Wireless Personal Communications (2021) 119:1369–1386
https://doi.org/10.1007/s11277-021-08285-7

Trust Enforced Computational Offloading for Health Care Applications in Fog Computing

V. Meena[1] · Meghana Gorripatti[1] · T. Suriya Praba[1]

Accepted: 8 February 2021 / Published online: 6 April 2021

Abstract
Internet of Things (IoT) is a network of internet connected devices that generates huge amount of data every day. The usage of IoT devices such as smart wearables, smart phones, smart cities are increasing in the linear scale. Health care is one of the primary applications today that uses IoT devices. Data generated in this application may need computation, storage and data analytics operations which requires resourceful environment for remote patient health monitoring. The data related with health care applications are primarily private and should be readily available to the users. Enforcing these two constraints in cloud environment is a hard task. Fog computing is an emergent architecture for providing computation, storage, control and network services within user's proximity. To handle private data, the processing elements should be trustable entities in Fog environment. In this paper we propose novel Trust Enforced computation ofFLoading technique for trust worthy applications using fOg computiNg (TEFLON). The proposed system comprises of two algorithms namely optimal service offloader and trust assessment for addressing security and trust issues with reduced response time. And the simulation results show that proposed TEFLON framework improves success rate of fog collaboration with reduced average latency for delay sensitive applications and ensures trust for trustworthy applications.

Keywords Fog computing · Computational offloading · Trusted node identification · Health care · Response time

1 Introduction

The current generation of the computing world is driven by the internet enabled things connected with one another in the real world which is popularly called as Internet of Things (IoT). An analyst firm [1] forecasts that 5th generation IoT may reach 50 million connected objects by 2023.The IoT objects such as smart sensors, smart phones, smart wearables, etc., are capable of generating vast amount of data. International Data Corporation (IDC) forecasts that IoT devices may generate 79.4ZB of data in 2025. These IoT objects are just data

✉ T. Suriya Praba
suriyapraba@cse.sastra.edu

[1] School of Computing, SASTRA Deemed University, Thirumalaisamudram, Thanjavur 613401, India

generating sources but are not capable for performing computations. We need a resourceful environment to handle the computationally intensive and storage intensive tasks. One such solution for the issue is cloud computing.

Cloud computing is considered as network of resourceful servers that are remotely located. It provides on-demand facilities for computation, storage, infrastructure, platform, etc., with pay as you go policy. Cloud computing enables the IoT devices to outsource the massive data and allows to perform computations efficiently. Since Cloud computing servers are geographically dispersed, the jobs submitted to the cloud servers may get increased latency even for the jobs that requires lesser processing time. One solution to process IoT generated data efficiently with reduced latency is Fog computing.

Fog computing is an emergent architecture of large-scale distributed systems at the edge of the user premises. Fog is a resourceful environment that puts substantial facilities of cloud at network edge. It eliminates dedicated bandwidth requirement issue of centralized cloud infrastructure service. The overall service latency is reduced in Fog computing as responsible Fog service providers are designated nearby the data sources. Whereas the cloud infrastructure may be located far away from the data source. As a result, Fog nodes are easily overloaded with bunch of unleashed requests from IoT devices. So appropriate load balancing techniques may further increase Fog efficiency.

In Fog computing environment different vendors are involving for providing Fog-based services for various reasons. For example, most of the cloud service providers are bringing their services in the edge of the user premises for better performance. On the other hand, private cloud owners may lease their unused resources to the local businesses. Also, the Fog computing environment may be influenced by the various internet service providers or the wireless carriers used for communication. This flexibility and various parties' involvement in Fog environment upsurges another issue called Requirement of Trust (RoT) among Fog nodes. The following scenario (Fig. 1) gives an insight to the blend of emerging M2M technologies such as Fog and cloud computing for various IoT based applications.

Nowadays due to the drastic increase of aging population, chronic diseases and pandemic diseases such as COVID-19, most of the countries faces huge number of challenges. One among that is shortage of nursing staffs and healthcare professionals also the cost should be reduced while providing high quality service to the patients. Well known solution for this issue is to reduce manual supervision and manual patient monitoring. It can be an automated supervision and remote patient monitoring respectively. This may be achieved with the help of emerging IoT devices such as wearable low-cost data acquisition sensory devises. But trust is the most wanted requirement in these health care applications because of the sensitivity of patients personal and health care related data. Recent advancements in M2M technologies such as Fog computing can smartly fit into these kinds of trust worthy IoT based applications for rapid data processing and trust maintenances.

In this paper we propose Trust enforced computational offloading for health care applications in Fog computing (TEFLON). Ubiquitous IoT devices collects data from the destined input sources which needs to be further processed by resourceful environment. As the principle requirement here is to service the requester with rapid output whereas the processing entities should be trusted for the applications handling with sensitive data. We propose an efficient framework to enforce trust in the Fog environment. This also provide service with reduced response time by imposing parallelism in the Fog environment.

The organization of this paper is structured as follows: Sect. 2 elaborates related work associated with the proposed idea. Section 3 elaborates motivation behind this work and contribution. Section 4 explains proposed work with algorithm. Results and discussions were given in Sect. 5. Finally, Sect. 6 concludes the work with future directions.

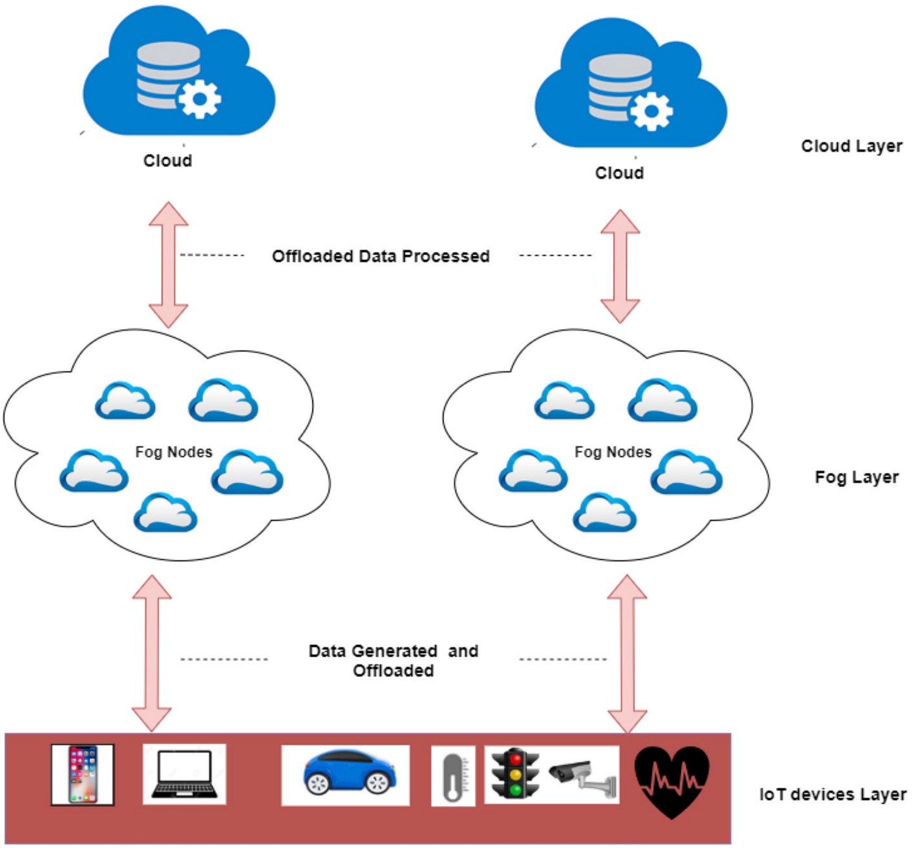


Fig. 1 IoT Layers

2 Related Work

In recent years the emergence of Internet of Things (IoT) devices and ubiquitous applications are increasing in linear scale. Nevertheless, the growth of communication technologies is also in the same rapidity to cope up with the growing needs, some of the delay sensitive and trustworthy applications needs exclusive techniques. Trust plays a vital role in locating malicious entities which are acting as a legal in the network. This section elaborates the related works of computational offloading for trustworthy and delay sensitive applications using Fog computing.

In [2] authors proposed a commitment approach to assess the trust for the nodes involved in the Fog collaboration with the help of direct and indirect trust assessments. They used quality of protection and quality of services for their assessments. They isolated the malicious node and also reduces the response time by 15 s. In [3] authors have proposed Fog based hierarchical trust system to solve trust deficiencies in cyber security. They used two parts of structure for trust assessment. The one is behavior monitoring part in wireless sensor layer and the another one is called data analysis part which is relying in Fog layer. The structure being formed was used to find trust between cloud service providers and sensor service providers. Proposed hierarchical structure saves network energy, enables rapid detection of malicious nodes in acceptable delay. In [4] the authors has made a survey of the Fog computation architecture and report the threat and trust issues. And also, they analyzed the existing methods for solving those issues and highlighted the challenges and future directions. A novel trust and reputation-based model [TRFIoT] [5] was proposed to farm out the malicious nodes involving in Fog layer with the help of multi-source trust evaluation system. To make the trustworthy and reliable system the authors have used trust feedback and periodic trust feedback systems. In [6] authors have explained the importance of task offloading in various computing paradigms such as Fog, edge and cloud. That offloading is performed based on various resource constraint requirements such as computationally intensive, energy conservation and latency management. Authors have also analyzed various key technologies available for offloading in Fog computing.

In [7] authors have implemented an artificial intelligence algorithm to present trust management architecture. It gives applications with improved quality of services, security and privacy with low cost. This architecture uses block chain based smart contracts and it is executed on Ethereum ledger. The proposed architecture is used to manage the trust on camera, dataflow and node selection. A load balancing technique on edge data centers [8] was proposed to provide better load balancing by identifying the idle edge data centers. The load balancing achieves optimized value for response time and resource utilization. It also provides security by incorporating authentication on destination edge data centers. In [9] authors described the way to rate the reputation by considering beta probability density functions. In that function reputation rating is derived by combining various feedbacks.

A novel small cell base stations coalition algorithm [10] is proposed to share the computation resources among Small cell Base Stations(SBS). Authors also construct social trust network, in order to defend against various security attacks possible in SBS collaboration. The proposed system achieves computing performance by exploiting ultra-dense deployment of SBSs. In [11] authors have designed trust management protocol to provide secure routing in delay tolerant network. It finds the operational settings at runtime with respect to dynamic network environments. Authors find the impact gain in delivery ratio by maintaining the tradeoff between message overhead and message delay. In [12] authors have proposed trust management model based on collaborative filtering. This uses direct and indirect trust to minimize convergence time and also strongly defend against collusion attacks. They support service composition applications based on Service Oriented Architecture (SOA)—based IoT systems. Performance analysis of proposed system is assessed against Peer Trust and Eigen Trust environments.

In [13] authors have introduced Fog layer between mobile user and cloud environment. The optimal workload allocation among cloud and Fog decreases transmission delay (latency) with minimal power consumption. Authors also have investigated trade-off between energy consumption and transmission delay. In [14] authors have proposed the framework for offloading the computational intensive jobs by considering the elastic and fixed CPU frequency. Algorithms are implemented on the basis of linear relaxation-based approach and semidefinite relaxation-based approach for fixed CPU frequency. And exhaustive search-based approach is used for elastic type. Proposed method minimizes the energy consumption and total execution latency. In [15] author proposed the fuzzy trust model in order to handle the inaccurate, incomplete information in VANET due to interference occurred in movable and immovable objects. Proposed model is constructed based on experience to efficiently find faulty nodes, malicious attackers experience and also correctness of the data. In [16] authors have proposed tensor based cloud edge computing model. The service tensors are also used here for supporting large scale environments. For small scale data, the processing is carried out in edge plane.

The following table (Table 1) illustrates the comparative analysis for TEFLON with existing research works.

3 Background and Motivation

The emergence of ubiquitous computing and embedded systems have attracted most of the applications on its scope. In the twenty-first century researchers are focusing to bring down the computing resources at the edge of the network for rapid response. Lot of M2M technologies have been emerged and Fog computing is still an open research area. Achieving trust in Fog environment is also a notable issue since plenty of applications are running with its sensitive data. These key requirements rise the motivation to create a trust enforced load balanced Fog environment for trust worthy applications provided with reduced response time. Following subsections elaborates background and motivation.

3.1 Fog Architecture

Fog architecture follows traditional distributed computing architecture which may be designed to service for either be specific to application or nonspecific. Currently there is no universally acceptable architecture available for Fog computing. In this proposed work we adapted a well-known Fog architecture which is already described in [17]. Introducing Fog layer in-between IoT and cloud layers' aids to gain a good vision of benefits and functionalities. Main stratums of adopted architecture are IoT device layer, Fog layer and cloud layer which are described further.

3.1.1 IoT Devices Layer

This is the initial layer and is responsible for generating and transmitting generated data to the next level layer. This layer consists of large number of diverse IoT devices (i.e., wearable sensory devices, mobile devices, etc.,) which is playing the major role of data generation. These devices are communicating to the next level layer called Fog layer using popular communication protocols such as MQTT, CoAP, XMPP, DDS, AMQP, Bluetooth, etc., The data generated are being sent to the resourceful Fog environment by the source for their inputs getting processed.

3.1.2 Fog Layer

This is a resourceful layer comparing to the previous layer which comprises of number of distributed nodes that are located in various vendor specific locations. Each Fog node is comprised of computational capabilities, data storage and communication facility. The key contribution of this layer is processing of data received from IoT devices by utilizing its capability. Primary focus of Fog computing is to improve the efficiency of IoT services by reducing the request response time for the latency aware applications and minimizing the communication and processing load over the cloud. Fog nodes acts as bridge between IoT devices and cloud.

Table 1 Comparative analysis for TEFLON

Authors	Trust	Latency	Load balancing	Parallelism
Mohammed Al et al. [2]	Direct and indirect trust used	High	Low	Not addressed
Tian Wang et al. [3]	Hidden attacks detected	Moderate	Low	Not addressed
Yasir Hussain and Zhiqiu Huang[5]	Trust and Reputation based model used	Low	Low	Not addressed
Deepak Puthal et al. [8]	Authentication is provided on edge centers	High	High	Not addressed
Audun Josang [9]	Beta probability density function used	Very low	Load	Not addressed
Lixing Chen et al. [10]	SBS based collaboration used	Low	High	Achieved at low level
Ing-Ray Chen et al. [11]	Direct, indirect, healthiness and selfishness of Fog nodes uesd	High	Low	Not addressed
Ruilong Deng et al. [13]	Not addressed	High	High	Achieved in Moderate level
Thinh Quang Dinh et al. [14]	Not addressed	High	High	Achieved in Moderate level
TEFLON	Direct and indirect trust calculated by considering various parameters	High	High	Achieved the Micro level Parallelism

3.1.3 Cloud Layer

This is the top most layer in the architecture. It enables the architecture to access the cloud resources in efficiently. Whereas, cloud layer is highly resourceful environment comparing to the Fog layer which may do big data processing, computations that Fog cannot handle.

3.2 RoT Applications (Requirement of Trust)

Emergence embedded systems and mobility have increased the usage of IoT devices in the past decade. So, these tiny IoT devices are forced to handle all of the applications including trust worthy applications. But at the same time these devices cannot handle computationally intensive tasks because of its resource scarcity. So, the task has to be offloaded to the nearby resourceful environment, which should be highly trusted at the same time for trustworthy applications. Some of the examples include health care related applications which are handling sensitive data, financial applications handling transactions, military applications, etc.

3.3 Threats and Security Attacks in Fog

Fog computing environment is highly vulnerable due to involvement of multiple vendors such as owners of private cloud (lease providers) and various Internet Service Providers (ISPs). Fog network consist of number of Fog nodes in which reliability should be ensured for Fog-to-Fog collaboration. This may be affected by purposefully deployed malicious nodes through which it is attacked. The following are the possible attacks in the Fog environment.

3.3.1 Denial of Service

This kind of attacks makes intended users unable to access Fog service, by flooding Fog nodes with superfluous request. Legitimate users cannot avail the services since Fog nodes are overloaded with unwanted requests. Also, it generates high traffic in the network and consumes valuable network bandwidth resource. This attack creates disrupts for Fog-to-Fog collaboration and most importantly due to limited resource Fog environment is more vulnerable than cloud.

3.3.2 Jammers and Spam Generation

In this attack unwanted fake data are generated in large amount by malicious Fog nodes. The generated bogus data floods the network and consumes bandwidth and makes legitimate Fog nodes underutilized. So, the Fog network is always busy with serving fake requests whereas intended users are waiting for the service.

3.3.3 Impersonation

In this attack a malicious Fog node act as an authentic node but provides fake services to the legitimate users. This fake node breaches the privacy policy and hacks data such as user credentials by providing phishing services.

3.3.4 Tampering Fog Node

In this attack Fog nodes are tampered and becomes malicious node which creates transmission delay, allowing attackers to modify data and packet dropping. But here this is very difficult to identify those tampered nodes because the effects generated by this tampered node may be caused by some other common drawbacks of the communication networks such as unstable channel conditions, etc.

Achieving trust for the applications running with sensitive data and reducing latency are the two major key factors for successful Fog networking. These two requirements are addressed in the proposed framework trust enforced computational offloading for health care applications (TEFLON) with the following contributions:

1. Load balancing and parallelism are carried in an efficient way by a novel optimal service offloader algorithm. Also, it avoids micro level parallelism to make service composition as simple at the end.
2. Most of the computationally intensive tasks submitted to the Fog environment has reduced response time by load balancing and parallelism.
3. Trust is ensured in the Fog environment by trust assessment algorithm which calculates trust by direct and recommended trust values. So that trust worthy applications running in end devices are only offloaded to the highly trusted nodes.
4. As the Fog environment is trust enforced so service requests carried in that are highly secure
5. Even for trust assessment, trusted Fog nodes recommendations are taken into considerations so that possible penetration of malicious nodes is highly restricted.

4 Proposed Trust Enforced Computational Offloading for Healthcare Applications

In twenty first century number of IoT devices and IoT based applications are increasing in the exponential scale. But those devices cannot handle computationally intensive and storage intensive task due to its limited resources. So, such applications running in those devices have to be offloaded. The tasks usually be offloaded with cloud (Public or private clouds). One of the major issues faced with cloud services is latency due to distance and bandwidth limitations. Usage of Fog services addresses the above-mentioned issue by providing services in user premises. Fog computing is the extension of cloud computing and provides computation, storage and networking services in the edge of the end device. But most of the applications running in end IoT devices today are trustworthy applications which are handling sensitive data.

Proposed framework ensures trust and reduces response time by two novel algorithms namely trust assessment and optimal service offloader respectively. Following Sect. 4.1 describes about the optimal service offloader algorithm. Then trust assessment part is described in Sect. 4.2 to assess trust by calculating direct and recommended trust.

4.1 Optimal Service Offloader

Task offloaded to the Fog network by a service requester may superfluous the Fog nodes. Fog network consists of number of Fog nodes in which the service request may be distributed. Notations used in the algorithm are given in Table 2.

Algorithm 1 describes service offloading for the Fog environment to offload service to the best fit Fog node.

Table 2 Notations used

S. No	Notations	Description
1	*F_x*	Fog node X
2	*HTL_TABLE*	Highest Trust Level
3	*rR*	Required resources
4	*aR*	Available resources
5	*NFU*	Number of Fog used
6	*F_i*	ith Fog node
7	*P_rate*	Packet loss rate
8	*cT*	Computationally intensive task
9	*MX_allowed*	Maximum nodes allowed for load distribution
10	*Adj*	Adjacency Factor
11	*W1, W2, W3, W4*	Weight constants
12	*T_direct*	Direct trust value
13	*T_rec*	Recommended Trust value
14	*T_delay*	Vulnerable delay
15	*T_max*	Maximum allowable time to forward request
16	*T_history*	Previous trust vale
17	*T_newTrust*	Total trust value for node Fog_i
18	*REC_TABLE*	Recommended trust table used for recommended trust assessment

Algorithm 1: Optimal offloader

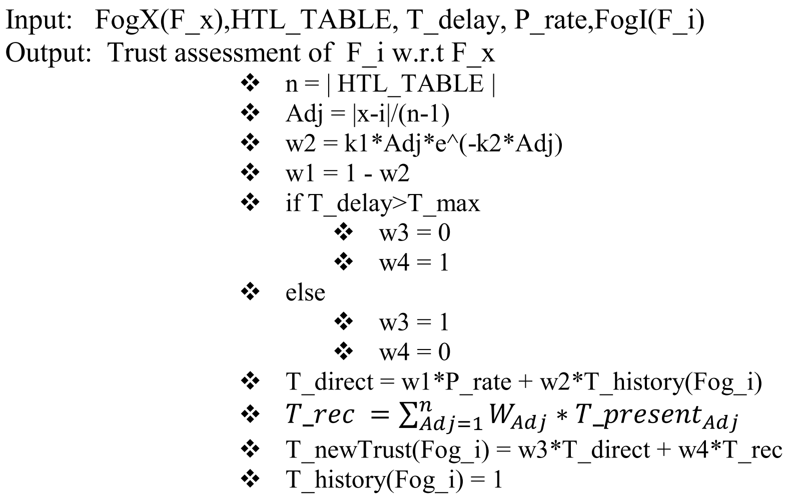
```
Input:  $FogX(F_x), HTL_{TABLE}, RequiredResources\ (rR)$
Output: $offloadingserviceswithbestfitfognode$
    ❖ $aR = aR_{in_{F_x}}$
    ❖ $if aR \geq rR$
        ❖ $assigncTtoF_x$
        ❖ $returntrue$
    ❖ $Label : StartTime = time()$
            ❖ $INDEX = 1$            //index in HTL Table
            ❖ NFU = 0
            ❖ while not end of HTL_TABLE
                ❖ F_i = Fog in HTL_TABLE[INDEX]
                ❖ aR = aR_in_F_i
                ❖ if aR>= 0.5rR
                    ❖ rR = rR - min(aR,rR)
                    ❖ NFU = NFU + 1
                    ❖ assign cT to F_i
                    ❖ TrustAssessment(F_i, time()-startTime(),
                      P_rate, F_x)
                ❖ if rR == 0
                    ❖ return true
                ❖ if NFU == MX_allowed
                    ❖ wait()
                    ❖ assign cT to F_x
            ❖ if end of HTL_TABLE
                ❖ wait()
                ❖ go to Label
```

When the applications running in the end devices are computationally intensive (cT), the devices have to offload service to the nearby Fog node known as initial Fog node (*F_x*). Then service offloader extracts the resource requirement (*rR*) from the request (*cT*) submitted. It is checked with the currently available resources (aR). If submitted service request needs more resource than current availability then it refers HTL_TABLE (Highest Trust Level) to find the Fog node which is highly trustable Fog node (*F_i*) else request (cT) will be completed by (*F_x*). Then (*F_x*) sends query to (*F_i*) to know its current availability. If it is capable of serving the request (cT) with more than fifty percentage of required resources then request (*cT*) will be assigned to (*F_i*) and the value of NFU will be incremented also the trust assessment algorithm will be invoked. Else next index in HTL_TABLE will be chosen for further processing. In optimal service offloader the distribution of service request (cT) will be carried out by considering fifty percentage of availability so that load balancing and task parallelism are ensured and the results obtained for the health care dataset were discussed in Sect. 5. In this algorithm micro level parallelism is avoided (line no: 18) to reduce the complexity of service composition. And importantly the request (cT) will be distributed only for highly trustable Fog nodes.

4.2 Trust Assessment

The major issue in Fog collaboration is to identify trustable neighboring Fog entities properly to offload computationally intensive and trust aware service request.

Algorithm 2: Trust Assessment



```
Input:  FogX(F_x),HTL_TABLE, T_delay, P_rate,FogI(F_i)
Output: Trust assessment of F_i w.r.t F_x
    ❖ n = | HTL_TABLE |
    ❖ Adj = |x-i|/(n-1)
    ❖ w2 = k1*Adj*e^(-k2*Adj)
    ❖ w1 = 1 - w2
    ❖ if T_delay>T_max
        ❖ w3 = 0
        ❖ w4 = 1
    ❖ else
        ❖ w3 = 1
        ❖ w4 = 0
    ❖ T_direct = w1*P_rate + w2*T_history(Fog_i)
    ❖ $T\_rec = \sum_{Adj=1}^{n} W_{Adj} * T\_present_{Adj}$
    ❖ T_newTrust(Fog_i) = w3*T_direct + w4*T_rec
    ❖ T_history(Fog_i) = 1
```

Optimal service offloader (Algorithm 1) offloads the service to the best fit Fog node whereas trust assessment (Algorithm 2) contributes to calculate trust value for each Fog node *Fog_i* in the Fog network.

In the trust assessment algorithm trust value for a particular Fog node will be calculated by considering direct and recommended trust values.

$$T\_newTrust = w3 * T\_direct + w4 * T\_rec \tag{1}$$

Here *w3* and *w4* are weightage parameters. Direct trust is the trust value calculated for the node *Fog_i* with respective to current *Fog_x* node by considering the parameters known as packet loss rate(*P_rate*) and history value (*T_history*). Here (*P_rate*) is defined as ratio of number of packet lost with respective to total packet received so for by *Fog_x*. And *T_history* is the value (0.0 to 1.0) purely depends on the successful completion of previously assigned service request. Direct trust is calculated as per Eq. (2).

$$T\_direct = w1*P_{rate} + w2*T\_history \tag{2}$$

Here w1 and w2 are weightage parameters calculated as follows.

$$w2 = k1 * Adj * e^{(-k2*Adj)} \tag{3}$$

where k1 and k2 are some initialized real constants

$$w1 + w2 = 1 \tag{4}$$

where Adj represents adjacency factor. Each node *Fog_i* will be assigned with its Fog id based on its location. The Adj factor will get less value if the target node (*Fog_i*) is nearer with respective to current *Fog_x* node.

Adj is calculated as in Eq. (5).

$$Adj = |x - i|/(n - 1) \tag{5}$$

Here $|x - i|$ will be less if *Fog_i* is more adjacent with *Fog_x* and here $(n - 1)$ indicates maximum distance they have in between. Here w2 is considered for *T_history* calculation since the history value maintained by nearest Fog node will be given higher weightage with respective to *Fog_x*. And this is the reason for calculating Adj value whereas if Adj value is more, then calculated w2 will have less value.

Direct trust value can be given weightage based on the parameter *T_delay*. *T_delay* represents vulnerable delay. It is defined as the time taken by *Fog_i* to forward the service request to the node Fog_j. If T_delay is greater than T_max (Maximum delay allowed by the node Fog_i—it vary based on the network characteristics such as throughput, latency jitter, etc.,) then this direct trust contribution for the total trust calculation will not be considered.

Another contribution for total trust calculation is recommended trust which is calculated as Eq. (6).

$$T\_rec = \sum_{Adj=1}^{n} W_{Adj} * T\_present_{Adj} \tag{6}$$

Here $T\_present_{Adj}$ is the trust value maintained by Adj Fog node with respective to the node (*Fog_i*). And W_{Adj} is the weight calculated for the particular Adj node by the node (*Fog_x*) which currently seeks for the recommended trust. Here W_{Adj} is calculated as in Eq. (7)

$$W_{Adj} = \frac{k}{\sum_{1}^{nN} k} \tag{7}$$

Here highest weighted value will be produced for the Adj node which is highly trusted with respective to (*Fog_x*). So, the recommendation given by highly trusted nodes are considered more for recommended trust calculation than rest in the sequence. Here *nN*represents node number maintained in *REC_TABLE* (Recommended trust table used for recommended trust assessment) which is generated by reversing *HTL_TABLE* from smallest to largest. And k represents location value of the nodes in the *REC_TABLE.*

After calculating direct and recommended trust, total trust for the particular (*Fog_i*) node will be calculated by current trust seeking node (*Fog_x*). So, an optimal service offloader considers calculated total trust while offloading trust worthy applications to the Fog environment. Overall architectural flow of the proposed TEFLON is illustrated in Fig. 2.

5 Simulation of TEFLON

In this section the proposed trust enforced computational offloading (TEFLON) framework is evaluated to assess its secure offloading characteristics in Fog to Fog collaboration. And various parameters such as latency, collaboration successful rate and packet distribution are

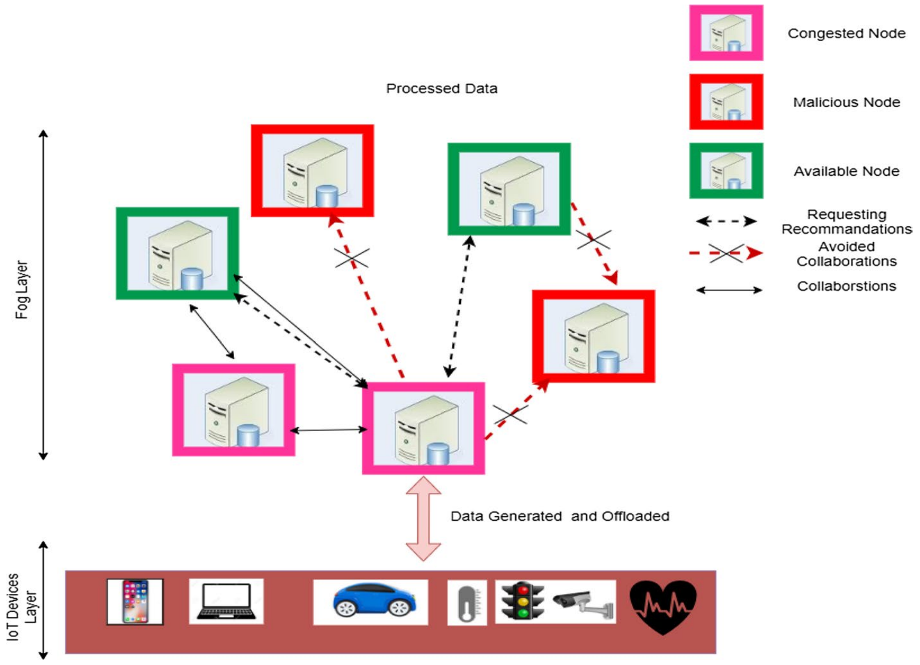


Fig. 2 Proposed TEFLON framework

measured and compared with popular Fog collaboration bench marking algorithms such as RWO (Random Walks Offloading) and NFO (Nearest Fog Offloading) by taking health care applications data set.

5.1 Simulation Parameters

The proposed TEFLON framework has been simulated using MATLAB (2016b) with the processor corei7 and 8 GB RAM. Simulation parameters are given in the Table 3.

5.2 Results and Discussion

In this section the proposed TEFLON framework is validated for its numerical accuracy for various parameters such as latency, successful collaboration rate, packet distribution, etc., The proposed framework TEFLON is compared against renowned bench marking Fog collaboration algorithms such as Random Walk Offloading and Nearest Fog Offloading.

The proposed TEFLON framework is self-evaluated to show trust value maintained among Fog nodes are asymmetric. That is if Fog_x node has some level of trust worthiness towards Fog_i node then it is not necessary that Fog_i node also should have the same level of trust on Fog_x. For instance, Fog_7 has the trust level towards Fog_15 as 0.138 but Fog_15 has 0.05 towards Fog_7. Trust asymmetric is shown in Fig. 3. Here horizontal and depth axis represents Fog id and vertical axis represents HTL score.

And HTL value maintained by Fog nodes are not transitive. Figure 4 shows this property by taking Fog Index in horizontal axis and HTL score in vertical axis. For example, if

Table 3 Simulation parameters

S. No	Simulation parameter	Value
1	Environment	MATLAB (2016b)
2	Number of Fog nodes	20
3	Network topology	Mesh topology
4	Data set	Image segmentation (Heart disease)
5	Number of Instances Tested	1500
6	Number of attributes	19
7	Data type	Multivariate
8	Bandwidth	64 Mbps
9	Operating System	Windows 10
10	Fog central processing power	2.4 GHz
11	RAM	8 GB

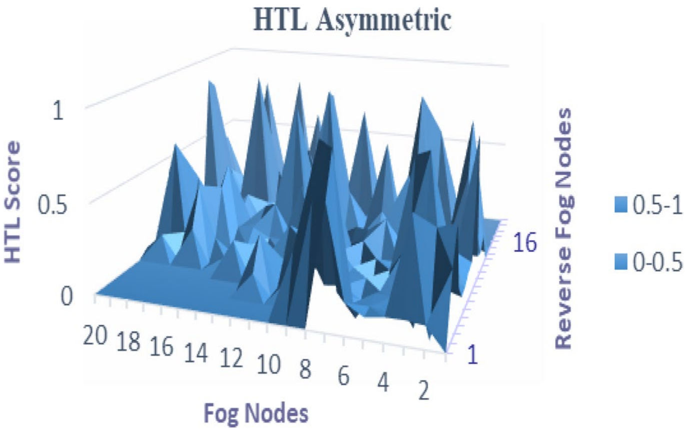


Fig. 3 Asymmetric Trust (HTL Score) property

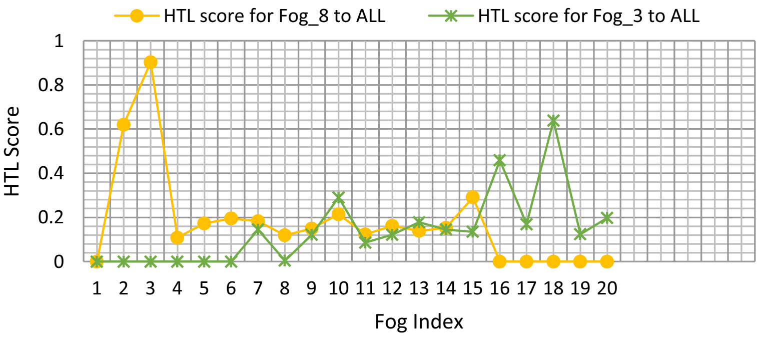


Fig. 4 Proving HTL is not transitive

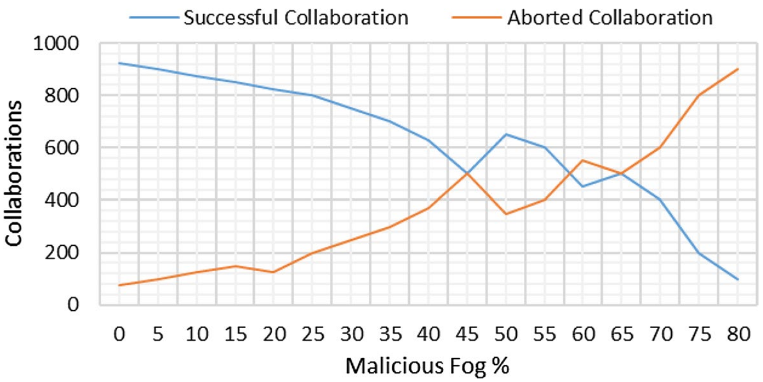


Fig. 5 Successful vs aborted collaborating when Malicious Fog increases

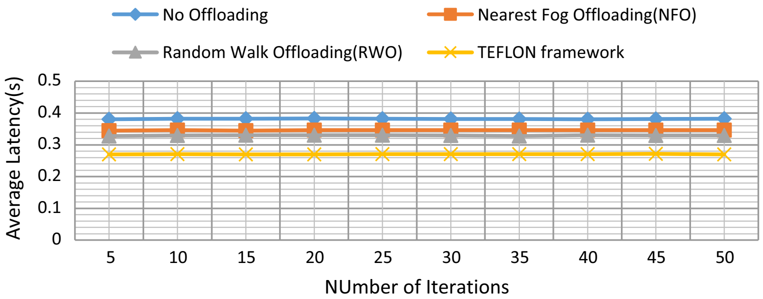


Fig. 6 Average Latency of Proposed TEFLON with NFO, RWO, and No offloading

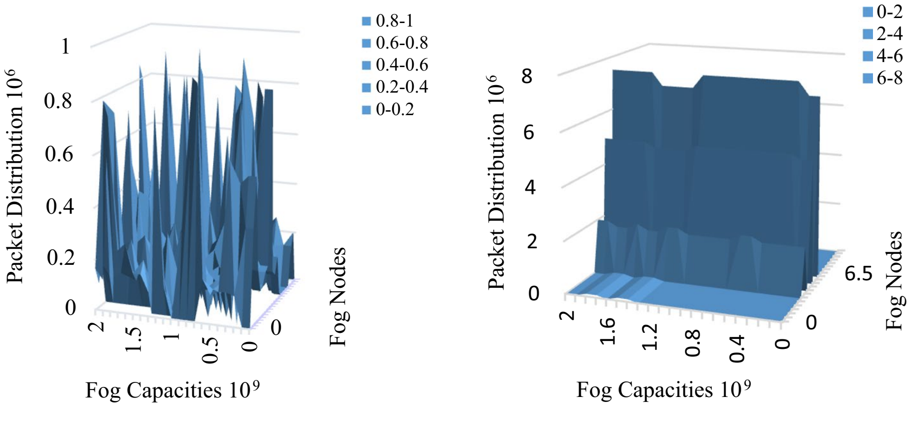
Fog_8 trusts Fog_3 and if Fog_3 trusts Fog_18 and it is not necessary that Fog_8 should trust Fog_18.

Figure 5 shows that if number of malicious Fog nodes increases in Fog environment then number of aborted Fog collaboration is also increasing hence number of successful Fog collaborations decreases. And horizontal axis in the figure shows percentage of malicious Fog nodes increased in a linear scale and vertical axis represents number of Fog collaborations.

Figure 6 shows measured average latency of proposed TEFLON framework against the following Fog collaboration techniques namely random walks offloading, nearest Fog offloading and average latency without offloading. For maintaining consistency among various algorithms number tasks taken for various iterations are fixed. And from the figure it is clearly shown that average latency for proposed TEFLON (0.27 s) is comparatively low with other bench marking Fog collaboration algorithms.

Figure 7 shows the packet distribution over Fog nodes. The proposed TEFLON framework is compared with NFO and RWO offloading algorithms. And to ensure same level of consistency among the offloading algorithms, either heavy or light packets has been taken throughout as the generated service requests.

Figure 7a shows packet distribution over Fog nodes by considering heavy packets (image segmentation for heart disease prediction) for NFO and RWO offloading algorithms. And Fig. 7b shows packet distribution of proposed TEFLON framework with the



(a) **(b)**

Fig. 7 **a** Packet Distribution of RWO and NFO, **b** Packet Distribution of proposed TEFLON

same input as in the case of Fig. 7a. And it is clearly shown that in TEFLON packet distribution is smooth. Because in TEFLON framework the current work load of designated Fog nodes was considered while performing offloading.

6 Conclusions

In this paper we proposed a framework TEFLON which comprises of two algorithms namely optimal service offloader and trust assessment to address the major issues of Fog environment. Optimal service offloader ensures reduced latency for delay sensitive applications by enforcing efficient parallelism. And trust assessment ensures trust for data sensitive applications by calculating direct and recommended trust values of Fog nodes. Here image segmentation-based healthcare data set for predicting heart diseases has been taken for evaluating the proposed framework. And simulation of proposed TEFLON framework outperforms popular Fog collaboration benchmarking algorithms such as RWO and NFO. In future the proposed framework can be extended to address the security and privacy issues in Fog collaboration with light weight secure algorithms.

References

1. you-need-to-know-about-the-iot-right-now/. https://www.zdnet.com/article/what-is-the-internet-of-things-everything-
2. Al-Khafajiy, M., Baker, T., Asim, M., Guo, Z., Ranjan, R., Longo, A., et al. (2020). COMITMENT: A fog computing trust management approach. *Journal of Parallel and Distributed Computing, 137,* 1–16. https://doi.org/10.1016/j.jpdc.2019.10.006.
3. Wang, T., Zhang, G., Bhuiyan, M. Z. A., Liu, A., Jia, W., & Xie, M. (2018). A novel trust mechanism based on fog computing in sensor–cloud system. *Future Generation Computer Systems*. https://doi.org/10.1016/j.future.2018.05.049.
4. 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.

5. Hussain, Y., & Huang, Z., (2018). TRFIoT: Trust and reputation model for fog-based IoT. In *International conference on cloud computing and security*, pp. 187–198. Springer, Cham. https://doi.org/10.1007/978-3-030-00021-9_18
6. Aazam, M., Zeadally, S., & Harras, K. A. (2018). Offloading in fog computing for IoT: Review, enabling technologies, and research opportunities. *Future Generation Computer Systems, 87*(2018), 278–289. https://doi.org/10.1016/j.future.2018.04.057.
7. Kochovski, P., Gec, S., Stankovski, V., Bajec, M., & Drobintsev, P. D. (2019). Trust management in a blockchain based fog computing platform with trustless smart oracles. *Future Generation Computer Systems, 101,* 747–759. https://doi.org/10.1016/j.future.2019.07.030.
8. Puthal, D., Ranjan, R., Nanda, A., Nanda, P., Jayaraman, P. P., & Zomaya, A. Y. (2019). Secure authentication and load balancing of distributed edge datacenters. *Journal of Parallel and Distributed Computing, 124,* 60–69. https://doi.org/10.1016/j.jpdc.2018.10.007.
9. Jøsang, A. (2007). Trust and reputation systems. In *Foundations of security analysis and design IV*, pp. 209–245. Springer, Berlin. https://doi.org/10.1007/978-3-540-74810-6_8
10. Chen, L., & Xu, J. (2017). Socially trusted collaborative edge computing in ultra dense networks. In *Proceedings of the second ACM/IEEE symposium on edge computing*, pp. 1–11, https://doi.org/10.1145/3132211.3134451
11. Chen, R., Bao, F., Chang, M., & Cho, J. H. (2013). Dynamic trust management for delay tolerant networks and its application to secure routing. *IEEE Transactions on Parallel and Distributed Systems, 25*(5), 1200–1210. https://doi.org/10.1109/TPDS.2013.116.
12. Guo, J., Chen, R., Tsai, J. J., & Al-Hamadi, H. (2016). Trust-based IoT participatory sensing for hazard detection and response. In *International conference on service-oriented computing*, pp. 79–84. Springer, Cham. https://doi.org/10.1007/978-3-319-68136-8_7
13. Deng, R., Lu, R., Lai, C., Luan, T. H., & Liang, H. (2016). Optimal workload allocation in fog-cloud computing toward balanced delay and power consumption. *IEEE Internet of Things Journal, 3*(6), 1171–1181. https://doi.org/10.1109/JIOT.2016.2565516.
14. Dinh, T. Q., Tang, J., La, Q. D., & Quek, T. Q. (2017). Offloading in mobile edge computing: Task allocation and computational frequency scaling. *IEEE Transactions on Communications, 65*(8), 3571–3584. https://doi.org/10.1109/TCOMM.2017.2699660.
15. Soleymani, S. A., Abdullah, A. H., Zareei, M., Anisi, M. H., Vargas-Rosales, C., Khan, M. K., & Goudarzi, S. (2017). A secure trust model based on fuzzy logic in vehicular ad hoc networks with fog computing. *IEEE Access, 5,* 15619–15629. https://doi.org/10.1109/ACCESS.2017.2733225.
16. Wang, X., Yang, L. T., Xie, X., Jin, J., & Deen, M. J. (2017). A cloud-edge computing framework for cyber-physical-social services. *IEEE Communications Magazine, 55*(11), 80–85. https://doi.org/10.1109/MCOM.2017.1700360.
17. Fan, Q., & Ansari, N. (2018). Towards workload balancing in fog computing empowered IoT. *IEEE Transactions on Network Science and Engineering*. https://doi.org/10.1109/TNSE.2018.2852762.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

V. Meena is an Assistant Professor and pursuing Ph.D in School of Computing ,SASTRA Deemed University, Thanjavur, India. She received her B.E. in Computer Science from Periyar Maniammai College of Technology for Women, Bharathidasan University, TamilNadu in 2000 and M.Tech in Advanced Computing from SASTRA Deemed University, Thanjavur in 2008. Her research interest includes Mobile cloud computing, Fog computing, optimization algorithms and machine learning.

Meghana Gorripatti is an undergraduate student in Computer Science & Engineering at SASTRA University, Thanjavur, Tamil Nadu. Her research interests include fog computing, mobile computing and wireless sensor networks.

Dr. T. Suriya Praba received her Ph.D degree from the Department of Computer Science and Engineering at Anna University in 2020. She is currently Assistant Professor in the Department of Computer Science and Engineering at SASTRA Deemed University. Her research interests include: Machine Learning, Fog Computing, Wireless Networks, Internet of Things, Network Security.