated comparatively to calculate customer trust
$\omega 2, \eta 2, \mu 2, \gamma 2, \rho 2$	Weight factors calculated comparatively to calculate server trust
α, β	Beta distribution parameters
x_i	Random variable with beta distribution
$maxiteration$	Maximum repetition of automatic learning

3.1 Criteria of trust

The trust criterion is the information needed to calculate the level of trust of a node. Usually in trust management systems, more than one feature is used to evaluate the trust of customers and fog servers. One measure of trust is service quality, which assesses the server's ability to successfully perform a request or service from a variety of perspectives. Because the customer is interested in choosing a server that can provide the right service, among the various criteria, the most important is QoS. On the other hand, the choice of the customer by the servers relies heavily on social relationships. In the proposed two-way trust management system, the criteria that fog servers will use to assess fog customer reliability are selected from the criteria of friendliness, honesty, ownership and energy consumption, while fog customers can choose from criteria such as delay, package delivery rate. In the following, some criteria about reliability, response time, ownership and power consumption to assess the value of a server's trust are presented.

3.1.1 Ownership criteria for fog clients and fog servers

Each fog node has a node owner. This criterion is assumed to be devices belonging to the same person trusting each other [40]. Therefore, if one node encounters another node that belongs to one person, the value of trust is setto one, otherwise, the value of this criterion is zero.

$$T_{i,j}^{\text{Ownership}} = \begin{Bmatrix} 1 & \text{Owner of i,j is same person,} \\ 0 & \text{Otherwise} \end{Bmatrix} \qquad (1)$$

3.1.2 QoS Criteria: To assess the trustworthiness of fog servers

– **Energy consumption**: The energy consumption can be expressed by Eq. (2) for all fog servers and cloud when they have serviced the input modules.

$$T_{i,j}^{\text{EC}} = E_c + (T_n - T_{lu}) * P_h \qquad (2)$$

We calculate the energy consumption of fog server by the power of all hosts in a certain time frame of execution. Where E_c is energy consumption in the current state, T_n is the current time, T_{lu} is the update time at the last utilization, and P_h is the power of a host in the last utilization [35].

– **Latency**: The time required for a fog server to provide customer service. Excessive latency and irregularity in response time may lead to unauthorized entry and intrusion into the system. The latency of application execution is calculated by the system clock and tuple end time.

$$T_{TN} = \begin{Bmatrix} SC - T_{st} & if\, T_a \geq 0, \\ \dfrac{T_{st} * N_{ET} + (SC - T_{st})}{N_{ET} + 1} & if\, T_a \geq 0 \end{Bmatrix} \qquad (3)$$

The end time of tuple is calculated by Eq. (3). Where T_{st} is the tuple start time, SC is the system clock, $(SC - T_{st})$ is the execution time, and $N_{ET}$is the number of executed tuple types. T_a is the average CPU time basedon the tuple type. CC is the system clock and ET is the emitting time of a tuple. T_s is transfer time between two modules [35].

$$T_{TR} = \frac{T_{st} * N_{RT} + SC - T_s}{N_{ET} + 1} \qquad (4)$$

The tuple receipt time can be given by Eq. (4). Where N_{RT} is the number of receipt tuple types. According to Eq. (5), the application latency is calculated by difference time between tuple end time in a module and tuple receipt time in another module.

$$Latency = T_{TR} - T_{TN} \qquad (5)$$

- **Packet Delivery Ratio (PDR)**: The number of packages successfully received is the total number of shipments. This ratio is the number of packets received by the destination node to the number of packets sentby the source node. Also in another solution, the PDR is modeled usinga well-known Gilbert-Elliott closed-loop model (Markov dual-state model) [41] in which p is the probability of transition from good to bad node

$$p = P(q_t = B \,||\, q_{t-1} = G) \tag{6}$$

and r is the probability of transition from bad to good.

$$p = P\left(q_t = G \,||\, q_{t-1} = B\right) \tag{7}$$

Model parameters are obtained by observing the loss of packets in the links of a good and bad node. Hence, the probability of transition from good to bad and the probability of transition from bad to good is based on the PDR result of the FC environment.

$$A = \begin{pmatrix} 1-p & p \\ r & 1-r \end{pmatrix} \tag{8}$$

- **Reliability**: Reliability can be defined as how a service can operate with- out failure over a period of time and under certain conditions. Therefore, reliability is defined based on the average commitment time given by the service provider for failure or previous failures experienced by users. This value can be expressed by Eq. (9).

$$Reliability = (1 - \frac{numFailure}{n}) * \text{Pmttf} \tag{9}$$

Where, *numFailure* is the number of users who experienced a crash or failure in a period of time less than what the service provider has promised,*n* is the number of users, and *Pmttf* is the average time the service provider has committed to failing.

- **Response Time**: Indicates the time required for the service provider to submit the request. The response time can be given by Eq. (10). Response time is the average response time, T_i is the time between when user i requested service and when the service was actually available, and *n* is the total number of service requests sent to a server.

$$ResponseTime = \frac{1}{n}\sum_{i=1}^{n} T_i \tag{10}$$

– **End-to-End Packet Forwarding Ratio(EPFR)**: This criterion is de- fined as the ratio between the number of packets received by the application node of the destination node to the number of packets sent by the source node. Eq. (11) can be used to calculate that Sent by the source node. Where, *k* indicates the number of successful receipts and *n* represents the total number of packets sent.

$$\mathrm{EFPT} = \frac{\sum_{i=1}^{k} \mathrm{RECV}_i}{\sum_{i=1}^{n} \mathrm{SEND}_i} \tag{11}$$

3.1.3 Social criteria: to assess the trust of fog customers

– **Energy Consumption**: This parameter can be considered in determining the amount of energy consumption of a customer: data size, transmission distance, computational energy of data sent and the amount of MIPS requested. It can be expressed by Eq. (2).
– **Friendship**: Indicates the degree of proximity of one node compared to other nodes. The calculation of friendship can be related to the history of interaction [42] in such a way that the experience of more positive interaction between the two nodes indicates more trust and confidence between them. Friendship is calculated as the rate of a customer's successful connection requests (SCR_i) to the maximum of all connection requests (CR). If the request is accepted by the server, the connection request will be considered successful. A server accepts a client connection request if the client trust exceeds the threshold required by particular application software.

$$T_{i,j}^{\text{Friendship}} = \frac{SCR_i}{\max(CR)} \tag{12}$$

– **Honesty**: Honesty evaluates the belief that a node is reliable based on direct observations of other nodes over a period of time. This criterion is calculated by keeping the number of suspected fraudulent experiences of the trusted node observed by the trusting node over a period of time using aset of anomaly detection rules, such as large differences in recommendationas well as interruptions, re-transmissions, repetitions, and delay occurs [43, 44]. Hence, according to Eq. (13), honesty can be measured as the rate of trust validity ($V\ TP_i$) and realized connection requests (RCR). Realized connection requests result in a reliable relationship between the truster and the trustee. Exaggerated customer recommendations are perceived as unreliable releases, and connection requests are rejected by low-trust nodes.

$$T_{i,j}^{\text{Honesty}} = \frac{VTP_i}{\max(RCR)} \tag{13}$$

4 The proposed approach

Given that in the proposed two-way trust management system, in additionto the direct trust that is obtained through self-monitoring of servers and customers, indirect trust that results from the recommendations of neighboring nodes will be used, so to assess the trust of used methods to address the uncertainties and inaccuracies in the recommendations. For this category of problems, methods based on fuzzy logic and subjective logic are proposed. These methods allow conclusions about trust to be presented with insufficient evidence. The proposed system is based on a specific version of the belief theory called subjective logic [45], distributed and event-oriented, which uses social trust, QoS, and energy consumption information with multiple trust characteristics to calculate the trust values of fog nodes which the trustee (trustor) and the trustee evaluate each other to establish a trusted relationship. The direct trust gained from self-observation and the indirect trust gained from the recommendations of neighboring nodes is used to determine the final amount of trust.

4.1 Subjective Logic

Mental logic is a kind of probabilistic logic and a special form of belief the- ory that creates a relationship between uncertainty and belief. Mental logic is suitable for modeling and analyzing situations or propositions that involve uncertainty and relatively unreliable sources. The proposition is expressed as a probability in the range 0 to 1 [45]. Trust is one of these propositions. The proposed system in this research will use subjective logic to gather the recom- mendations of neighboring fog servers where there is uncertainty. Mental logic is based on the belief that trust as a claim that expresses a theory about an entity is subjective and is experienced differently by each person, so it is not general and objective. Practitioners may not be able to meet all the criteria fortrust to assess a node's level of trust. This means that trust is calculated with insufficient evidence, and each node in the FC environment mentally calculates its own amount of trust in each node it encounters.

In subjective logic, uncertain probabilities are presented as ideas and opin- ions. The comment or degree of trust in node x, denoted by W_x, is defined as the following [45]:

$$W_x = \left(b_x, d_x, u_x, a_x\right) \tag{14}$$

Where b_x is the belief in the reliability of node x, d_x is the belief in the relia- bility of the node x, u_x is the uncertainty for the conclusion of the reliability of the node, and the atomic value of a_x is the uncertainty. To what extent doesit play a role in the amount of trust If the atomic value is 0.5, the probabil- ity of presenting a true or false output for an opinion is equal (equal chanceof presenting a true or false output). The sum of belief, unbelief, and uncer- tainty must be equal to 1. The degree of trust expressed as a quadruple can be converted to a single value of trust using the following equation:

$$P\left(x\right) = b_x + a_x * u_x \tag{15}$$

To calculate the degree of trust of nodes in the fog network, we obtain the values of belief (b_x), unbelief (d_x) and uncertainty (u_x) of node x from its positive and negative experiences [23, 45]. Fog nodes calculate the good and bad experiences of the nodes they encounter and send these values as a men- tal trust when asked for advice. Some advisers may not be telling the truth. To increase the accuracy of the trust, calculations must be made on the rec- ommendations. Weighing the recommendations and then combining them to determine the accuracy of the recommendations, the next task is to "gather trust" from the five dimensions of trust calculation. In subjective logic, the combination is performed using two operators, Discounting and Consensus [46].

4.1.1 Discounting operator

The node that wants to calculate the amount of trust in other nodes compares the recommendations received using discounting (can be shown by the operator $\otimes$) with the amount of trust it has in the recommenders (Fig. 3). Hence the trust values of reputable consultants are more than the trust values of less trusted consultants. This is essential for defending against trust-based attacks.

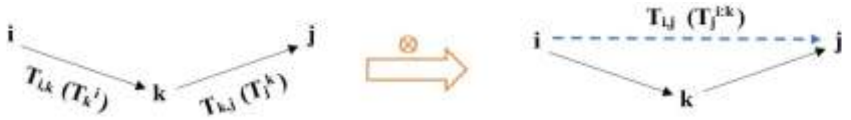

Fig. 3: Discounting operator

Assume that the trust node *i* has subjective trust T_i, *k* with respect to the recommending node *k*, and the recommending node *k* has subjective trust T_k, *j* with respect to the trusted node *j*, these expressions are shown in Eq. (16). The indirect trust assessment of node *i* to node *j*, which is based on the recommendations of node *k*, is calculated by combining trust *k* to *j* and trust *i* to *k* as follows:

$$T_{i,j} = (b_{i,k}b_{k,j}, b_{i,k}d_{k,j}, d_{i,k} + u_{i,k} + b_{i,k}u_{k,j}, a_{k,j}) \tag{16}$$

4.1.2 Consensus operator

The recommendations of several advisors are combined using consensus (indicated by operator $\oplus$ This operator works the same for each recommendation. Assume that nodes *i* and *k* have recommendations for node *j* as shown in Eq.

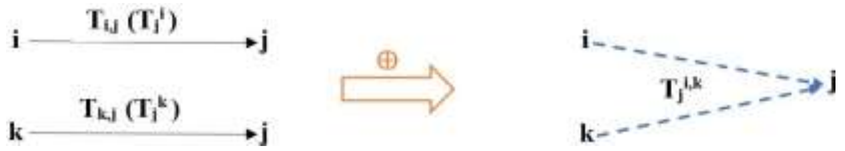

Fig. 4: Consensus operator

(17). The recommendation for *j* obtained from the combination of recommendation *i* and *k* is calculated by the following equation.

$$T_{ik,j} = \left(\frac{b_{i,j} * u_{k,j} + b_{k,j} * u_{i,j}}{k}, \frac{d_{i,j} * u_{k,j} + d_{k,j} * u_{i,j}}{k}, \frac{u_{i,j} * u_{k,j}}{k}, a_{ik,j}\right) \tag{17a}$$

$$k = u_{i,j} + u_{k,j} - u_{i,j} * u_{k,j} \tag{17b}$$

$$a_{ik,j} = \frac{a_{i,j} * u_{k,j} + a_{k,j} * u_{i,j} - (a_{i,j} + a_{k,j}) * u_{i,j} * u_{k,j}}{u_{i,j} + u_{k,j} - 2u_{i,j} * u_{k,j}} \tag{17c}$$

4.1.3 Trust calculation

Indirect trust is calculated as subjective trust, using the application of reduction and consensus operators on recommendations obtained from neighboring nodes. In order to obtain more reliable recommendations to be more resistant to trust-based attacks, recommendations can be obtained using a threshold- based filter only from reputable advisors [47]. If the number of neighboring nodes is limited, all recommendations can be considered, and then trust-basedattacks can be prevented in the confidence-building phase by using reduction and consensus operators.

Suppose node i has confidence in the recommenders $r1$, $r2$, . . . , rk, respectively, at time t as $T_{i,r1}(t)$, $T_{i,r2}(t)$,. . . , $T_{i,rk}(t)$, and the recommenders trust node j at time t as $T_{r1,j}(t)$, $T_{r2,j}(t)$, . . . , $T_{rk,j}(t)$. Then, using the reduction and consensus operators, the cumulative and final indirect trust in node j, which is evaluated by node i, is calculated by Eq. (18).

$$T_{i,j}^{Indirect}(t) = (T_{i,r_1}(t) \otimes T_{r_1,j}(t)) \oplus (T_{i,r_2}(t) \otimes T_{r_2,j}(t)) \oplus ... \oplus (T_{i,rk}(t) \otimes T_{r_k,j}(t)) \quad (18)$$

In addition to the recommendations, the trust management system also depends on the calculated trust value of a node from direct observation. The amount of direct trust in node j, which is evaluated by node i at time t, is calculated using Eq. (19).

$$T_{i,j}^{Direct}(t) = \omega T_{i,j}^{x}(t) + \eta T_{i,j}^{y}(t) + \mu T_{i,j}^{z}(t) + \rho T_{i,j}^{w}(t) \quad (19)$$

In Eq. (19), x, y, z and w are the criteria of trust and ω, η, μ and ρ are the weight factors of the criteria of individual trust. For fog servers, x, y, z, and w can be latency, PDR, ownership, and power consumption, respectively, and for fog customers, it can be friendliness, honesty, ownership, and power consumption, respectively. Weight factors of the QoS criteria are energy consumption and social trust based on the degree of trust and trustworthiness belonging to the node owner as well as the trusted reputation. The reputation here shows the amount of overall trust in a node in previous encounters.

If the trustee and the trustee are owned by one person or the trustee's reputation is higher than the threshold set for the trust, the average value fortrust criteria is considered. Otherwise, half of the weight provided for other trust criteria can be assigned to the ownership criterion. This simple weighting method encourages well-credited nodes and punishes badly credited nodes. In addition, this method is suitable for customers with limited resources. Eq. (20) is used to calculate the final amount of trust in a node:

$$T_{i,j}(t) = (1-\gamma) * T_{i,j}^{Direct}(t) + \gamma * T_{i,j}^{Indirect}(t) \tag{20}$$

The γ factor determines the share of direct and indirect trust in the final amount of trust. The contribution of indirect trust γ to total trust is determined based on the amount of trust in the trusted person in previous trust calculations $T_{i,j}(t-\Delta t)$.

$$\Gamma^{Indirect} = \frac{n(rec) * T_{i,j}(t-\Delta t)}{\max(rec) + n(rec) * T_{i,j}(t-\Delta t)} \tag{21}$$

The number of recommenders with $n(rec)$ and the maximum possible recommenders determined during the simulation settings is displayed with max(rec) and the indirect trust share is calculated by Eq. (21). The weight factor of indirect trust in the past experience and the current number of recommenders is normalized to the maximum possible number of recommenders per node. This formula shows that the share of indirect trust does not exceed half of the total trust and the increase in share is not proportional to the number of recommenders.

4.2 Beyesian learning automata

Learning automation, which is a type of reinforcement learning, can be considered as a single object with a limited number of actions. It selects an action from its action set based on the probability vector and applies it to a random environment. The environment receives the action as input and generates a response signal. With a constant unknown probability distribution, the response is evaluated to determine whether it is favorable or unfavorable to the environment. If the answer is appropriate, action will be selected. Otherwise, the answers will be updated. Therefore, it automatically uses the environment response to select its next action, which is determined by the learning automata algorithm. Selective action greatly affects search performance, and during this process, the system automatically learns to select the optimal action. The relationship between random automatics and the environment is shown in Fig. 2 ina generalized way in the fog environment [48, 49]. BLA is inherently Bayesian in nature and is based on the calculation of rewards or punishments and random sampling of the beta distribution [35]. Bayesian learning is a probabilistic method of inference that weighs the evidence of hypotheses. Its purpose is to find optimal solutions to problems. The beta distribution formula is as follows [50]:

In BLA, we use the beta distribution with two positive parameters as α and β. The probability density function can be calculated by Eq. (22) [35] as follows.

$$f\left(x;\alpha,\beta\right)=\frac{x^{\alpha-1}(1-x)^{\beta-1}}{\int_0^1 u^{\alpha-1}(1-u)^{\beta-1}du},x\in\left[0,1\right] \tag{22}$$

4.2.1 TTLA

In this research, BLA is used to learn all the moves and actions in the network in order to achieve the most reliable node (and the lowest energy consumption). Details of the steps of the two-way trust calculation algorithm are described below. The final trust value is in the range [1, 0], where 0 indicates complete distrust and 1 indicates complete trust [43]. The threshold or ignorance based on which trust and distrust are recognized varies depending on the type of application. For most applications, this value is set to 0.5, but for health and safety-sensitive applications, this value is usually higher.

According to Algorithm. 1, at first the initial parameters are set. The selection process for a fog server is then performed by BLA. The assessment of direct trust in the server is performed using direct observations by calculating the set criteria. Evaluation of indirect trust in the server is done by aggregating the recommendations of neighboring nodes through mental logic. Next, the two direct and indirect trusts are combined based on their respective weights and threshold comparisons. Following the BLA algorithm, the α and β parameters (encouragement and punishment) are increased. In the next step, the client requests the best server selected by BLA. The assessment of direct trust in the customer is done using direct observations by calculating the set criteria. The evaluation of indirect trust in the customer is done by aggregating the recommendations of neighboring nodes through mental logic. Both direct and indirect trusts are combined based on the respective weights and threshold comparisons. If the customer is trusted, mutual trust is established and the connection is established for service, otherwise, the customer is considered untrustworthy and the request is rejected by the server.

5 Evaluation

The performance of the proposed approach is presented in this section. The simulation environment is iFogsim library [51]. This simulator has many classes to implement resource management strategies. We have extended some classes as the ModulePlacementEdgeward for trust management and offloading, con- troller for more output metrics. Also, the DCNS class is customized based on the architecture of this paper. We simulated the proposed algorithm and compare the results with two other methods. The one is a two-way trust man- agement system in fog systems (MTTA) [2] and another one is a fuzzy-based trust evaluation framework (Fuzzy) [29].

Algorithm 1 TTLA

```
Input: $C_1, C_2, ..., C_n, S_1, S_2, ..., S_n, \omega1, \eta1, \mu1, \rho1, \gamma1, \omega2, \eta2, \mu2, \rho2, \gamma2, \alpha_i = \beta_i = 1, R = 1, MaxIteration$
Output: Trusted connection established
1: for all $C_i \in Clients$ do
2:     while R ≤ MaxIteration do
3:         for all $j \in C_i$'s neighboring servers do
4:             Calculate $x_j$ randomly from beta distribution as Eq. (22).
5:             /* Calculate direct trust of server $S_j$. */
6:             Calculate latency, PDR, and energy consumption as Eqs. (5), (2), and (8).
7:             if $(C_i, S_j) \in$ same person then
8:                 $T_{i,j}^{Ownership} = 1$.
9:             else
10:                $T_{i,j}^{Ownership} = 0$.
11:            end if
12:            Calculate $T_{i,j}^{Direct}$ under $\omega2, \eta2, \mu2$, and $\rho2$ as Eq. (19).
13:            /*Calculate indirect trust of server $S_j$ */
14:            $ST_i$ = Initial Subjective Trust Value of $S_j$.
15:            Calculate $T_{i,j}^{Indirect} = ST_i.Belief + ST_i.Uncertainty * STi.Atomicity$.
16:            /* Calculate total trust of server $S_j$ */
17:            $T_{i,j}(t) = (1 - \gamma1) * T_{i,j}^{Direct}(t) + \gamma1 * T_{i,j}^{Indirect}(t)$
18:            if $T_{i,j}(t) \geq Thereshold1$ then
19:                $\alpha_i = \alpha_i + 1$.
20:            else
21:                $\beta_i = \beta_i + 1$.
22:            end if
23:        end for
24:    end while
25:    /* Calculate direct trust of client $C_i$ */
26:    Calculate $T_{i,j}^{Friendship}$, $T_{i,j}^{Honesty}$, and $T_{i,j}^{EC}$ as Eqs. (12), (13), and (2).
27:    if $(C_i, S_j) \in$ same person then
28:        $T_{i,j}^{Ownership} = 1$.
29:    else
30:        $T_{i,j}^{Ownership} = 0$.
31:    end if
32:    Calculate $T_{i,j}^{Direct}$ under $\omega1, \eta1, \mu1$, and $\rho1$ as Eq. (19).
33:    /* Calculate indirect trust of client $C_i$ */
34:    $ST_j$ = Initial Subjective Trust Value of $C_i$.
35:    Calculate $T_{i,j}^{Indirect} = ST_i.Belief + ST_i.Uncertainty * STi.Atomicity$.
36:    /* Calculate total trust of client $C_i$ */
37:    $T_{i,j}(t) = (1 - \gamma2) * T_{i,j}^{Direct}(t) + \gamma2 * T_{i,j}^{Indirect}(t)$
38:    if $T_{i,j}(t) \geq Thereshold2$ then
39:        Make trusted connection.
40:    else
41:        Connection failed because of unreliable client.
42:    end if
43: end for
```

5.1 Simulation configuration

This section presents all configurations for devices and application modules. The number of areas equals 4 and we evaluate the simulation based on different number of cameras and fog devices (FDs). Table 3 shows the edge configuration for the application.

Table 3: The edge configuration for the application

Source module	Destination module	Tuple CPU length (B)	Tuple New length (B)
Camera	Motion detector	1000	20000
Motion detector	Object detector	2000	2000
Object detector	User interface	500	2000
Object detector	Object tracker	1000	100
Object tracker	PTZ control	28	100

Table 4: FD configuration

Property	Cloud	Proxy-server	Area	Camera
MIPS	44800	2800	2800	500
RAM	40000	4000	4000	1000
UpBw	100	10000	10000	10000
DownBw	10000	10000	10000	10000
Level	0	1	1	3
Rate per MIPS	0.01	0	0	0
Busy power	1648	107.339	107.339	87.53
Idle power	1.332	83.4333	83.4333	82.44

As shown in Table 4, each FD has several parameters including the MIPS, RAM (Kilobyte), UpBW (Upper bandwidth by kilobyte per second), DownBW (Down bandwidth by kilobyte per second), level in the hierarchical topology, rate per MIPS, busy and idle power (Mega Watt). The application module only has a bandwidth feature that is set in the UpBW column. Table 5 shows the host's configuration. The host's properties include the storage, bandwidth, architecture, operating system, VM model, time zone, cost, cost per memory, and cost per storage. Table 6 shows the modules' properties, such as the RAM, MIPS, size of the module, and bandwidth.

5.2 Energy consumption

One of important metrics in this research is the energy consumption. According to Fig. 5, the proposed trusted approach as TTLA has better results than Fuzzy and MTTA methods. In fact, using the BLA algorithm made more

Table 5: Configuration of host

Parameter	Value
Storage	1000000 B
BW	10000 B/S
Architecture	x86
OS	Linux
VM model	Xen
Time zone	10
Cost	3
Cost per memory	0.05
Cost per storage	0.01

Table 6: Configuration of the application module

RAM	MIPS	Size	BW
10 B	1000	10000 B	1000 B/S

efficient trusted strategy than other methods and this affects the energy consumption of FDs.

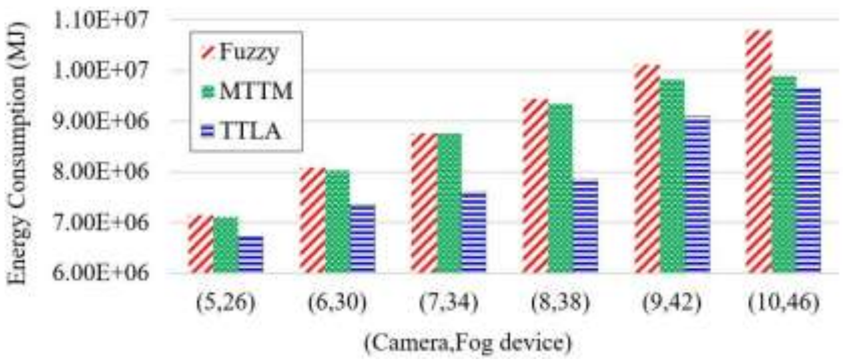

Fig. 5: Analysis of the energy consumption

5.3 Total execution cost

Fig. 6 shows total execution cost of the system. According to this diagram, when the number of FDs increases more than 30, cost will be raised. Obviously and totally the TTLA method has less cost than others. As another result of this figure, using intelligent methods is useful to optimize the trust issue in this research.

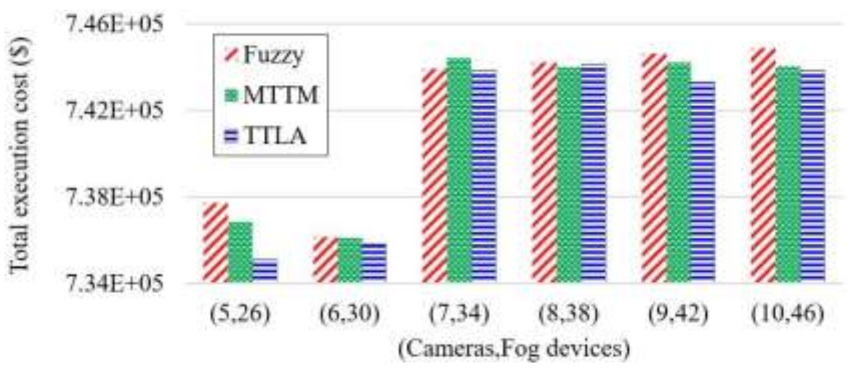

Fig. 6: Analysis of total execution cost

5.4 Network usage

Analysis of the network usage helps us to understand which method is more efficient in resource usage. As much as a method can keep resources busy and reduce their idle time, it is the better. According to Fig. 7, Fuzzy has least network usage and also TTLA is the most efficient than others.

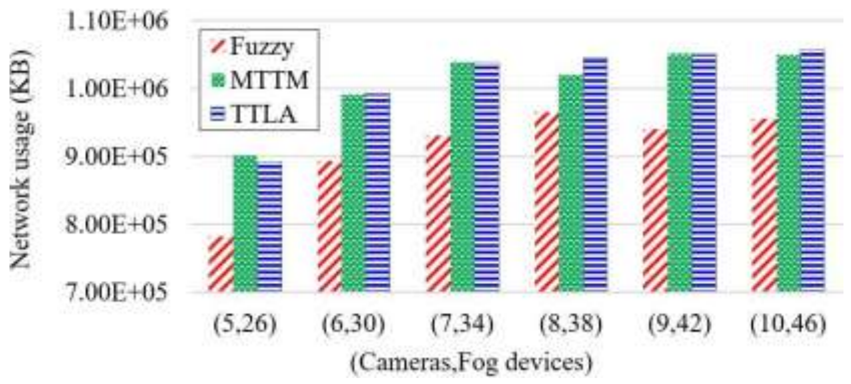

Fig. 7: Analysis of network usage

5.5 Delay

Fig. 8 demonstrates the TTLA can make a trusted connection between FDs with least latency. It helps the network for real-time applications. In this diagram, we can see a gentle growth of delay value while increasing the number of FDs.

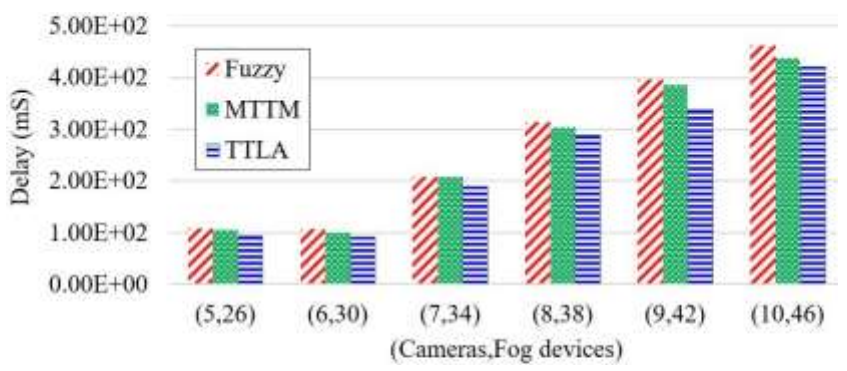

Fig. 8: Analysis of the delay

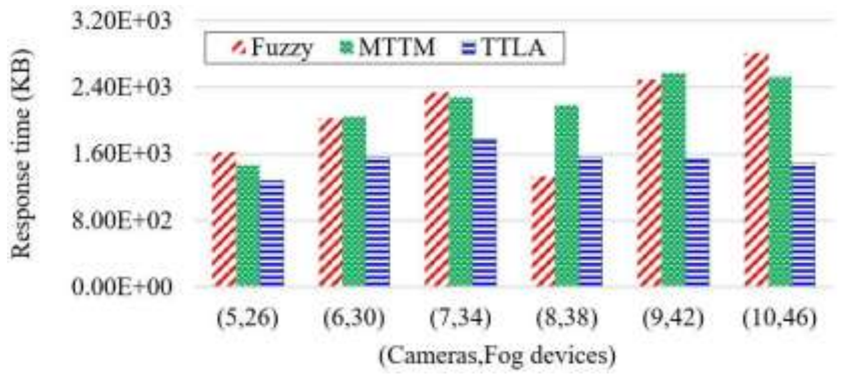

Fig. 9: Analysis of the response time

5.6 Response time

Since real-time applications need a fast response messages thus the response time is a good metric in this respect. According to Fig. 9, the TTLA has a response time between $1.28_* 10^3$ and $1.78_* 10^3$. It is such a good results than other methods with range of $1.28_* 10^3$ and $2.82_* 10^3$. It means the TTLA can find a trusted FD to offload task to it faster that other methods.

5.7 Trust value

Fig. 10 shows the trust value when a client request servers to find more trusted nodes of them. Since, the TTLA and MTTA are based on a trust value, we compare them in this figure. The higher trust by the TTLA proves more secure method than MTTA. The highest trust value of the TTLA are related to evaluation with 26 and 38 FDs.

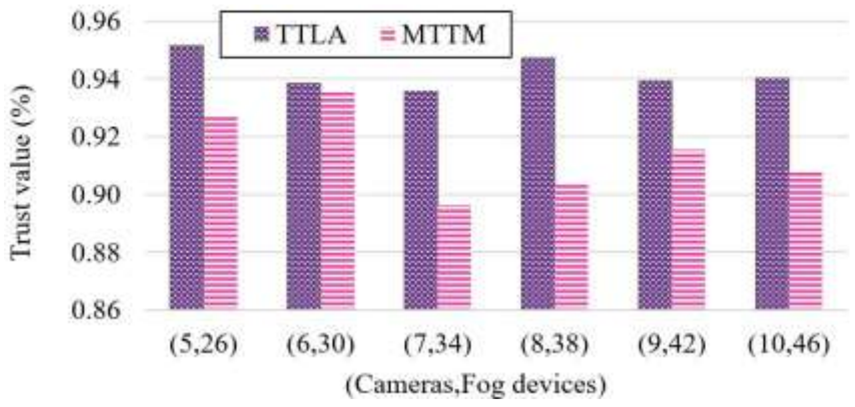

Fig. 10: Analysis of the trust value

6 Conclusion

This paper presents a two-way trust management strategy based on BLA in FC-based applications. In each communication, we have some fog nodes as clients and servers. Each of them tends to make communication with the highest trusted nodes. Also, energy efficiency is another challenge of this work. A two-way trust algorithm helps us to manage security between client and server. Of course, when the number of nodes increases, we need a machine learning algorithm instead of the heuristics. After using the BLA, the proposed approach was better than Fuzzy and MTTA methods. The proposed approach outperforms the others by 10% in energy consumption, 5% in network usage, 9% in latency, 28% in response time, and 3% in trust value. For future work, we try to extend the proposed strategy to a fully distributed method. All nodes in the network need to be aware of the network without transferring sensitive secure data. In this respect, developing a distributed trust management algorithm can be useful.

Ethics approval and consent to participate
The paper is original that has not been submitted/published anywhere.

Consent for publication
All authors agree with publishing this work.

Availability of data and materials
The authors declare that there is no data and materials related to this work.

Competing interests
The authors declare that they have no conflict of interest.

Funding
The authors declare that there is no funding for this work.

Authors' contributions
Niloofar Barati Bakhtiari wrote drafted the manuscript with support from Dr. Rafighi as supervisor and Dr. Ahsan as advisor of article. All authors discussed and analyzed the results of the research and contributed to the writing of the final manuscript.

Acknowledgements
The authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve the quality of this paper.

References

1. Ogundoyin, S. O., & Kamil, I. A. (2021). A trust management system for fog computing services. Internet of Things, 14, 100382. https://doi.org/10.1016/j.iot.2021.100382
2. Alemneh, E., Senouci, S. M., Brunet, P., & Tegegne, T. (2020). A two-way trust management system for fog computing. Future Generation Computer Systems, 106, 206-220. https://doi.org/10.1016/j.future.2019.12.045

3. Kang, J., Yu, R., Huang, X., & Zhang, Y. (2017). Privacy-preserved pseudonym scheme for fog computing supported internet of vehicles. IEEE Transactions on Intelligent Trans- portation Systems, 19(8), 2627-2637. https://doi.org/10.1109/TITS.2017.2764095
4. Okay, F. Y., & Ozdemir, S. (2016, May). A fog computing based smart grid model. In 2016 international symposium on networks, computers and communications (ISNCC) (pp.1-6). IEEE. https://doi.org/10.1109/ISNCC.2016.7746062
5. Rahmani, A. M., Gia, T. N., Negash, B., Anzanpour, A., Azimi, I., Jiang, M., & Lilje- berg, P. (2018). Exploiting smart e-Health gateways at the edge of healthcare Internet-of-Things: A fog computing approach. Future Generation Computer Systems, 78, 641-658. https://doi.org/10.1016/j.future.2017.02.014
6. Al Hamid, H. A., Rahman, S. M. M., Hossain, M. S., Almogren, A.,& Alamri, A. (2017). A security model for preserving the privacy of medical big data in a healthcare cloud using a fog computing facility with pairing-based cryptography. IEEE Access, 5, 22313-22328. https://doi.org/10.1109/ACCESS.2017.2757844
7. Elmisery, A. M., Rho, S., & Botvich, D. (2016). A fog based middleware for automated compliance with OECD privacy principles in internet of healthcare things. IEEE Access,4, 8418-8441. https://doi.org/10.1109/ACCESS.2016.2631546
8. Moosavi, S. R., Gia, T. N., Nigussie, E., Rahmani, A. M., Virtanen, S., Ten- hunen, H., & Isoaho, J. (2016). End-to-end security scheme for mobility enabled healthcare Internet of Things. Future Generation Computer Systems, 64, 108-124. https://doi.org/10.1016/j.future.2016.02.020
9. Liu, X., Deng, R. H., Yang, Y., Tran, H. N., & Zhong, S. (2018). Hybrid privacy-preserving clinical decision support system in fog–cloud computing. Future Generation Computer Systems, 78, 825-837. https://doi.org/10.1016/j.future.2017.03.018
10. Bonomi, F., Milito, R., Zhu, J., & Addepalli, S. (2012, August). Fog computing and its role in the internet of things. In Proceedings of the first edition of the MCC workshop on Mobile cloud computing (pp. 13-16). https://doi.org/10.1145/2342509.2342513
11. Zhang, P., Zhou, M., & Fortino, G. (2018). Security and trust issues in Fog computing: A survey. Future Generation Computer Systems, 88, 16-27. https://doi.org/10.1016/j.future.2018.05.008
12. Alemneh, E., Senouci, S. M., & Brunet, P. (2017, October). PV-Alert: A fog-based architecture for safeguarding vulnerable road users. In 2017 Global In- formation Infrastructure and Networking Symposium (GIIS) (pp. 9-15). IEEE. https://doi.org/10.1109/GIIS.2017.8169804
13. Hbaieb, A., Ayed, S., & Chaari, L. (2022). A survey of trust man- agement in the Internet of Vehicles. Computer Networks, 203, 108558. https://doi.org/10.1016/j.comnet.2021.108558
14. Lai, C. F., Song, D. Y., Hwang, R. H., & Lai, Y. X. (2016, July). A QoS-aware streaming service over fog computing infrastructures. In 2016 Digital Media Industry & Academic Forum (DMIAF) (pp. 94-98). IEEE. https://doi.org/10.1109/DMIAF.2016.7574909
15. Fernández-Caramés, T. M., Fraga-Lamas, P., Suárez-Albela, M., & Vilar-Montesinos, M. (2018). A fog computing and cloudlet based augmented reality system for the industry 4.0 shipyard. Sensors, 18(6), 1798. https://doi.org/10.3390/s18061798
16. Aazam, M., Zeadally, S., & Harras, K. A. (2018). Offloading in fog computing for IoT: Review, enabling technologies, and research opportunities. Future Generation ComputerSystems, 87, 278-289. https://doi.org/10.1016/j.future.2018.04.057
17. Tang, B., Chen, Z., Hefferman, G., Pei, S., Wei, T., He, H., & Yang, Q. (2017). Incorpo- rating intelligence in fog computing for big data analysis in smart cities. IEEE Transactions on Industrial informatics, 13(5), 2140-2150. https://doi.org/10.1109/TII.2017.2679740
18. Singh, S., & Kandpal, M. (2022). A Comprehensive Survey on Trust Management in Fog Computing. ICT Analysis and Applications, 87-97. https://doi.org/10.1007/978-981- 16-5655-2 9
19. Mukherjee, M., Matam, R., Shu, L., Maglaras, L., Ferrag, M. A., Choudhury, N., & Kumar, V. (2017). Security and privacy in fog computing: Challenges. IEEE Access, 5, 19293-19304. https://doi.org/10.1109/ACCESS.2017.2749422
20. Tariq, N., Asim, M., Al-Obeidat, F., Zubair Farooqi, M., Baker, T., Hammoudeh, M., & Ghafir, I. (2019). The security of big data in fog-enabled IoT applications including blockchain: A survey. Sensors, 19(8), 1788. https://doi.org/10.3390/s19081788
21. Ni, J., Zhang, K., Lin, X., & Shen, X. (2017). Securing fog computing for internet of things applications: Challenges and solutions. IEEE Communications Surveys & Tutorials,20(1), 601-628. https://doi.org/10.1109/COMST.2017.2762345
22. Wei, L., Yang, Y., Wu, J., Long, C., & Li, B. (2022). Trust Management for Internet of Things: A Comprehensive Study. IEEE Internet of Things Journal. https://doi.org/10.1109/JIOT.2021.3139989
23. Dybedokken, T. S. (2017). Trust management in fog computing (Master's thesis, NTNU). http://hdl.handle.net/11250/2454375
24. Yan, Z., Zhang, P., & Vasilakos, A. V. (2014). A survey on trust management for Internet of Things. Journal of network and computer applications, 42, 120-134.https://doi.org/10.1016/j.jnca.2014.01.014
25. Okay, F. Y., & Ozdemir, S. (2018). Routing in fog-enabled IoT platforms: A sur- vey and an SDN-based solution. IEEE Internet of Things Journal, 5(6), 4871-4889. https://doi.org/10.1109/JIOT.2018.2882781
26. Agushaka, J. O., & Ezugwu, A. E. (2022). Initialisation Approaches for Population-Based Metaheuristic Algorithms: A Comprehensive Review. Applied Sciences, 12(2), 896.https://doi.org/10.3390/app12020896
27. Thakur, P. S., Kiran, U., & Sahu, O. P. (2022). A Review on: Machine Learn-ing Techniques to Mitigate Security Risks in IoT Framework State of the Art. In Futuristic Communication and Network Technologies (pp. 671-679). Springer, Singa- pore. https://www.springerprofessional.de/en/a-review-on-machine-learning-techniques- to-mitigate-security-ris/19748932

28. Saha, A., Chowdhury, C., Jana, M., & Biswas, S. (2021). IoT Sensor Data Analysis and Fusion Applying Machine Learning and Meta-Heuristic Approaches. Enabling AI Applications in Data Science, 441-469. http://dx.doi.org/10.1007/978-3-030-52067-0 20
29. Rahman, F. H., Au, T. W., Newaz, S. S., Suhaili, W. S., & Lee, G. M. (2020). Find my trustworthy fogs: A fuzzy-based trust evaluation framework. Future Generation Computer Systems, 109, 562-572. https://doi.org/10.1016/j.future.2018.05.061
30. Rathee, G., Sandhu, R., Saini, H., Sivaram, M., & Dhasarathan, V. (2020). A trust computed framework for IoT devices and fog computing environment. Wireless Networks, 26(4), 2339-2351. https://doi.org/10.1007/s11276-019-02106-3
31. Al-Khafajiy, M., Baker, T., Asim, M., Guo, Z., Ranjan, R., Longo, A., Puthal, D., & Taylor, M. (2020). COMITMENT: a fog computing trust management approach. Journal of Parallel and Distributed Computing, 137, 1-16. https://dx.doi.org/10.1016/j.jpdc.2019.10.006
32. Hussain, Y., Zhiqiu, H., Akbar, M. A., Alsanad, A., Alsanad, A. A. A., Nawaz, A., Khan, I.A., & Khan, Z. U. (2020). Context-aware trust and reputation model for fog-based IoT. IEEE Access, 8, 31622-31632. https://doi.org/10.1109/ACCESS.2020.2972968
33. Awan, K. A., Din, I. U., Almogren, A., Almajed, H., Mohiuddin, I., & Guizani, M. (2020). Neurotrust-artificial neural network-based intelligent trust management mechanism for large-scale internet of medical things. IEEE Internet of Things Journal. http://dx.doi.org/10.1109/JIOT.2020.3029221
34. El-Sayed, H., Ignatious, H. A., Kulkarni, P., & Bouktif, S. (2020). Machine learning based trust management framework for vehicular networks. Vehicular Communications, 25, 100256. https://doi.org/10.1016/j.vehcom.2020.100256
35. Farahbakhsh, F., Shahidinejad, A., & Ghobaei-Arani, M. (2021). Multiuser context-aware computation offloading in mobile edge computing based on Bayesian learning automata. Transactions on Emerging Telecommunications Technologies, 32(1), e4127. https://doi.org/10.1002/ett.4127
36. Wang, T., Zeng, J., Lai, Y., Cai, Y., Tian, H., Chen, Y., & Wang, B. (2020). Data collection from WSNs to the cloud based on mobile Fog elements. Future Generation Computer Systems, 105, 864-872. https://doi.org/10.1016/j.future.2017.07.031
37. Ilyas, M., Ullah, Z., Khan, F. A., Chaudary, M. H., Malik, M. S. A., Zaheer, Z., & Durrani, H. U. R. (2020). Trust-based energy-efficient routing protocol for Internet of things–based sensor networks. International Journal of Distributed Sensor Networks, 16(10), 1550147720964358. https://doi.org/10.1177
38. Subramanian, N., GB, S. M., Martin, J. P., & Chandrasekaran, K. (2020, January). HTmRPL++: A Trust-Aware RPL Routing Protocol for Fog Enabled Internet of Things. In 2020 International Conference on COMmunication Systems & NETworkS (COM- SNETS) (pp. 1-5). IEEE.
39. K. Haseeb, N. Islam, Y. Javed and U. Tariq, ”A Lightweight Secure and Energy-Efficient Fog-Based Routing Protocol for Constraint Sensors Network,” Energies, vol. 14, 2020. https://doi.org/10.3390/en14010089
40. Atzori, L., Iera, A., & Morabito, G. (2011). Siot: Giving a social structure to the internet of things. IEEE communications letters, 15(11), 1193-1195. https://doi.org/10.1109/LCOMM.2011.090911.111340
41. Haßlinger, G., & Hohlfeld, O. (2008, March). The Gilbert-Elliott model for packet loss in real time services on the Internet. In 14th GI/ITG Conference-Measurement, Modellingand Evalutation of Computer and Communication Systems (pp. 1-15). VDE.
42. Velloso, P. B., Laufer, R. P., Cunha, D. D. O., Duarte, O. C. M., & Pujolle, G. (2010). Trust management in mobile ad hoc networks using a scalable maturity-based model. IEEE transactions on network and service management, 7(3), 172-185. https://doi.org/10.1109/TNSM.2010.1009.I9P0339
43. F. Bao and I. Chen, ”Dynamic trust management for internet of things applications,” in Proceedings of the international workshop on Self-aware internet of things, ACM, 2012. https://doi.org/10.1145/2378023.2378025
44. I. Chen and J. Guo, ”Dynamic Hierarchical Trust Management of Mobile Groups and Its Application to Misbehaving Node Detection,” in IEEE 28th International Conference on Advanced Information Networking and Applications, 2014. https://doi.org/10.1109/AINA.2014.13
45. Jøsang, A. (2016). Subjective logic (pp. 51-82). Cham: Springer. https://doi.org/10.1007/978-3-319-42337-1
46. Jøsang, A., Marsh, S., & Pope, S. (2006, May). Exploring different types of trust propa- gation. In International Conference on Trust Management (pp. 179-192). Springer, Berlin,Heidelberg. https://doi.org/10.1007/11755593 14
47. Chen, R., Guo, J., Bao, F., & Cho, J. H. (2013, December). Integrated social and quality of service trust management of mobile groups in ad hoc networks. In 2013 9th International Conference on Information, Communications & Signal Processing (pp. 1-5).IEEE. https://doi.org/10.1109/ICICS.2013.6782950
48. Granmo, O. C. (2008, December). A Bayesian learning automaton for solving two-armed Bernoulli bandit problems. In 2008 Seventh International Conference on Machine Learning and Applications (pp. 23-30). IEEE. http://dx.doi.org/10.1108/17563781011049179
49. Gheisari, S., & Meybodi, M. R. (2017). A new reasoning and learning model for Cogni- tive Wireless Sensor Networks based on Bayesian networks and learning automata coop- eration. Computer Networks, 124, 11-26. https://doi.org/10.1016/j.comnet.2017.05.031
50. Mahmoudi, M., Faez, K., & Ghasemi, A. (2020). Defense against primary user emulation attackers based on adaptive Bayesian learning automata in cognitive radio networks. Ad Hoc Networks, 102, 102147. https://doi.org/10.1016/j.adhoc.2020.102147
51. Gupta, H., Vahid Dastjerdi, A., Ghosh, S. K., & Buyya, R. (2017). iFogSim: A toolkit for modeling and simulation of resource management techniques in the Internet of Things, Edge and Fog computing environments. Software: Practice and Experience, 47(9), 1275-1296. https://doi.org/10.1002/spe.2509