Computers and Electrical Engineering 102 (2022) 108195

Contents lists available at ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng

A social qualitative trust framework for Fog computing ☆

Mahnoor Hamza [a], Waseem Iqbal [a,*], Awais Ahmad [b], Muhammad Babar [c], Sohaib Khan [a]

[a] Department of Information Security, National University of Sciences and Technology (NUST), Islamabad, 44000, Pakistan
[b] Department of Computer Science, Air University, Islamabad, 44000, Pakistan
[c] Department of Computer Science, Allama Iqbal Open University (AIOU), Islamabad, 44000, Pakistan

ARTICLE INFO

Keywords:
Fog architecture
Bayes model
Fog nodes
Trust management system
Edge computing
Fog computing

ABSTRACT

The field of Internet of Things (IoT) is evolving at an exponential rate where megabytes of data is being processed each passing minute. Fog computing is an emerging field, regarded as the new way forward to prevent the suspected data outburst of the exhausted IoT devices. Fogging is believed to reduce latency, and enhance efficiency, ease of deployment and flexibility, however, various security and privacy concerns hinder its deployment by major platforms. Among all the security concerns, lack of a proper trust management system is of primary importance. This paper explores the Service-oriented Internet of Things (SIoT) domain to study its similarity with Fogging and creates a trust management scheme (TMS). The paper further explores the necessary trust requirements for Fogging and incorporates them into a two-way trust management scheme based on Bayes model, the model allows both the service requestor and the service provider to validate each other before connecting. The model calculates the value of trust using these metrics and combines them in a unique way with Bayes trust to have an accurate trust value. The proposed scheme is simulated in Netlogo, an agent-based network simulator. The subject scheme is capable of effectively preventing a legitimate node from connecting with a malicious node. The results illustrates high accuracy and faster convergence and also shows resilience against trust-based network attacks. The system is compared against SIoT trust management models due to lack of similar trust management models in Fog Computing.

1. Introduction

Cisco Global Cloud Index claims that the amount of data produced by machines, things and people will cross 847 zettabytes by the end of 2021 [1]. To handle such a large amount of data it is imperative to provide resourceful devices at the network edge to minimize the bandwidth or latency issues [2] and also fulfill the Internet of Things (IoT) security requirements [3]. This paved way for introduction to Fog computing which supports applications involving a short response time, mobility and data confidentiality while improving latency and power consumption [4,5].

Fog is a multi-layer architecture, that enables processing and analytics to be carried out at the logical extreme of the network. Much like the IoT network, it is a network of heterogeneous nodes providing services to one another at a close proximity. Cisco has defined a fog node as a "mini-cloud" closer to the edge devices [6], it offers countless benefits to the traditional IoT-Cloud infrastructure, for example, it enhances security and reduces the risk of data leakage [7]. It also improves various IoT applications

☆ This paper is for special section VSI-sss. Reviews were processed by Guest Editor Dr. Sajid Anwar and recommended for publication.

* Corresponding author.
E-mail address: waseem.iqbal@mcs.edu.pk (W. Iqbal).

https://doi.org/10.1016/j.compeleceng.2022.108195
Received 7 December 2021; Received in revised form 13 June 2022; Accepted 21 June 2022
Available online 12 July 2022

such as e-health, smart cities, automated traffic control etc [8,9]. However, it is a new field and needs time before it is mature enough for worldwide adoption. OpenFog has introduced a generalized system architecture and guidelines for accurate and easy modeling of Fog [10] to encourage research in this field. Fog network requires a trust management system to make the behavior of nodes predictable in the otherwise geometrically distributed network [11]. The acceptability and deployment of fog has amplified the need for a secure and efficient method for data transfer and reliable service provider selection. One such method is to establish trust between the network entities by means of a trust management system (TMS).

Trust can be used as an assessment criterion to determine the security level of a respective node [12], but the situation of trust in fog infrastructure is rather complex due to its infrastructure [13]. Trust can be defined as the confidence on an object that it will behave in a predictable manner in accordance to the set of policies, trust is subjective and depends upon the network characteristics (such as deployment model, environment, security requirement, and application type). Some critical applications may require a high level of trust whereas others may not. A TMS requires a trustor and a trustee for its formation, the trustor is the entity that places its trust in another entity namely 'trustee'. In Service-oriented IoT (SIoT), the objects of the network form social relations like human beings and make autonomous decisions based on their experiences. Trust management in fog can be derived from SIoT due to similarities in their network entities. They both have service requesters (SR); the nodes requesting a service and service providers (SP); the nodes providing the service. The SR and SP communicate on the basis of confidence and reliability. A fog network requires to have a certain level of trust among the nodes for successful collaboration and communication. Furthermore, it requires trust to be established at both ends of the communication i.e. the service requester and the service provider both should only communicate if they have established trust [14].

This paper highlights the lack of a proper TMS in Fog and explores the requirements of SIoT that can be applied to it keeping in view its distributed nature and resource-constraint network elements. Furthermore, we propose a two-way trust management scheme for fog computing in the light of the existing models present in other computing paradigms. Our model makes use of the beta reputation function with belief discounting based on Bayesian inference, which computes the recommendations, these are then discounted which wards off various network attacks such as ballot stuffing and bad mouthing attack. The proposed technique is different from existing techniques on various accounts, the technique is inspired by how the trust management works in SIoT, it yields better results, faster convergence of trust, and is one of a kind system as it considers all the attributes necessary in a fog network such as centrality, reputation, service score etc. It will improve the reliability of the network, by making efficient predictions about the behavior of a node. It will allow monitoring of a large-scale distributed network for detection and expulsion of malicious objects and rogue nodes. Furthermore, it will strengthen the communication among nodes by promoting offloading to other trustworthy nodes in the network with minimum overhead.

Security and privacy has always been an issue in a dynamic Fog network, although extensive research is available on trust management schemes in other platforms such as IoT, SIoT and Cloud but the existing research on trust management schemes in Fog as a secure platform is negligible. The existing research majorly focuses on new encryption schemes and protocols, this approach can be computationally intensive and may require a central entity to process the complex encryption keys and protocols. Fog networks are generally large-scale networks comprising of various network objects also called fog nodes (i.e., any device with sufficient processing power and memory), these nodes are bound to communicate with each other for various transactions, increasing the probability of attacks. The proposed two way trust management scheme should ensure that both the nodes have established a trusted connection prior to the transaction.

In the following research, we have proposed a two-way trust management scheme for Fog Computing Paradigm to increase its reliability and efficiency. Our research contributions are as follows:

1. We propose a generic two-way trust management scheme that considers social and qualitative trust metrics to calculate trust. Our approach allows both the fog nodes, the service requester (SR) and the service provider (SP), to evaluate and validate their trustworthiness. The trust is calculated by combining the trust metrics, direct observations and recommendations using the beta reputation function.
2. We estimate Bayes trust which is based on Bayesian inference, it predicts the future behavior of the node depending upon its current status of conduct, it also helps evade "on off" attack.
3. We evaluate the convergence and accuracy of the solution by the help of Netlogo an agent based simulation tool. The evaluation also incorporates the effects of variating good and bad nodes present in the network.
4. We validate the proposed SQT management model through simulations and experimental results.

The rest of the paper is distributed as follows: Section 2 briefly describes the related research articles of SIoT, IoT and cloud that has led to the Fog management trust framework. Section 3 highlights the general Fog structure and discusses the key parameters of the proposed SQT management framework. Section 4 presents an in-depth study of the Bayes model and the trust computation. The mathematical description of the proposed scheme is present in Section 5 The simulation setup, performance, and comparative analysis are described in Section 6 along with the results. And the paper is concluded in Section IX respectively

2. Related work

The purpose of Fog is to bridge the gap between data generating devices (IoT) and data processing facilities (Cloud). It has emerged as a supporting component to cloud computing, having a decentralized structure allowing users to place their resources on logical locations [15]. Trust is necessary in an environment with risk, uncertainties and frequent collaborations such as Fog. Due to it has some similarities with SIoT [16], therefore we have thoroughly studied and analyzed the TMSs in SIoT to build a suitable

Table 1
Comparison and analysis of literature review.

Research paper	Trust metrics	Trust model	Main contributions
Service-oriented Internet of Things (SIoT)			
[17]	Social trust parameters	Behavior based model	Builds a reliable SIoT network by incorporating its trust metrics and updating trust with minimum overhead
[18]	Reputation	Guarantor based trust model	A reputation model involving a third party guarantor
[19]	1. Direct trust 2. Indirect trust	Communities of interest based trust model	A hybrid trust model that integrates social behavior of objects with their communities to model trust
[20]	1. Direct trust 2. Indirect trust	Context based model	Combines social relations of objects with context and capacity of the objects

TMS for Fog. Trust in Fog is more than just secure transfer of data, it encompasses the canons of integrity, consistency, truthfulness and reliability of a party on its service provider. It helps the node make a wiser decision for secure data transfer depending upon different parameters. In this section, the most popular techniques in existing trust management systems of SIoT and Cloud will be discussed briefly.

2.1. TMS in SIoT and MANETs

Among many popular techniques Bayesian inference is the most popular trust computation model. It was used to develop a reputation system that models trust, it uses beta reputation function to map the positive and negative experiences. These systems were capable of computing the average trust and (see Table 1).

The authors in [17] use parameters such as direct observations, centrality, community of interest (CoI), and cooperativeness to incorporate the social relations of the objects and protect the network against on off forwarding attacks. [18] proposes a centralized guarantor-based system to measure the trust of the SP. The model revolves around reputation of the network objects and neglects direct observations altogether, the major drawback of such a system is that it is unsuitable for low latency applications.

The model in [19] requires each inter-community nodes to have an elected admin responsible for managing service requests, trust calculation and seamless network operation. The same author proposes a multi-trust context-based trust management model in [20], the model works well for dynamic networks, but the overhead may be an issue in larger networks. Truong et al. published a series of researches [21,22], that mimics human cognitive process for developing trust in different situations, he proposes a model based on experience, reputation and knowledge for SIoT networks.

2.2. TMS in cloud

Cloud also has some well-established trust models, but they cannot be directly applied to Fog due to different infrastructure. The TMS in Cloud are mostly centralized, SLA based and easy to monitor and validate. Whereas, the Fog network has heterogeneous nodes that are vendor specific and dynamic in nature due to which the static reputation-based trust models of Cloud cannot be implemented to Fog. Furthermore, SLA based trust models require a licensed third party to constantly monitor and validate the nodes, which is possible for Cloud but not for Fog. Moreover, trust in Cloud is a unidirectional requirement whereas it is a bidirectional requirement in Fog.

2.3. TMS in fog

Very limited work has been done on trust computational models in Fog including [23–28]. S.A. Soleymani et al. [23] models trust using fuzzy logic to combine experience and plausibility, the model uses a set of modules to authenticate, calculate and choose the most trustworthy node. Wang et al. [24–26] proposes a model for trustworthy communication using regression analysis and fitting function that relates trust value with the communication variable in sensor cloud systems using a fog-based approach. The author extends his research in [25] and propose a hierarchical model as opposed to the linear model proposed in [24]. In both the researches the author uses fog layer to calculate the trust function, store its value and execute tasks based on the value of trust. Authors in [27] propose a lightweight scheme that use feedback from multiple sources to identify trustworthy IoT edge devices. All the above-mentioned researches use Fog as a supplementary layer either to reduce the computational cost or to enhance the storage capability of other networks. None of the models primarily focuses on Fog to create a trust model. Rahman et al. [28] propose a broker-based trust evaluation framework based on fuzzy logic for Fog that uses only QoS parameters and do not incorporate the social relations of the nodes (see Table 2).

The motivation behind creating a trust management model is to complement the rapidly growing Fog network by introducing trust in Fog we are in fact moving towards a more reliant and secure platform for safe data transfer. This work is different than the said models as it primarily focuses on fog computing paradigm while uniquely combining the observations, reputation and other

Table 2
Analysis of trust models in fog.

Research paper	Trust metrics	Trust model	Main contributions
Fog Networking Environment			
[23]	1. Experience 2. Authentication	Fuzzy trust-based model	The model uses a set of modules to authenticate, calculate and choose the most trustworthy node
[24–26]	1. QoS 2. Position 3. Unique identifier	Multiple linear regression model	A model for trustworthy communication using regression analysis and fitting function that relates trust value with the communication variable
[27]	Multi source feedback	Multisource feedback model	A lightweight scheme that uses feedbacks from multiple sources to identify trustworthy IoT edge devices.
[28]	QoS parameters	Broker based model	A broker-based trust evaluation framework based on fuzzy logic

trust metrics necessary in a fog network. It presents a two-way trust system, where both the fog nodes involved in the transaction establish trust before creating a connection. The primary focus of this model is to ensure trustworthy transfer of data in peer-to-peer communication between fog nodes. This model is independent of any third-party involvement which is one of the requirements of the fog network. This reputation system is based on Bayesian inference with discounting factor which gives it a sound mathematical base, we have shortlisted and uniquely combined important network parameters with experiences and recommendations.

3. Proposed trust management framework

3.1. Generic fog system model

Fog has a hierarchical structure with multiple fog nodes capable of performing computations at each level. The Cloud is present at the top having maximum intelligence and resources, The nodes at lower levels are responsible for data collection, whereas higher level nodes carry out complex tasks such as filtering, compression and transformation of raw data. This multilevel infrastructure is responsible for seamless data transfer, processing and filtering closer to the edge of the network as shown in Fig. 1. The nodes at the same level can engage in a variety of activities such as load balancing, data sharing, resilience and fault tolerance [16]. This peer to peer communication lays the foundation for establishing a trust management framework in Fog. A fog node can be represented by any device with enough memory, storage and processing power such as tablets, laptops, smart devices (smart watch, smart cars etc.) DVRs and CCTV cameras etc. [29]. Hence, it is imperative to have both the nodes develop trust before the actual exchange of data takes place.

A generic Fog architecture has three major shareholders; the IoT devices (for data generation), the fog nodes (for data transfer and filtering) and cloud servers (for data processing and storage). This research focuses on establishing trust among the fog nodes for secure transfer of data. For simplicity we consider a single layer fog architecture, without compromising on any of its key features for the proposed TMS. Each fog node is connected to its neighbor and a set of IoT devices based on its location and service type. Hence, Fog nodes can always communicate with their neighboring nodes for trust value exchange. This research uses a simplified single layer fog architecture as depicted in Fig. 2 to build a collaborative trust management system. The following section presents a comprehensive description of the proposed framework, its working environment, case study and the trust metrics required to compute the overall trust of the nodes.

3.2. SQT (Social Qualitative Trust) system model

The proposed solution focuses on the establishment of trust among fog nodes for offloading, data sharing and other services. A fog node present in the geographical range of another fog node will request/avail its services if and only if it satisfies its trust threshold. We propose a two-way trust management scheme that allows both the communicating parties to validate each other before establishing a connection, it prevents rogue nodes from entering the network. Each node maintains a rank which decreases every time it fails to provide a decent service; malicious nodes are removed from the network when their rank falls below the set threshold. The malicious nodes are then blacklisted and broadcast to the whole network to avoid any discrepancy.

Let us suppose a fog node requests a service from its neighboring node, to create a connection, the SP node will validate the authenticity of the SR node before accepting its connection request. To do so SP will ask the neighboring nodes for their recommendations of the SR. It will formulate trust using the beta reputation system, if trust lies above the threshold the SP will accept the connection request otherwise it will reject it. Similarly, SR will also validate the SP on the same lines. In Fog it is important for both the communicating nodes to validate one another beforehand as any device can become a fog node, thereby increasing the attack probability. Therefore, the SR will also validate the SP to ensure authentic service before the actual data transfer takes place. To keep the overhead minimum and maintaining the efficiency of the system, the nodes will only exchange trust values with their immediate neighbors.

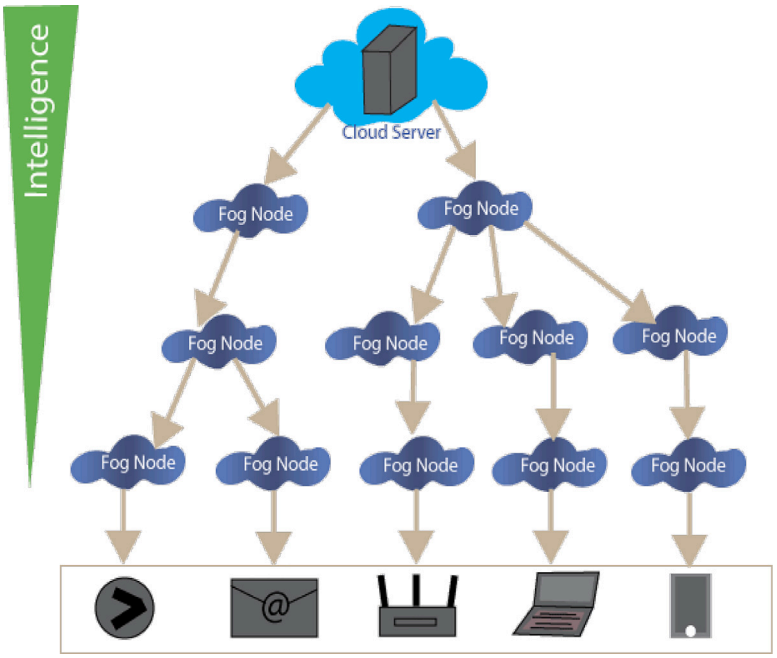

Fig. 1. A multi-layer hierarchical Fog model with Cloud at the top and IoT devices at the bottom. The intelligence of the system decreases from top (Cloud) to bottom (IoT devices).

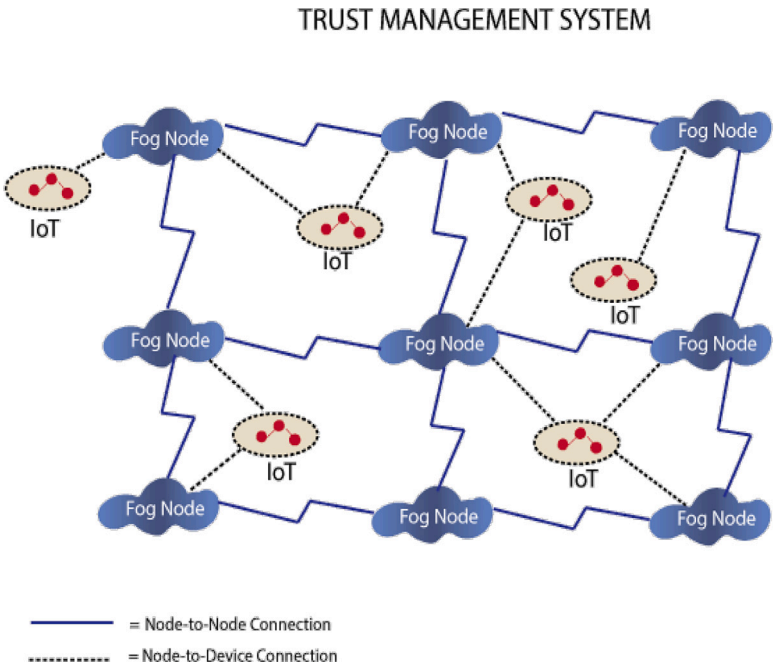

Fig. 2. A simplified single layer fog model with fog nodes connected with their neighboring nodes via node–node connection shown by the straight line. Each node is also connected to a bunch of IoT devices via a node–device connection shown by the dotted line.

The proposed solution divides fog nodes into two main categories; Service Requestor (SR) and Service Provider (SP). If a SR desires a service it will ping its neighboring nodes for service request, one of the neighbors will confirm its availability to become the SP. However, SP must first ensure the authenticity of the SR before proceeding with the service request as a measure for protection against rogue SRs. The SP will calculate a trust value for the SR by consulting with its neighbors and its own experience. The SRs with trust value lower than the acceptable threshold will be deemed untrustworthy and malicious, these malicious nodes will

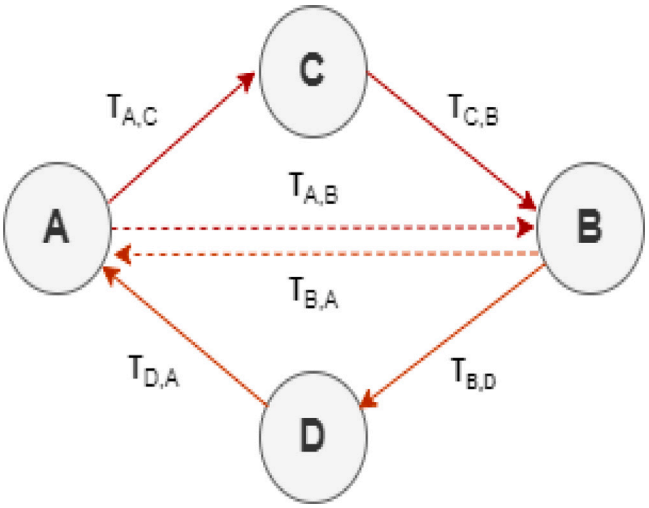

Fig. 3. In a world where node A does not know node B directly it will ask its neighboring nodes (node C in this scenario) to establish its trust value for node B. Similarly, node B will ask Node D to establish trust value of node A.

be refused service immediately. Our model works on a rank system, each node maintains a rank which decreases each time it is detected as a fraudulent node, if the rank of any node becomes zero then it is exiled from the network. This system is in place to allow a protection barrier for legitimate nodes from being exiled from the network after one odd bad service. The SP updates and stores the trust value of any malicious node detected for future reference. Meanwhile, the authenticated SR also wants to ensure the validity of the SP before proceeding with the service request to minimize the probability of receiving a malicious payload in place of the actual service. The process of SP validation is similar to that discussed above, it will compute the trust value of SP by asking its neighbors for recommendations and combining them with its own experience. After both SP and SR have validated one another, a trusted connection is established, and real communication can take place.

4. Bayes model and trust computation

4.1. Bayes model and discounting operation

In a fog network trust is dynamic, subjective, intransitive and asymmetric. Trust computation must yield a dynamic value of trust in conjunction with the trust requirements and the design dimensions [30]. SQT is a distributed system using social and network parameters for trust computation, it follows a multi-trust and event-driven trust approach. Trust is asymmetric and subjective which implies that each node in a network experiences it differently based on limited data, the best way to calculate trust in an architecture such as Fog is by using Bayes Model. The advantage of using Bayes model in Fog is that it allows a node to predict the future behavior of other nodes based on its current observations. It also accurately calculates trust in an unknown and risky environment where the communicating parties have different owners. At the core of Bayes model is beta probability density function, which gives it a sound mathematical base for feedback accumulation. Consider a process that may have y, y′ as its possible outcomes such that γ represents the number of times y is observed and $\hat{\gamma}$ represents the number of times y′ is observed then to observe the future behavior of the process, the parameters of the beta distribution function are set as:

$$\alpha = \gamma + 1 \tag{1}$$

$$\beta = \hat{\gamma} + 1 \tag{2}$$

where $\gamma, \hat{\gamma} \geq 0$. Hence, the probability expectation of the beta distribution is given by:

$$E(p) = \alpha/(\alpha + \beta) \tag{3}$$

Consider four nodes (A,B,C & D), where node A & node B are trustor and trustee, and node C & node D are the recommenders as shown in Fig. 3. Now, by using Eq. (3) the reputation function is expressed as:

$$E(\phi(p|\gamma_{AB}, \hat{\gamma}_{AB})) = \frac{\gamma_{AB} + 1}{\gamma_{AB} + \hat{\gamma}_{AB} + 2} \tag{4}$$

Eq. (4) gives the subjective reputation of node B from node A's perspective, it is not possible to calculate the objective value of reputation as each node has different experiences with B.

4.1.1. Discounting operation

The node calculating trust assigns different weights to the recommendations it receive depending upon its own relation with the recommender nodes, this process is called discounting. If a node does not trust one of the recommender nodes then it can completely ignore its recommendation, similarly it gives more weight to the recommendations of a trusted node as compared to a node with low trust value. This technique helps evade trust-based attacks such as bad mouthing and ballot stuffing attacks. As shown in Fig. 3, let us suppose node A is the trustor gathering reputation of node B on the recommendation of node C, then the beta distribution is given by α_{AB} and β_{AB} as:

$$R_{AB} = \frac{\alpha_{AB}}{\alpha_{AB} + \beta_{AB}} \tag{5}$$

$R_{AB} = R_{AC} \otimes R_{CB}$, where $\otimes$ is called the discounting operator. The α_{AB} and β_{AB} parameters in Eq. (4) are updated as follows:

$$\alpha_{AB} = \alpha_{AB}^{prev} + \frac{2\alpha_{AC}\alpha_{CB}}{[(\beta_{AC}+2)(\alpha_{CB}+\beta_{CB}+2)] + 2\alpha_{AC}} \tag{6}$$

$$\beta_{AB} = \beta_{AB}^{prev} + \frac{2\alpha_{AC}\beta_{CB}}{[(\beta_{AC}+2)(\alpha_{CB}+\beta_{CB}+2)] + 2\alpha_{AC}} \tag{7}$$

Eqs. (6) and (7) represent how the α_{AB} and β_{AB} parameters depend upon the node's trust on the recommender. Discounting helps achieve an unbiased reputation of the trustee node which helps in evading many trust-based attacks.

5. Mathematical description of proposed scheme

5.1. Trust metrics

A trust indicator is an essential part of the trust calculation as it is the property based on which the value of trust is determined, different authors use different trust indicators depending upon their requirements. Generally, more than one parameter is required to build an effective trust management system. Our model considers various social and qualitative trust metrics for its trust calculation; the social metrics, include direct observations made by the trustor as well as the recommendations by the neighboring nodes. This section will briefly discuss the trust metrics of our proposed SQT management system.

5.1.1. Direct trust

The experience of the trustor after a successful transaction with the trustee determines the direct trust. In a trust management system, it is imperative for a node to have the ability to calculate individualistic trust for an unbiased decision. Direct trust holds maximum weightage in our proposed system to decrease the effect of various false recommendation attacks. D_{AB} denotes the trust of node B as calculated by node A for transaction k. The relevance of transaction k is given by transaction factor $tf_{AB}^{k} \in [0,1]$ between the two nodes. The feedback of node B given by node A is represented by $f_{AB}^{k} \in [0,1]$ then the formula for n transactions is given by [17]:

$$D_{AB} = \frac{\sum_{k=1}^{n} tf_{AB}^{k} f_{AB}^{k}}{\sum_{k=1}^{n} f_{AB}^{k}} \tag{8}$$

5.1.2. Reputation function

The reputation metric is of utmost importance when there has been no prior transaction between the trustor and the trustee, in this scenario the trustor greatly depends on the reputation of the trustee in the network. The SQT system model uses Bayesian inference to combine feedback from the recommenders. The simplicity and flexibility of the Bayesian formulation qualifies it as the best approach for this model. The reputation of node B as perceived by node A is given by Eq. (5).

The reputation function is completed by the discounting step that predicts the future behavior of the node B as seen by node A, in this case, based on its past behavior. This protects against the network feedback attacks such as bad mouthing and ballot stuffing attacks. The discounting function is given by Eqs. (6) and (7) as discussed in Section 4

5.1.3. Degree centrality

Degree centrality represents the number of direct connections a node has in a network, higher degree centrality means the node has great importance within the network. The reputation function will be influenced by the degree centrality value of a node, if node B has a higher degree centrality then its recommendation will be higher and vice versa. To minimize the effect of centrality on the reputation function we calculate C_{AB}:

$$C_{AB} = X_{AB} \cap X_{A} \tag{9}$$

where, X_{AB} and X_{A} represent the mutual friends of node A and B and the friends of node A respectively.

5.1.4. Service score

A reward and penalty metric is added to make the SQT system to provide an extra layer of protection against malicious nodes.

$$S_B = \begin{cases} 1 \times wt_s \text{ reward} \\ -1 \times wt_s \text{ penalty} \end{cases} \tag{10}$$

where wt_s represents the weight of the service.

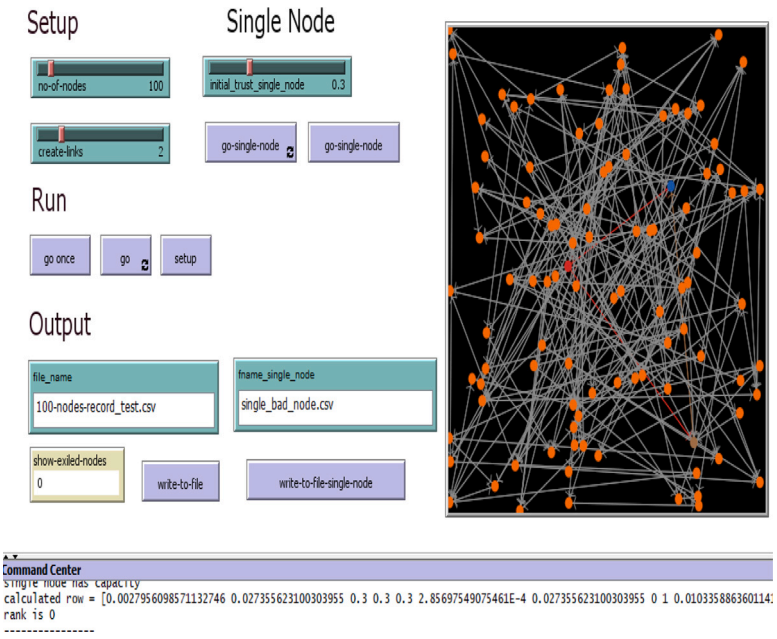

Fig. 4. Initial setup of the simulation. The highlighted nodes are 1-hop neighbors, both blue and red highlighted nodes are calculating trust of one another as shown in Command Center.

5.2. Mathematical model

5.2.1. Bayes Trust

The Bayes model defines trust in terms of collective desirable and undesirable behavior of the trustee as observed by other nodes, it is given by the following equation:

$$ET_B = \frac{\alpha_B + 1}{\alpha_B + \beta_B + 2} \tag{11}$$

where α_B *and* β_B denote the desirable and undesirable behavior of the trustee (node B in this scenario) respectively. Bayes trust showcases the predicted behavior of the trustee, in other words it is the expected behavior of the node, this value might not always be the same as the computed trust as the object which is malicious might become innocent in the future and vice versa.

5.2.2. Calculated trust

The calculated trust is determined by combining all the trust metrics defined by Eqs. (5) (8) (9) (10) in the following sequence:

$$T_{AB} = \delta D_{AB} + \sigma R + \omega C_{AB} + \theta S_B \tag{12}$$

where δ, σ, ω *and* θ are the weights assigned for normalizing the data. The weights assigned are variable as they tend to keep the value of T_{AB} between 0 and 1. We have tested our system against different values of these weights and seen how they affect the trust in the network.

5.2.3. Final trust

The final trust of the node B is calculated by:

$$T_{final} = (ET_B * T_{AB}) - m_e \tag{13}$$

m_e is the marginal error, hence it is taken out from the final equation. The final trust value in our proposed model is given by the product of Bayes trust with the calculated trust. T_{AB} is the calculated trust obtained by aggregating the network trust metrics and

ET_B is the predicted future behavior of the node in light of its calculated trust, for example if the calculated trust $T_{AB} = 0.8$ then the expected behavior of the node is desirable $\therefore ET_B \geq 0.5$.

[N1,N2,N3,N4,...][δ1,σ1,ω1,θ1,δ2,σ2,ω2,θ2] Trusted connection
for *i between* 1 *and* n **do**
  Nb = neighborhood size of i
  for *j between* 1 *and* Nb **do**
    if Ni *has capacity* & $rank \geq 1$ **then**
      */Trust metrics calculation
      $D = \frac{\sum_{k=1}^{n} tf^k f^k}{\sum_{k=1}^{n} f^k}$
      $R = \frac{\alpha}{\alpha+\beta}$
      $C = X_{AB} \cap X_A$
      $S = \begin{cases} 1 \times wt_s \text{ reward} \\ -1 \times wt_s \text{ penalty} \end{cases}$
      */ calculated trust of trustor node (node A)
      $T_{BA} = \delta 2 D_{BA} + \sigma 2 R_{BA} + \omega 2 C_{BA} + \theta 2 S_B$
      for k *between* 1 *and* Nb **do**
        */discounting step
        $T_{BA} = T_{BC} \otimes T_{CA}$
      end
      */ Bayes trust of trustor node (node A)
      $ET_B = \frac{\alpha_A+1}{\alpha_A+\beta_A+2}$
      if $T_{BA} \geq threshold$ **then**
        */calculated trust of trustee node (node B)
        $T_{AB} = \delta 1 D_{AB} + \sigma 1 R_{AB} + \omega 1 C_{AB} + \theta S_B$
        for k *between* 1 *and* Nb **do**
          */discounting step
          $T_{AB} = T_{AC} \otimes T_{CB}$
        end
        */Bayes trust of trustee node (node B)
        $ET_B = \frac{\alpha_B+1}{\alpha_B+\beta_B+2}$
        $T_{final} = (ET_B * T_{AB}) - Me$
        if $T_{final} \geq threshold$ **then**
          Trusted connection is established
        end
      end
    end
  end
end

Algorithm 1: Trust Computation Algorithm

5.3. Algorithm

The detailed working of our proposed model is described in Algorithm (1). The *threshold* for our proposed model is 0.5, but it can be higher for more critical applications.

Node A initiates the communication with node B by sending a service request. The node B computes the trust of node A by running the algorithm, the trust metrics are assigned appropriate weights for calculation of T_{BA}. Lastly, it is combined with the Bayes trust to get the value of final trust. If $T_{BA} \geq threshold_1$ then it will allow the communication to proceed. After getting the connection approval node A will compute the trust of node B, it will follow the same steps; assign weights to the trust metrics, calculate T_{AB} and combine it with Bayes trust to get the final trust. Now, if $T_{final} \geq threshold$ then a trusted connection is established between the two nodes. Our model requires both the SR and SP to establish trust before creating a connection, if either one of nodes fail to establish trust then a trusted connection will not be created and node A will look for other SPs in its neighborhood.

6. Proposed scheme performance and comparative analysis

The proposed system is tried and tested in a simulation environment. The details of simulation parameters, evaluation and performance, and the comparative analysis of the proposed framework is discussed in this section.

Table 3
Simulation parameters.

Parameters	Nodes		
	100	500	800
δ	0.1	0.3	0.6
σ	0.6	0.4	0.1
ω	0.3	0.3	0.3
θ	0.3	0.3	0.3
Threshold	0.4	0.4	0.4
Rank	5	10	15
Capacity	10	10	10
Computation cycle	25	25	25

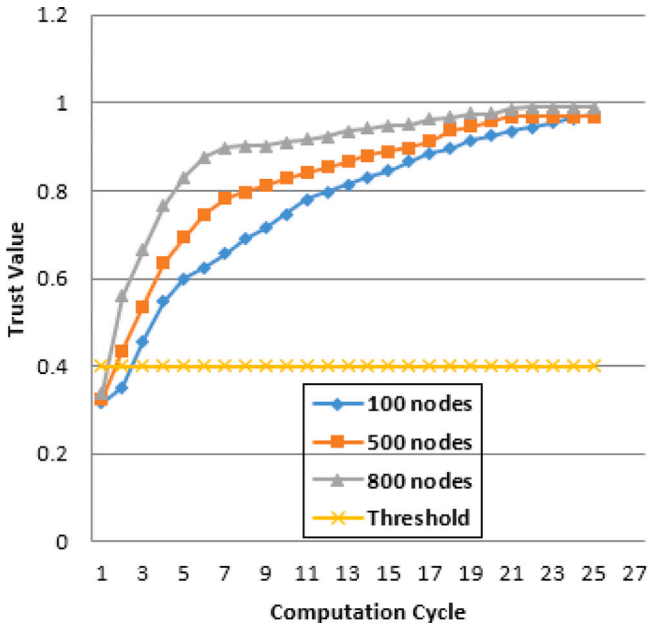

Fig. 5. Trust value of a randomly selected good fog node in a network of 100, 500 and 800 nodes. The value converges around the 6th cycle.

6.1. Simulation setup

A simulation is created in Netlogo v 6.1.1 to evaluate the proposed trust management system for Fog computing shown in Fig. 4. Using Netlogo, we have created a large network (800 nodes), a medium size network (500 nodes) and a small network (100 nodes). The list of simulation parameters and their respective values are given in Table 3.

We have carried out the simulation for 100, 500 and 800 nodes respectively. In our setup any node can be a service requestor or service provider, each node will take recommendations from its 1-hop neighbors only to build reputation rating of the trustee node. The trust update is event-driven, simulation runs for 2 min during which time the trust values are calculated and updated accordingly.

6.2. Evaluation and performance of SQT framework

This section discusses the evaluation and performance of the SQT framework in normal conditions as well as in the presence of malicious nodes. We have carried out the simulation for 100, 500 and 800 nodes respectively, the default simulation parameters of which are presented in Table 3.

Fig. 5 shows the performance of a random node in the proposed framework under normal circumstances, it exhibits the accuracy and convergence of the trust value in networks with 100, 500 and 800 nodes. The trust value converges quickly with more accuracy when more weight is given to the direct observations as in case of 800 nodes network. Where as trust value converges later in the computation cycle when more weight is assigned to indirect observations.

The SQT trust model is simulated in Netlogo v 6.1.1, the code of which is available on github (https://github.com/MH9196/FogTrustModel/blob/main/directed_graph_network_with_single_node.nlogo). The performance analysis of a randomly selected bad node is shown in Fig. 6. The algorithm is designed to penalize a node on bad service, this makes it twice as hard for the node to

Table 4
Comparison of SIoT trust model with the proposed SQT model.

Contribution	Two-way trust	Distributed approach	Resilient against on off attack	Resilient against other trust-based attacks (SPA, BSA, BMA, OSA)	Low computation cost
Kowshalya	✗	✗	✓	✗	✓
TMCoI-SIoT	✗	✗	✓	✗	✓
SQT	✓	✓	✓	✓	✓

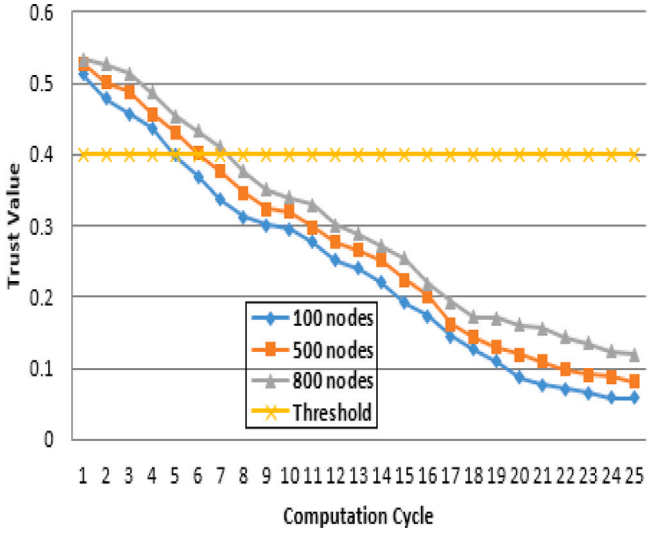

Fig. 6. Trust value of a randomly selected bad fog node in a network of 100, 500 and 800 nodes. The trust value declines sharply which makes it hard for a bad node to carry out attacks.

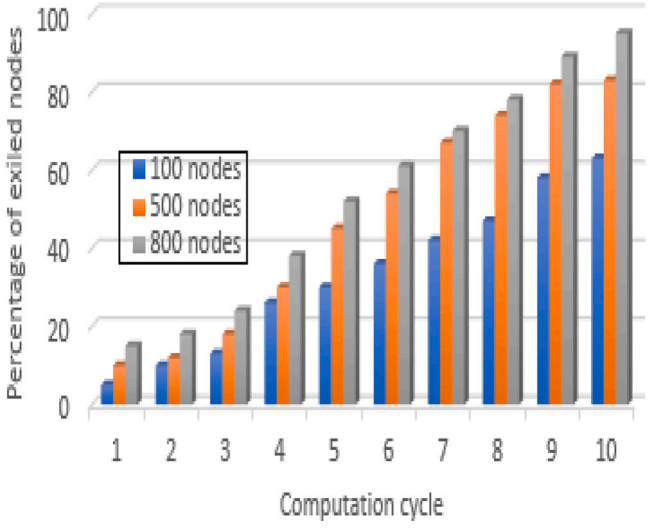

Fig. 7. Percentage of exiled nodes over the course of time in a network of 100, 500 and 800 nodes.

recover its reputation. Each node maintains a rank which decreases whenever it behaves undesirably, if the rank of a node becomes zero it is eventually kicked out of the network. The percentage of the exiled nodes from the network are given in Fig. 7.

6.3. Comparative analysis

Due to limited work available on the subject, we carried out the comparative analysis with the model introduced by A.M. Kowshalya et al. [17] and TMCoI-SIoT [19] with the proposed SQT model against on off attack. All the models can detect on off

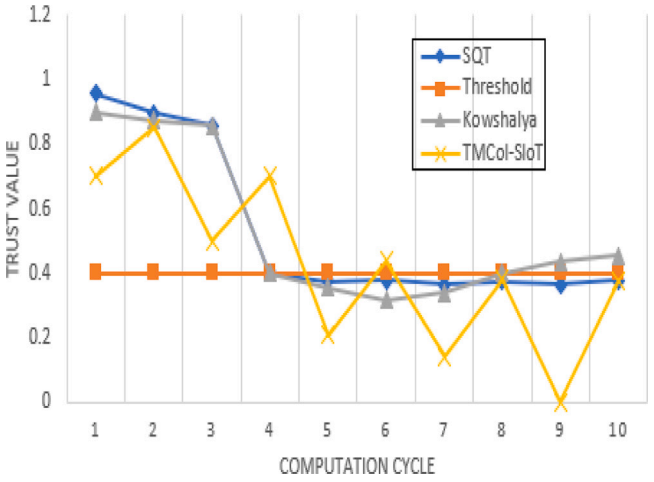

Fig. 8. Comparison of SQT system model with two existing SIoT models, Kowshalya and TMCoI-SIoT in the presence of on off selective forwarding attack.

attack, but the proposed SQT model can not only identify but make it hard for the malicious node to recover its reputation as shown in Fig. 8. An in depth comparison of the models is given in Table 4. The proposed model is built on a distributed approach, which fits very well with the structure of a Fog computing environment, each node is capable of calculating trust and circulating it among its peers. Whereas, the existing models have a centralized entity that is either elected or permanently exists to calculate the value of trust and circulate it among the network. The proposed SQT model is a two-way trust approach, it requires both communicating nodes to validate each other before connecting. Moreover, SQT is also resilient against other attacks such as:

1. Self-promotion attack(SPA) as it does not allow any node to self-recommend
2. Bad-mouthing attack(BMA) as it only considers recommendations from trusted neighbors
3. Ballot-stuffing attack(BSA) due to weighted recommendations
4. Opportunistic service attack(OSA) as it eliminates nodes with inconsistent behavior over time

7. Conclusion and future work

This research aims to define a trust management system for fog computing environments that can mitigate its prevalent security issues. The existing techniques lack the necessary requirements of trust in a fog environment. In this paper, a peer-to-peer trust management system is proposed that enables fog nodes to develop trust before connecting with others. This system prevents fog nodes from making untrustworthy connections and increases the probability of malicious node detection at earlier stages. The system calculates trust and predicts the future behavior of a node through Bayesian Inference. The system is evaluated where the behavior of a single node is observed, it can be seen that a single good node converges to 1 quickly and a single bad node is detected in the earlier cycles, its trust value decreases and falls below the threshold. The system is also resilient towards trust-based network attacks, it detects them quite early and expels the malicious nodes over continued bad behavior. The system can be improved to accommodate location awareness of the nodes.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

[1] Thomas Barnett J. Cisco global cloud index 2015–2020. 2020, https://www.cisco.com/c/dam/m/en_us/service-provider/ciscoknowledgenetwork/files/622_11_15-16-Cisco_GCI_CKN_2015-2020_AMER_EMEAR_NOV2016.pdf. [Accessed 1 June 2020].
[2] Abi Sen AA, Yamin M. Advantages of using fog in IoT applications. Int J Inf Technol 2020;1–9.
[3] Iqbal W, Abbas H, Daneshmand M, Rauf B, Bangash YA. An in-depth analysis of IoT security requirements, challenges, and their countermeasures via software-defined security. IEEE Internet Things J 2020;7(10):10250–76.

[4] Jahantigh MN, Rahmani AM, Navimirour NJ, Rezaee A. Integration of Internet of Things and cloud computing: a systematic survey. IET Commun 2019;14(2):165–76.
[5] Roman R, Lopez J, Mambo M. Mobile edge computing, fog et al.: A survey and analysis of security threats and challenges. Future Gener Comput Syst 2018;78:680–98.
[6] Marín-Tordera E, Masip-Bruin X, García-Almiñana J, Jukan A, Ren G-J, Zhu J. Do we all really know what a fog node is? Current trends towards an open definition. Comput Commun 2017;109:117–30.
[7] Mäkitalo N, Ometov A, Kannisto J, Andreev S, Koucheryavy Y, Mikkonen T. Safe and secure execution at the network edge: a framework for coordinating cloud, fog, and edge. IEEE Softw 2018;35(1):30–7.
[8] Rahmani AM, Gia TN, Negash B, Anzanpour A, Azimi I, Jiang M, Liljeberg P. Exploiting smart e-Health gateways at the edge of healthcare Internet-of-Things: A fog computing approach. Future Gener Comput Syst 2018;78:641–58.
[9] Alemneh E, Senouci S-M, Brunet P. PV-Alert: A fog-based architecture for safeguarding vulnerable road users. In: 2017 global information infrastructure and networking symposium. IEEE; 2017, p. 9–15.
[10] OpenFog consortium: The technology of OpenFog computing and networking. 2020, https://site.ieee.org/denver-com/files/2017/06/OpenFog-Consortium-Reference-Architecture-Summary-presentation-for-Denver-Summit.pdf. [Accessed 1 June 2020].
[11] Patwary AA-N, Naha RK, Garg S, Battula SK, Patwary MAK, Aghasian E, Amin MB, Mahanti A, Gong M. Towards secure fog computing: A survey on trust management, privacy, authentication, threats and access control. Electronics 2021;10(10):1171.
[12] Hao Z, Novak E, Yi S, Li Q. Challenges and software architecture for fog computing. IEEE Internet Comput 2017;21(2):44–53.
[13] Alrawais A, Alhothaily A, Hu C, Cheng X. Fog computing for the internet of things: Security and privacy issues. IEEE Internet Comput 2017;21(2):34–42.
[14] Mukherjee M, Matam R, Shu L, Maglaras L, Ferrag MA, Choudhury N, Kumar V. Security and privacy in fog computing: Challenges. IEEE Access 2017;5:19293–304.
[15] Sabireen H, Neelanarayanan V. A review on fog computing: architecture, fog with IoT, algorithms and research challenges. Ict Express 2021;7(2):162–76.
[16] Dybedokken TS. Trust management in fog computing (Master's thesis), NTNU; 2017.
[17] Kowshalya AM, Valarmathi M. Trust management for reliable decision making among social objects in the Social Internet of Things. IET Netw 2017;6(4):75–80.
[18] Xiao H, Sidhu N, Christianson B. Guarantor and reputation based trust model for social internet of things. In: 2015 international wireless communications and mobile computing conference. IEEE; 2015, p. 600–5.
[19] Abderrahim OB, Elhdhili MH, Saidane L. TMCoI-SIOT: A trust management system based on communities of interest for the social Internet of Things. In: 2017 13th international wireless communications and mobile computing conference. IEEE; 2017, p. 747–52.
[20] Abderrahim OB, Elhedhili MH, Saidane L. CTMS-SIOT: A context-based trust management system for the social Internet of Things. In: 2017 13th international wireless communications and mobile computing conference. IEEE; 2017, p. 1903–8.
[21] Truong NB, Um T-W, Lee GM. A reputation and knowledge based trust service platform for trustworthy social internet of things. In: Innovations in clouds, internet and networks. 2016.
[22] Truong NB, Um T-W, Zhou B, Lee GM. From personal experience to global reputation for trust evaluation in the social internet of things. In: GLOBECOM 2017-2017 IEEE global communications conference. IEEE; 2017, p. 1–7.
[23] Soleymani SA, Abdullah AH, Zareei M, Anisi MH, Vargas-Rosales C, Khan MK, Goudarzi S. A secure trust model based on fuzzy logic in vehicular ad hoc networks with fog computing. IEEE Access 2017;5:15619–29.
[24] Wang T, Li Y, Chen Y, Tian H, Cai Y, Jia W, Wang B. Fog-based evaluation approach for trustworthy communication in sensor-cloud system. IEEE Commun Lett 2017;21(11):2532–5.
[25] Wang T, Lu Y, Cao Z, Shu L, Zheng X, Liu A, Xie M. When sensor-cloud meets mobile edge computing. Sensors 2019;19(23):5324.
[26] Wang T, Lu Y, Cao Z, Shu L, Zheng X, Liu A, Xie M. When sensor-cloud meets mobile edge computing. Sensors 2019;19(23):5324.
[27] Yuan J, Li X. A reliable and lightweight trust computing mechanism for IoT edge devices based on multi-source feedback information fusion. IEEE Access 2018;6:23626–38.
[28] Rahman FH, Au T-W, Newaz SS, Suhaili WS, Lee GM. Find my trustworthy fogs: A fuzzy-based trust evaluation framework. Future Gener Comput Syst 2020;109:562–72.
[29] Wazid M, Das AK, Kumar N, Vasilakos AV. Design of secure key management and user authentication scheme for fog computing services. Future Gener Comput Syst 2019;91:475–92.
[30] Guo J, Chen R, Tsai JJ. A survey of trust computation models for service management in internet of things systems. Comput Commun 2017;97:1–14.

Mahnoor Hamza is an Electrical Engineer majoring in telecommunication. She completed her post graduation in Information Security from National University of Sciences and Technology (NUST). Currently pursuing a career in Governance, Risk, and Compliance(GRC) domain. Her most favorite niche includes securing the Internet of Things, Cloud computing, Fog Computing Environment, and Edge Computing. She is passionate about spreading awareness regarding how exploitative and insecure networks and systems can be.

Waseem Iqbal has received his Ph.D. degree from National University of Sciences and Technology (NUST) in 2021. He has authored over 67 scientific research articles in prestigious international journals (ISI-Indexed) like IoTJ, FGCS, Systems Journal, MTAP, IEEE Sensors, ACM Computing Surveys, and IEEE Comm Surveys & Tutorials, etc., along with reputed conferences like ICC etc. Dr. Waseem has conducted more than 15 CEH, CHFI, CSCU, and Forensics practical hands-on workshops for industry and public.

Awais Ahmad received his Ph.D. in Computer Science and Engineering from Kyungpook National University, Daegu, Korea. Dr Awais has published more than 150 International Journals (Cumulative Impact Factor: 260+)/Conferences/Book Chapters in various reputed IEEE Transactions, IEEE Magazines, ACM Transactions, Elsevier, and Springer Journals, whereas in leading conferences, i.e., IEEE GLOBECOM, IEEE INFOCOM, IEEE LCN, and IEEE ICC, respectively. Dr. Awais is also serving as Guest editor in various Elsevier and Springer Journals, including Future Generation Computer Systems (Elsevier), Sustainable City and Societies (Elsevier), and Computational Intelligence and Complexity (Springer), Multimedia Tools and Applications (Springer), IEEE Access, and Real-Time Image Processing Journal (Springer).

Muhammad Babar received his Ph.D. degree in Computer Software Engineering from the National University Sciences and Technology (NUST), Islamabad, Pakistan in 2018. His research area includes but is not limited to Big Data Analytics, Machine Learning, the Internet of Things (IoT), Smart City Design and Planning, Security, and Social Web of Things (SWOT). He has published his research work in various IEEE, Elsevier, and Springer journals. He is an active reviewer of Elsevier FGCS, Springer MONET, Elsevier Computer Networks, IEEE Access, Elsevier Heliyon, and Wiley Transactions on Emerging Telecommunications Technologies.

Sohaib Khan did his bachelor's in Telecommunication engineering and master's in Information Security from the National University of Sciences and Technology, Pakistan. He has been teaching various core subjects related to information security since 2017. His active area of interest includes computer security, Network Security, Digital forensics, and Cryptography. Apart from research activities he has also conducted various practical workshops related to vulnerability assessment and penetration testing.