Concurrent service access and management framework for user-centric future internet of things in smart cities

P. Gomathi¹ · S. Baskar² · P. Mohamed Shakeel³

Received: 28 March 2020 / Accepted: 23 May 2020 / Published online: 8 June 2020
© The Author(s) 2020

Abstract

Future Internet of Things (FIoT) is a service concentric distributed architecture that is used by the smart city users for information sharing and access. The design of FIoT focuses in achieving reliable service and response to the growing user demands through different interoperability features. In this manuscript, concurrent service access and management framework is introduced to improve the swiftness in user concentric request processing. Based on the availability of the services and the density of the users, the concurrency in information access is provided to the users in a reliable manner. The framework incorporates convolution neural learning process in linear and differential manner for improving the access and service usage rates of the requesting users. The access sessions are differentiated for the accessible and offloaded requests to the available service providers based on the learning instances. The proposed framework is assessed using the metrics access rate, service usage rate, access delay, time lag, and failure ratio.

Keywords Access management · Convolution neural network · IoT · Request concurrency · Service allocation

Introduction

Future Internet of Things (FIoT) paradigm endorses sophisticated communication architectures and heterogeneous technologies for satisfying the service requirements of the varying users [1]. The conventional IoT architectures and infrastructures along with the information communication technologies (ICTs) are preserved in this platform with the concentration of service reliability and swiftness in access. The need for ubiquitous service requirements and resource access is increasing the recent times due to the portability of information and user-centric application demands [2, 5]. To meet the service requirements as user demands in a scalable and flexible manner, the ICTs are exploited to another level of communication along with data sharing and responsive integrations. The future aspect of distributed information

sharing extends to online storage, computation, and access due to portable and light-weight technologies [3]. The aim and intent of these service platforms is to enhance the quality of customer experience of irrespective category of service and use of applications. Precisely, swift operations and on-demand computation and storage requirements of the users are to be satisfied by exploiting multi-level of communication in an IoT platform [4].

The multi-level platform incorporates human-machine, device-to-device, human-robot, machine-to-machine, human-digital things, etc. for ensuring seamless and less complex information sharing platforms [5]. With the emerging smart city applications and service requirements in real world, the usage of data and exploitation of resources are massive. Handling large data resources and service features in IoT is facilitated with the incorporation of cloud-like infrastructures [6]. The distributed storage and sharing features of the cloud are directly inherited using the ICT in the smart city environment. As IoT-Cloud scales the communication architecture of the smart city in various applications such as transportation, traffic management, road safety, healthcare, data sciences, smart home management, etc., the need for service architectures is obligatory [7, 8]. The conventional short-range communication architectures are less

✉ S. Baskar
tobaskars@gmail.com

¹ N.S.N. College of Engineering and Technology, Karur, India

² Department of Electronics and Communication Engineering, Karpagam Academy of Higher Education, Coimbatore, India

³ Faculty of Information and Communication Technology, Universiti Teknikal Malaysia Melaka, Durian Tunggal, Malaysia

feasible in supporting scalable and flexible data sharing and service provisioning architectures [9].

Service-centric architecture design and implementation is focused with the assimilation of multiple platforms such as cloud, fog, mobile-edge computing, etc. The platform, application, and infrastructure as a service of the cloud paradigms are extended to the user devices through the intermediate architectures such as edge, fog and IoT [2, 10]. These architectures assures reliable and ease of access for the distributed environment, improving the service requirements for the end-users. The design of such integrated architecture requires flexible and extended ICT for supporting wide range of applications [5, 7, 11]. The application category of the users varies from tiny sensors to human-related multimedia services for which timely access and concurrency are required. Cloud service providers grant concurrency and access control methods for differential environments along with the knowledge of the resource sharing and service provisioning [10, 12].

Related works

Ren et al. [13] proposed an IoT Service Function Chains (IoT-SFCs) Orchestration in SDN-IoT Network Systems. In this work, a two-step processing is done: the first one is the composition, which is a multi-criterion based on ordered and modeled to resolve the order of involved IoT network functions. The second one is the establishment of a service function chain to illustrate the composed IoT-SFC.

Li et al. [14] introduced a resource service chain (RSC) to notice an anomaly business by means of business mining. The RSC is measured to evaluate the object, and the second is based on QoS making an anomaly business a standard of RSCs. The last one anomaly is detected in the business for collaborative tasks in IoT.

A cloud computing for context-aware IoT is developed by Lee et al. [15]. It consists of two layers such as the cloud control layer (CCL) and a user control layer (UCL). The CCL is used to manage the resource allocation in the cloud, scheduling, the service profile, and adaptation policy. The UCL is used to control the end-to-end service connection and context from the user view side.

Chen et al. [16] proposed a new heuristic method on IoT-Data Intensive Service Components Deployment (iDiSC) in the edge-cloud hybrid system. It is used for optimal operation development through the minimum guaranteed latency.

A Markov Approximation method is introduced by He et al. [17], for Optimal Service Orchestration. The author developed a Service Chain (SC) orchestration and learns the Virtual Network Functions (VNFs) placements. It increases the multiple instances to reduce the cost and delay, and it is more associated with the network guarantee and load balancing.

A Performance and Resource Aware Orchestration System is observed by Wang et al. [18]. The linear programming model makes an effective approximation for Service Function Chain (SFC) to avoid resource idealness. The prototype system is used to perform the IoT (PRSFC-IoT) which is modeled ahead on Open Stack for online SFC orchestration.

Khansari and Sharifian et al. [19] designed an evolutionary game theory in the fog-computing environment. A cloud-related framework for the collection and composition of IoT resources is used to boost the QoS. The multi-objective evolutionary game theory improves the evaporation-based water cycle algorithm (EG-ERWCA) in cloud assistant to optimize the CPU usage and power consumption.

Nasiri et al. [20] proposed a distributed stream processing in the data processing layer on smart cities. The ability to hold real-time data processing is done on Distributed stream processing frameworks (DSPFs). In a smart city, a variety of IoT devices produce continuous data required within a short period of time and analysis.

Kim et al. [21] proposed an intelligent IoT common service platform architecture and implements the fundamental procedures. An IoT broker is used for the connectivity and verifies the importance of the modeled work in the service-oriented platform. The IoT has huge potential knowledge for a higher variety of services.

A microservice is used to connect the IoT management platforms and AI service for chain management is modeled by Kousiouris et al. [22]. The author integrates the microservice system and implements the postprocessing tasks to provide the chain monitoring from the online systems.

Model-Driven Development of Service-Oriented IoT Applications is developed by Sosa-Reyna et al. [23]. This study consists of four stages such as abstraction, viewpoints, granularity, and service oriented. Using this methodology, the smart vehicle is addressed in heterogeneous devices in multiple ways.

QoS-aware service commendation based on the relational topic model and factorization machines for IoT Mashup applications is observed by Cao et al. [24]. The initial step of this is to characterize the relationship among the Mashup, services and links. The second one is to exploit the machine factorization to train the topics of latent to predict the link relationship.

Concurrent service access and management framework (CSAMF)

This framework is designed for improving the concurrency in service access for the varying user density in an IoT-based smart city. The purpose of the framework is to leverage the accessibility rate of resources and to reduce the failures in service sharing. In this framework, IoT, cloud and user requirements are consolidated to form a reliable solution. The reliable

solution provides both cost effective and responsive service platform for the smart city users. In a smart city environment, the service requests and access for different applications are processed as per the user requirements. The density of service requests and response relies on the accessible and available feature of the IoT-cloud infrastructure. In Fig. 1, the architecture of IoT-cloud smart city where CSAMF is deployed is illustrated.

The proposed frame work consists of two different process in a parallel manner, namely access management and service availability. These process are independent but are jointly used for validating the performance and improving the response reliability of the users. Depending on the requesting service or application, service availability varies for retaining the quality of reliability. The design of this framework retains cost and delay outcomes and its assessment using fitness based learning method.

Access management

The purpose of this process design is to balance the request and response of the users in an optimal manner. With the available infrastructure units and resources, this process ensures that no additional cost for resource access is required in the IoT architecture. The cost refers to the connection establishment with respect to the time span of the access time (t_a) and query time (t_q). Therefore, the objective of access management is given as

$$\left. \begin{array}{l} \text{maximize } r_r \forall r_q \text{ in } (t_a - t_q) \\ \text{minimize } \left(\frac{t_a - t_q}{t_a} \right) \\ \text{such that} \\ t_q \leq t_a \text{ and} \\ t_a > t_{\max} \end{array} \right\} \quad (1)$$

In Eq. (1), the variables r_r and r_q denoted the request responses and queries, respectively. The factor $\left(\frac{t_a - t_q}{t_a} \right)$ indicates the cost based on access and $t_{\max}$ is the maximum valid

time of the query request. Therefore, the objective is to maximize request response by providing less time for access. In a smart city, the density of users (ρ) varies with different instances and the request query for the application/services varies. However, the cloud and IoT are responsible for handling r_q and satisfying Eq. (1). Let $\{1, 2, \dots, S\}$ represent the t_q sessions in an IoT environment. The session refers to the augmentation of r_r from different devices in a non-replicated manner. If replication r_q occurs at two or more S , then it indicates the failure in response/access. The concurrent nature of r_q requires non-overlapping t_a irrespective of t_q . This means that the arriving request queries must not overlap with the other t_a responses providing multiple concurrencies in resource occurs. Now, the objective in Eq. (1) for the S is defined as in Eq. (2)

$$\left. \begin{array}{l} \text{maximize } \sum_{i=1}^S \left(\frac{r_r}{r_q} \right)_i \forall S \in [t_q, t_a] \\ \text{such that } t_q \leq S \leq t_a \end{array} \right\} \quad (2)$$

Based on the satisfactory condition that prevails in the cloud resource access, the fitness is estimated. This fitness conditions are sustained for all $(t_q - t_a)$ session, ensuring maximum access is provided for the r_q . The fitness (F) is estimated for two concurrently overlapping S , i.e., the continuous S that accepts requests and generates reliable r_r is accounted for estimating the fitness. This fitness is estimated using Eq. (3)

$$\left. \begin{array}{l} F_c = \frac{\tau - \tau_{\min}}{1 - \tau_{\min}}, \\ \text{where } \tau = \frac{r_r}{r_q} \text{ and } \tau_{\min} = \left[0, 1 - \frac{r_r}{r_q} \right], \text{ if } r_r < r_q \\ \text{and} \\ F_t = \frac{t_a - t_q}{(t_a - t_q) + t_l}, \\ \text{and} \\ F = \frac{F_c + F_t}{S}. \end{array} \right\} \quad (3)$$

Fig. 1 IoT-cloud-smart city architecture

In Eq. (3), the fitness is estimated with respect to time and request, where t_l is the time lag between two successive S . The fitness information is validated using linear convolution learning method for verifying if F_τ and F_t are mapped with each other. If the fitness is mapped precisely such that $F = 1$ or near to 1, then the access is high, increasing the reliability of the session. In this learning process, the above fitness is estimated only if t_a is allocated for a r_q . If that is the case, then F_t is the learning instance and the conditions in Eqs. (1) and (2) are the validations to ensure that F_τ is achieved. Let $\emptyset$ denote the solution space of the learning process such that the expected output is

$$\emptyset = \left. \begin{aligned} &\text{armin}_{r_q} \frac{r_\tau}{r_q} + \rho \left(1 - \frac{r_\tau}{r_q} \right) \pm \gamma, \\ &\text{where} \\ &\gamma = \frac{\rho \left(\frac{d}{dt_a} - \frac{r_\tau}{r_q} \right)}{F_t} \beta, \end{aligned} \right\} \quad (4)$$

where γ is the deviation between number of requests in the previous and current S and β is the rate of learning. The mapping of F_τ and F_t is a time t_a occurs in the convolution layer wherein, the probability of mapping, i.e., $p(F_\tau || F_t)$ is computed as

$$p(F_\tau || f_t) = \begin{cases} \sum_{i=1}^S \emptyset_i \left(\frac{F_\tau}{F_t} \right)_i + \gamma, & \text{if } F_\tau = F_t \\ \sum_{i=1}^S \emptyset_i \left(\frac{F_\tau}{F_t} \right) - \left(1 - \frac{r_\tau}{r_q} \right) \gamma, & \text{if } F_\tau \neq F_t. \end{cases} \quad (5)$$

This probability of mapping as in the above equation is performed for F_τ in $\emptyset$. Based on the analysis and validation, the number of instances in which mapping occurs is identified. The change in $p(F_\tau || F_t)$ is handled by concentrating on service availability. This learning process and communication for F_τ is presented in Fig. 2a, b.

The process of access management is performed between the IoT device and the IoT infrastructure. The allocation of t_a , r_τ and mapping relies on the resource availability in the cloud. Based on the $[\beta, \gamma]$ update pairs, the service chain is allocated. Therefore, for a session S with $t_l = 0$, the access is estimated as $\left(\frac{r_\tau \times \rho}{r_q} \right)$, wherein the response to the available service is given in t_a in a S . The conditions in Eqs. (2) and (3) are verified as follows:

If $t_l = 0$, then $(r_\tau - r_q)$ is the remaining requests that are to be allocated with $t_a = t_a + 1$ in the next S . Therefore, r_τ that $\in \emptyset$ shows maximum range between $\tau_{\min} \neq 0$ and $[(r_\tau - r_q), r_\tau]$. Similarly, for $(r_\tau - r_q)$, $(S + 1) \leq t_a$, the condition $t_a = t_a + 1$ falls in the consecutive S . In this case, t_q of $(r_q) > (r_\tau - r_q) \in S$ and the $(s + 1)$ handles the remaining requests. Here, the impact of β and γ is random where the probability of mapping as in Eq. (5) ensures the satisfaction

of Eq. (2). As per the objective and mapping performed in Eq. (5), the maximum access provided is $\left(\frac{r_\tau \times \rho}{r_q} \right)$ in S . This is maximized to ρ , if $S \leq t_a$ and does not require an additional $(S + 1)$. Besides, the learning conditions $(F_\tau = f_t)$ and $p(F_\tau || F_t) = 1$ determines the rate of β . Here, β is identified for all r_q process as in Fig. 2b to verify if the access in non-replication and F is attained in $\emptyset$. The β and γ factors are considered, if $S = S + 1$ or $t_a = t_a + 1$ where the concurrent access to the resources is augmented. The concurrency in access management is subject to the availability of resources in the IoT-cloud platform. The rate of learning is high, if $r_\tau = 0$ for all r_q allocated with t_a . On the other hand, the initial learning is focused on mapping F_τ with F_{t_a} such that all r_q (maximum r_q) is served. The changes in γ if exceed the solution space $\emptyset$, then the availability is to be considered in this case. In the next session, the service availability and its management in the IoT-cloud are discussed.

Service availability

In IoT cloud context, the rate of r_τ and retaining F_τ relies on the available resource. The rate of β and γ determines the availability and need for allocating new resource for the remaining r_τ . Therefore, the conditions that require more service availability are presented in Table 1 with their explanation.

The conditions and analysis presented in Table 1 discuss the possible assessment for improving the r_a on the given t_a . The optimal solutions on the $\emptyset$ rely on two factors, i.e., confining $\tau_{\min} \left(\frac{r_\tau - r_q}{r_q} \right)$ and $\gamma < 1 - \beta$ conditions at the time of resource management. The combination of service provides and virtual machines are jointly used for handling the query requests of the end-users. The available service providers/virtual machines are given by Eq. (6) as

$$A = \frac{r_q}{M} + \frac{r_q}{M^2} + 1, \quad \forall t_q < t_a, \quad (6)$$

where A and M denote the availability factor and service provider count. For the available M , the r_q satisfied at time t_a . The remaining requests are to be served as per the conditions in Table 1 to retain the swiftness and cost efficiency of the framework. The conditions in Table 1 define the inappropriate F_τ and F_t at different time intervals. The changes are S and offloaded requests to the all consecutive session are assessed if the access r_q is mapped to the availability resources/service in the next session. The ratio of availability of the M in its time period is given as $\frac{t_u}{r_q}$; therefore, Eq. (6) becomes

(a) Learning Process for $\emptyset$

(b) Communication between IoT device and IoT Gateway

Fig. 2 a Learning process for $\emptyset$. b Communication between IoT device and IoT gateway**Table 1** Condition and analysis of the service availability

Condition	Description	Analysis
$S > t_a$	The session that is allocated is not sufficient for the t_a of all the t_q received	If this case persists, then the service allocation is required to handle concurrent access. The service allocation is to be replicated in all instances
$t_a > t_{\max}$	The maximum request validity time is met but does not serve for any remaining request (i.e.) $(r_q - r_r)$ in time t_a	The remaining requests are to be allocated with the internal available service provider for balancing $t_{\max}$ and t_a
$F_{\tau} \neq F_t$	The fitness does not map the r_r and t_a or r_q and t_a , thereby reducing F	Increasing the S with respect to the available services and confining $(s + 1) \leq t_a$ is the general solution
$p(F_{\tau} F_t) \neq 1$	The rate of γ is high, reducing the r_r of the request queries in time t_a of in the next consecutive session $(S + 1)$	The rate of improving $p(F_{\tau} F_t)$ to 1 relies on confining t_1 to 0 and reducing the need for new $(S + 1)$

$$r_q = M \sqrt{\frac{t_u - 1}{M + 1}}, \quad \forall t_q < t_a, \quad (7)$$

which are the acceptable request queries from the previous S . Therefore, $(r_r - r_q) < r_q$ as in Eq. (7) are the request queries to be processed in $S + 1$. The condition in Table 1 is validated for the above r_q other than the generated r_q .

Therefore, $\frac{r_r}{r_q \in S + r_q \in S + 1}$ is the maximum achievable service response in the proposed framework. In this service availability process, the learning instances are modified based on the r_q and A [as in Eq. (6)]. The learning instances are different but cannot be augmented for different instances as it increases the complexity of processing and computation cost. Therefore, the linear learning method for $(r_r - r_q)$ is the input wherein the convolution layer filters the above conditions in a step-by-step manner for mapping F_r and F_t . This mapping is sequential such that the availability of M is the best-fit criteria for accepting $(r_r - r_q)$. The variations in $\frac{r_r}{r_q}$ is then computed in the $S + 1$ for the available M and r_q in the consecutive service mapping process. The overflowing r_q in the previous session is offloaded to the available M , wherein the errors are suppressed in F_r and F_t mapping. The errors are estimated as the deviation/change in r_r/r_q , where t_1 is not greater than t_a . The case of analysis for $S < t_a$ and $t_a \leq S + 1$ is retained by identifying the r_q that is needed for allocation time. The learning process is attuned as represented in Fig. 3a.

As illustrated in Fig. 3a, the process of learning focusses on A and rate of r_q . The left out r_q as in the above process

requires two sequential validations, i.e., $(r_r - r_q) < r_q$ and $\gamma < (1 - \beta)$. These two conditions if not satisfied, the error in learning process occurs. This error is restricted to the available resources as it may not accept the $(r_r - r_q)$; therefore, the request is offloaded to the M that is available is less t_1 . Therefore, a mediate of the request $\in (r_r - r_q)$ experiences a t_1 such that it is retained in the same state. The learning rate must be less compared to γ as the error would be high in assessing the M with A in time t_r . Therefore, with this condition, the requests that are not served for the different instances $(s + 1)$ are considered as the error and it forms the training set of the learning process. The process between IoT gateway and cloud is represented in Fig. 3b.

The service from the cloud experiences an additional time as $t_1 \neq 0$ and therefore, the condition $(t_a + 1) < S$ is verified by the gateway IoT to ensure maximum r_r is generated. However, the log in consecutive request processing is observed, wherein $r_q > (r_r - r_q)$ is retained such that the access rate is improved. The access and availability management are concurrently handled in this proposed method. Where the requests are arranged sequentially using different conditions in the learning process in the t_1 and error shifts, the resource usage and the acceptance rate of requests are increased.

Fig. 3 a Learning process using $(r_r - r_q)$. b IoT gateway and cloud process

(a) Learning Process using $(r_r - r_q)$

(b) IoT Gateway and Cloud Process

Performance analysis

CSAMF performance is verified using the simulations carried out in Contiki Cooja Simulator. The simulation environment is constructed in reference to the architecture presented in Fig. 1. For a smart city environment, a subnet consisting of multiple services such as traffic signal, transportation system, smart home, and industry is augmented to the scenario. In this scenario, 160 IoT devices are placed in the aforementioned applications. The detained configuration of the scenario components is presented in Table 2.

The above configuration is used for verifying the performance of the proposed framework using the metrics access ratio, failure rate, access time, time lag, and service usage rate. This performance is verified using a comparative analysis of the above metrics with the existing methods in [16, 17], and [19]. The methods are named as VPMIA, iDiSC, and EG-ERWCA respectively.

Access ratio

The proposed framework achieves high service access ratio by retaining $\frac{r_r}{r_q}$ and $\frac{r_r}{r_q \in S + r_q \in S + 1}$ at two different time instances t_a and $t_a + 1$. The change in request concentration is offloaded to the available M on the basis of t_1 , wherein maximum requests are provided with service access. The classification of the fitness as in the convolution layer increases the differentiation between the requests assigned to the service M . Therefore, $\left(\frac{r_r \times \rho}{r_q}\right)$ in S is the maximum achievable rate of access; whereas, the remaining $(r_r - r_q)$ in $(S + 1)$ satisfying the conditions in Eqs. (2) and (3) helps to maximize the ratio. Therefore, in both S and $S + 1$ sessions, the rate of r_r is high (without replication), which means the access to M is retained in an abruptly high range (Fig. 4).

Table 2 Configuration and values

Configuring factor	Value
Service providers	100
Resource servers	16
Gateway units	12
Server capacity (in Gb)	2 each
Requests/session	20–200
Avg. session time	240 s
Bandwidth (in Mbps)	1024

Failure rate

Failure rate per session in the proposed frame work is confined by verifying the acceptable limit of r_r and A of the M in $S + 1$, respectively [25]. For any two consecutive execution sessions, t_a or $t_a + 1$ is retained within sky filtering $r_q \in S$ and $(r_r - r_q) \in S + 1$ from $\emptyset$. Therefore, if $S, s + 1, \dots \in \emptyset$, then the solution generated is high, reducing the failure rate. The number of denied requests in this case is reduced, as the $p(F_r || F_t) = 1$ condition is verified in the linear learning process and $\gamma < (1 - \beta)$ is the verifying condition in service management learning. Therefore, the rate of r_r in S and $(S + 1)$ is retained to a high level. The error in the proposed framework is achieved only if t_1 is prolonged for a r_r such that t_a or $t_a + 1$ exceeds $S + 1$.

This error results in failure rate increment; whereas, the r_r that are not served at S are offloaded to $S + 1$ session (Fig. 5a, b).

Access time

Concurrent request processing and access in the proposed framework is augmented to gather by allocating M at precise time [26]. The swiftness in request query processing and M assigning are verified by differentiating the S for satisfying $t_q \leq S \leq t_a$ condition. To preserve this condition, $\emptyset$ is estimated on the basis of fitness and maximum probability of mapping r_q . Instead, the $\emptyset$ for $(r_r - r_q)$ in $S + 1$ is retained by verifying the learning rate and difference in response. The r_q for the M assigned in S are said to satisfy the conditions of Eq. (2) and (3), respectively. Whereas, the $(r_q - r_r)$ in $(S + 1)$ are concurrently assigned for M in $(S + 1)$ -based mapping. As long as the learning rate satisfies $\gamma < (1 - \beta)$ condition, $t_a \leq S + 1$ and therefore, the access time is less. However a time lapse is observed in $S + 1$ session, where $t_a + 1 \leq S$ achieves less access time (Fig. 6).

Fig. 4 Access ratio comparisons

(a) Service Providers versus Failure Rate/ S

(b) Failure % versus IoT Devices

Fig. 5 a Service providers versus Failure Rate/S. b Failure % versus IoT devices

Average time lag

For the varying request density, time lag varies depending on the availability of M and S . The concurrent user access increases the request rate, varying the S and t_a . Therefore, instantaneous availability of M serves both $r_q \in S$ and $r_q \in (S + 1)$ in the allocated t_a . The time lag for $r_q \in S$ is not observed as the requests are served in a first come first serve manner. The learning rate and difference is retained based on the response condition $(r_r - r_q)$. The average time lag in the proposed framework is confined to the level of A and r_q [as computed using Eq. (7)] is the required factor for improving the swiftness in access. On the other hand, if the Swiftness in processing is achieved, then the offloading r_q are processed within time $t_a + 1$, i.e., $t_a + 1 < S + 1$ and hence, a slighter t_1 is achieved in the proposed framework (Fig. 7).

Fig. 6 Average access time comparisons

Service usage rate

The service utilization rate of the proposed framework is achieved by suppressing the access time constraints and A of the M . In both S and $S + 1$ the classification and learning constraints is ignored. The condition verification based on F_r and $\gamma < (1 - \beta)$ helps to retain the number of requests that are currently active on the M . The chances of less r_r and unconditional offloading of r_q is prevented in this framework. The difference in offloaded and β performs a re-allocation of S and if $t_a + 1 > S$, then the concurrency is retained by assigning $(r_r - r_q)$ to the available M based on minimum t_1 . The performance for M with $t_1 = 0$ is given such that the changes in handling request are suppressed in a reliable manner. Therefore, the M is engaged to its maximum time by handling $r_q \in S$ or $r_q \in (S + 1)$ and hence, the service utilization is high (Fig. 8). In Table 3, the comparison of Access %,

Fig. 7 Average time lag comparisons

Fig. 8 Service usage rate comparisons

Failure Rate/ S , and M usage rate with respect to the varying service providers is presented.

In Table 4, the comparative analysis of access time, time lag, and failure % is presented.

From the comparative analysis, it is seen that with respect to the varying service providers, and request concentration, the proposed framework achieves better access and service utilization rate, reducing access time, and failure rate and time lag.

Conclusion

This manuscript discusses concurrent service access and management framework for future IoT in smart city environment. This system works independently of access and service management to increase the speed at which resources are used and to improve the quality of services. The adversaries in request processing and service access

Table 3 Comparative analysis with respect to the varying service providers

Metrics	VPMIA	iDiSC	EG-ERWCA	CSAMF
Access %	81.22	83.69	86.88	92.74
Failure rate/ S	0.737	0.641	0.291	0.095
M usage rate	33.35	57.44	92.47	179.98

Table 4 Comparative analysis of access time, time lag, and failure %

Metrics	VPMIA	iDiSC	EG-ERWCA	CSAMF
Average access time (ms)	96.919	45.46	34.515	7.307
Average t_l (ms)	58.119	25.975	36.762	19.162
Failure %	6.513	4.945	3.604	1.615

are sustenance through convolutional learning process. The learning process operates in a linear and differential manner for classifying the requests and offloading them to improve the concurrency in service access. The classified requests are allocated based on the availability of the service providers in different session confining the level of access time and, in future, optimized learning frameworks are planned to use in concurrent service access and management. The experimental results show that the proposed framework achieves less access time and failure rate, improving the service utilization and access rate.

Acknowledgements The authors would like to thank SERB, Science and Engineering Research Board, New Delhi, India for the funding to carry out the research work at our institution.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Amadeo M, Ruggeri G, Campolo C, Molinaro A (2019) IoT services allocation at the edge via named data networking: from optimal bounds to practical design. *IEEE Trans Netw Serv Manag* 16(2):661–674
- Badawy MM, Ali ZH, Ali HA (2019) QoS provisioning framework for service-oriented internet of things (IoT). *Clust Comput*. <https://doi.org/10.1007/s10586-019-02945-x>
- Femminella M, Pergolesi M, Reali G (2018) IoT, big data, and cloud computing value chain: pricing issues and solutions. *Ann Telecommun* 73(7–8):511–520
- Montori F, Bedogni L, Bononi L (2018) A collaborative internet of things architecture for smart cities and environmental monitoring. *IEEE Internet Things J* 5(2):592–605
- Baskar S, Shakeel PM, Kumar R, Burhanuddin MA, Sampath R (2020) A dynamic and interoperable communication framework for controlling the operations of wearable sensors in smart health-care applications. *Comput Commun* 149:17–26
- Park J-H, Salim MM, Jo JH, Sicato JCS, Rathore S, Park JH (2019) CIoT-Net: a scalable cognitive IoT based smart city network architecture. *Hum Centric Comput Inf Sci* 9(1):29
- Manogaran G, Shakeel PM, Fouad H, Nam Y, Baskar S, Chilamkurti N, Sundarasekar R (2019) Wearable IoT smart-log patch: an edge computing-based Bayesian deep learning network system for multi access physical monitoring system. *Sensors* 19(13):3030
- Tao M, Zuo J, Liu Z, Castiglione A, Palmieri F (2018) Multi-layer cloud architectural model and ontology-based security service framework for IoT-based smart homes. *Future Gener Comput Syst* 78:1040–1051

9. Sun D, Li G, Zhang Y, Zhu L, Gaire R (2019) Statistically managing cloud operations for latency-tail-tolerance in IoT-enabled smart cities. *J Parallel Distrib Comput* 127:184–195
10. Li X, Lian Z, Qin X, Jie W (2018) Topology-aware resource allocation for IoT services in clouds. *IEEE Access* 6:77880–77889
11. Bibri SE (2018) The IoT for smart sustainable cities of the future: an analytical framework for sensor-based big data applications for environmental sustainability. *Sustain Cities Soc* 38:230–253
12. Sodhro AH, Pirbhulal S, Luo Z, de Albuquerque VHC (2019) Towards an optimal resource management for IoT based Green and sustainable smart cities. *J Clean Prod* 220:1167–1179
13. Ren W, Sun Y, Luo H, Obaidat MS (2019) A new scheme for IoT service function chains orchestration in SDN-IoT network systems. *IEEE Syst J* 13(4):4081–4092
14. Li H, Tong J, Weng S, Dong X, He T (2019) Detecting a business anomaly based on QoS benchmarks of resource-service chains for collaborative tasks in the IoT. *IEEE Access* 7:165509–165519
15. Lee T-D, Lee BM, Noh W (2018) Hierarchical cloud computing architecture for context-aware IoT services. *IEEE Trans Consum Electron* 64(2):222–230
16. Chen X, Tang S, Lu Z, Wu J, Duan Y, Huang S-C, Tang Q (2019) iDiSC: a new approach to IoT-data-intensive service components deployment in edge-cloud-hybrid system. *IEEE Access* 7:59172–59184
17. He W, Guo S, Liang Y, Qiu X (2019) Markov approximation method for optimal service orchestration in IoT network. *IEEE Access* 7:49538–49548
18. Wang J, Qi H, Li K, Zhou X (2018) PRSFC-IoT: a performance and resource aware orchestration system of service function chaining for internet of things. *IEEE Internet Things J* 5(3):1400–1410
19. Khansari ME, Sharifian S (2019) A modified water cycle evolutionary game theory algorithm to utilize QoS for IoT services in cloud-assisted fog computing environments. *J Supercomput* 76:5578–5608
20. Nasiri H, Nasehi S, Goudarzi M (2019) Evaluation of distributed stream processing frameworks for IoT applications in Smart Cities. *J Big Data* 6(1):52
21. Kim J, Jeon Y, Kim H (2016) The intelligent IoT common service platform architecture and service implementation. *J Supercomput* 74(9):4242–4260
22. Kousiouris G, Tsarsitalidis S, Psomakelis E, Koloniaris S, Bardaki C, Tserpes K, Nikolaidou M, Anagnostopoulos D (2019) A microservice-based framework for integrating IoT management platforms, semantic and AI services for supply chain management. *ICT Express* 5(2):141–145
23. Sosa-Reyna CM, Tello-Leal E, Lara-Alabazares D (2018) Methodology for the model-driven development of service oriented IoT applications. *J Syst Architect* 90:15–22
24. Cao B, Liu J, Wen Y, Li H, Xiao Q, Chen J (2019) QoS-aware service recommendation based on relational topic model and factorization machines for IoT Mashup applications. *J Parallel Distrib Comput* 132:177–189
25. He J, Wei J, Chen K, Tang Z, Zhou Y, Zhang Y (2018) Multitier fog computing with large-scale IoT data analytics for smart cities. *IEEE Internet Things J* 5(2):677–686
26. Marques P, Manfroí D, Deitos E, Cegoni J, Castilhos R, Rochol J, Pignaton E, Kunst R (2019) An IoT-based smart cities infrastructure architecture applied to a waste management scenario. *Ad Hoc Netw* 87:200–208

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