RESEARCH ARTICLE OPEN ACCESS

Performance Evaluation of Augmented Reality on Fog Computing Platform

Kinan Nabeh Abbas [1], Ahmad Saker Ahmad [2]
Department of System and Computer Networks Engineering
Tishreen University
Latakia - Syria

ABSTRACT

Despite the massive development in cloud computing technologies to be in line with the evolution of the Internet of Things (IoT), cloud computing remains somewhat weak in handling applications that require real-time processing. Moreover, since most of the IoT applications are classified as real-time applications (such as augmented reality applications), it is not possible to rely entirely on cloud computing for these applications. According to Gartner, the number of IoT devices that are to connect to the Internet by 2020 will be around 5.8 billion. With this increase in the number of devices that may be located in a small geographical area (as the case of smart cities), the amount of data transferred to the cloud will be massive and require tremendous processing capabilities by the cloud to satisfy it. This is only costly, but most of the current clouds fail to match the requirements of such applications and devices, hence the need for a technology to eliminate this gap between cloud computing and the Internet of things.

Cisco introduced Fog Computing technology, an extension of cloud computing that places the cloud nearer to the things that generate data where it provides real-time processing and storage.

This research offers a practical study to implement an augmented reality application used to enhance learning about plants in a smart city applied in Barcelona. The authors studied Fog computing implementation for 8 differenet application placements. Results show that the HAFA and iFogStor-G application placement policies have achieved the best service time and the lowest use of cloud processors.

Keywords: — Fog Computing, IoT, Cloud Computing, Augmented Reality, HAFA, iFogStor-G, iFogStor-Z, Edge Computing.

I. INTRODUCTION

Augmented reality is a type of virtual reality that aims to duplicate the real environment in the computer and enhance it with virtual data that are not part of it. An augmented reality system generates a composite display for the user that mixes the real scene the user is looking at and the virtual scene created by the computer to enhance the actual scene with additional information. Cloud computing technologies have significantly evolved to support the Internet of Things technology and augmented reality and what has become known as Cloud of Things (CoT) has emerged. Despite the massive development in cloud computing technologies, cloud computing remains somewhat lacking in handling applications that require real-time processing, where the transmission time, processing time, and decision-making time must be in milliseconds. Furthermore, cloud computing exhibits weakness in dealing with large numbers of connected devices, especially as the number of these devices increases dramatically (smart cities), hence the need for a technology to eliminate the gap between cloud computing and IoT applications. In 2016, Cisco [1] introduced Fog Computing technology [2] to be a helping technology for cloud computing and contribute to providing processing and storage services for IoT devices and smartphones in their local networks. Cisco defined the fog computing architecture and demonstrated this technology would be an extension of cloud computing and facilitate its work in the field of Internet of Things. Researches in [5] suggested using fog computing as a model for managing the operation of IoT devices. They discussed theoretical scenarios for using IoT with fog computing and identified the benefits of deploying this model.

The study in [6] proposed several theoretical scenarios for the use of fog computing in the field of health care and augmented reality which requires tremendous processing capabilities during a very narrow time window. The design of these systems was discussed on a cloud-only placement and fog–cloud placement. A model of health care systems was designed using fog computing in both [7] and [8]. The model consisted of three layers, and the study discussed two policies for application placement: cloud-only placement and edge-only placement. Building on a similar model in terms of positioning and the number of layers, researchers in [9] designed a wireless sensor application (WSN) in a fog computing environment to measure gas density (co / co2) to detect an emergency and calculate the number of people in a room. The goal is to plan an effective rescue method or control ventilation and heating according to the number of people and thus save energy. Application placement policies were proposed in [3], [4], [10] which are (HAFA), (iFogStore) and (iFogStor-G) (iFogStor-Z). Still, these policies have not been implemented on practical applications of IoT and AR, nor for real smart cities. Although several implementations of

AR has been studied on the fog computing platform, the studies were theoretical covering simple scenarios at best. The authors found no previous work that covers real-life scenarios nor includes all current application placement policies.

The research aims to study the fog computing model as a practical model for augmented reality (AR). We implemented an AR game that used to enhance learning about plants within a virtual smart city located in Barcelona. The city contains fog nodes distributed geographically in 6 layers. Accurate geographical coordinates were obtained for schools, universities, supermarkets and government institutions. The foggy nodes in each location were simulated using the PFogSim emulator. The application was executed to determine the best application placement policy.

II. FOG COMPUTING

Fog computing is defined as a group of heterogeneous devices scattered in several places, does not contain a central processing node, and provides processing and storage services to users in real-time with or without third party assistance [2,5]. In smart cities where there are a massive number of devices, vehicles and people connected on the Internet, the work of these devices includes the presence of applications that provide services for them. These applications are published on all processing nodes in the city by relying on fog computing where the fog nodes are divided into several layers as follows [11] [12]:

I. Miniature Fog Nodes:

These are nodes with minimal processing and storage capabilities and very little use of network resources. These nodes are abundant and widespread throughout the city. They are located near the Internet of Things devices. These nodes can be mobile or fixed and consume very little electrical energy, and are classified as unreliable nodes and do not have high availability.

II. Small Fog Nodes:

These nodes have small capabilities such as smart cars, laptops and home routers. These nodes are located close to IoT devices and users, and have higher availability than miniature fog nodes. They are used for simple processing and cannot be used for long-term storage.

III. Community Fog Nodes:

These are nodes with medium resources such as servers in shopping malls or the train station. These nodes are common to several buildings and support a large number of applications. They are fixed non-mobile nodes that are used for basic processing operations. They cannot be used for analysis, extraction of patterns from data, complex mathematical operations nor long-term data storage.

IV. Edge Fog Nodes:

They are a small group of smooth and homogeneous nodes that are similar in specifications, such as servers at the university or school. These nodes are placed close together and are immovable. They can perform somewhat large processing operations, extract small statistical results and provide temporary storage.

V. MicroData Centres:

Miniature Clouds are managed by Service Providers (ISP) offering substantial processing capabilities, high privacy, and security.

VI. Infrastructure Cloud Data Centres:

They are large servers located in a single geographic region such as Google and Amazon data centres. The resources in these centres are unlimited and are used for very complex calculations, learning and training algorithms and long-term storage. They are reliable, available and highly secure.

III. APPLICATION PLACEMENT POLICIES IN FOG COMPUTING

I. Local Only Placement:

In this mode, applications are published locally to the nearest node in the local network, where small nodes and miniature nodes are used to host applications [13].

II. Edge Only Placement:

In this mode, applications are deployed to edge nodes. Applications are arranged according to distance or delay and executed accordingly. Therefore, we have two policies that place two subfolders [13]:

Edge by Distance: The applications are arranged according to the distance between the IoT device and the fog node that hosts the application.

Edge By Latency: The fog node that achieves the lowest network delay is chosen. The delay includes congestion, latency, and propagation delay.

III. Cloud Only Placement:

In this policy, all applications are published to the cloud, where small data centres and large data centres are used to host the applications. In this policy, all requests are transferred from IoT devices to the cloud.

IV. IFogStor:

It is an application placement policy in fog computing. The nodes and links are modelled as Generalized Assignment Problem and solved with linear programming [10]. In this method, the network is divided into three layers, the first layer contains the Internet of Things devices. The fog nodes are located in the second layer, while datacentres are in the last layer. This method aims to find the best application placement for fog nodes to reduce response time. Assuming we have a set of fog nodes represented by the group Sn = {Sn1, Sn2, ...Snm}, and we have a set of data (tasks) of IoT devices represented by the group D = {d1, d2, ...dz}, the algorithm finds the best Set D to Sn to reduce response time.

Two criteria must be respected when implementing this algorithm:

1) The node can't take tasks more than it can handle.

2) No task loss is allowed.

V. iFogStor-z:

It's an application placement policy that uses Heuristic solution where the network is divided into geographical regions based on service points (Pops) as shown in figure (1):

Fig 1 iFogStor-G Application Placement

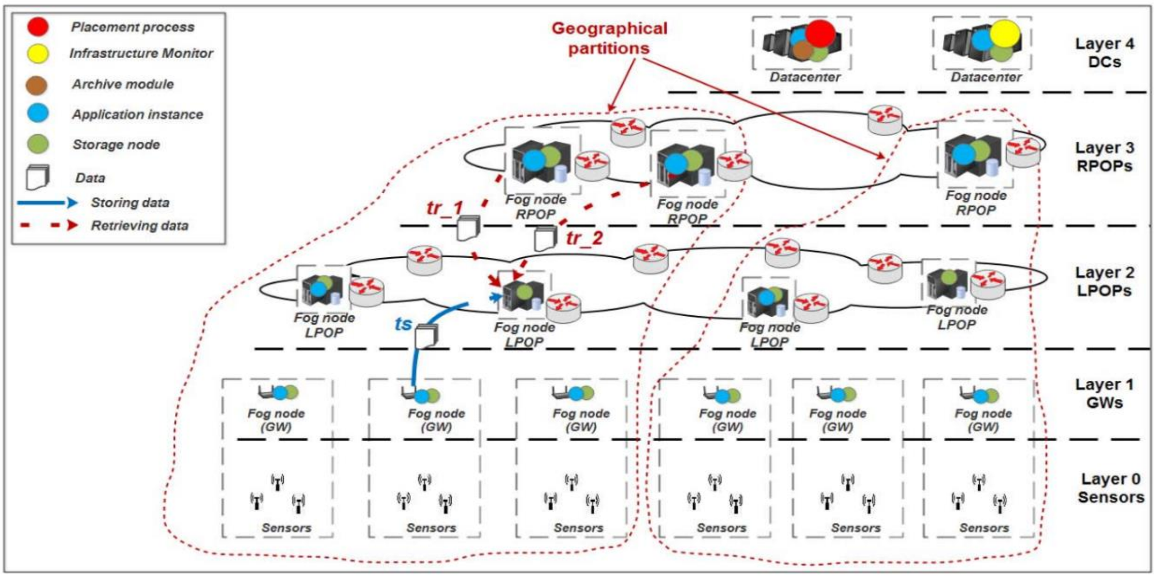

Fig 1 iFogStor-G Application Placement

Each section is a separate problem and is solved using the iFogStor algorithm, so the best solution is chosen among all the solutions. This method does not have an optimal solution in terms of response time, but it reduces the time to solve the problem very significantly [9].

VI. iFogStor-G:

It is a policy developed to solve application placement in fog computing using the Heuristic Solution. The algorithm works in four steps [4]:

1) Infrastructure Modelling.
2) Putting weights on the graph.
3) Graph Partitioning using K-Ways, as shown in Figure 2.
4) Data placement problem solving using ifogStor for each section.

VII. Hierarchical and Autonomous Fog Infrastructure (HAFA):

In order to facilitate the effective deployment of services and management of the Fog platform architecture, a hierarchical and independent fog architecture [3] (HAFA) has been proposed. The HAFA organizes the fog nodes in architecture made up of several layers that are logically interlinked depending on several factors such as location, the distance between the fog node, IoT devices and end-users, privacy and security requirements. The algorithm is implemented according to the following stages:

Layering, Grouping, Local Management, Inter-Layer Connection, Intra Layer Connection, Puddle Tree.

IV. PRACTICAL WORK

Geographical coordinates were obtained for schools, malls, hospitals, universities, and service providers for the city of Barcelona, and a 6-layer smart city was built using pFogSim [13]. This city was chosen because of the availability of data on the Internet [14], where nodes specifications were defined as shown in Table (I).

TABLE II
Characteristics of nodes

Place	Count	Num of CPUs in each node	RAM for each CPU	OS	VMs OS
Cloud	1	500	320000 GB	Xen	Linux
Service Provider	1	100	6400 GB	Xen	Linux
University	7	10	320 GB	Xen	Linux
Hospital	20	8	64 GB	Xen	Linux
Mall	23	6	64 GB	Xen	Linux
School	140	2	32 GB	Xen	Linux

The nodes described in Table 1 were defined according to the processing capabilities within the pFogSim emulator in levels, where schools were at the first level (weaker processing capabilities) and then malls, hospitals, universities and service providers in addition to a cloud service located outside the city. Network links between all levels were at a speed of 4G, which is 100 Mbps.

We implemented an augmented reality application used to enhance learning about plants. This application has been studied for different positioning policies (Local, Cloud, Edge by Latency, Edge By Distance, iFogStor, iFogStor-z, iFogStor –G, HAFA) and a better application placement policy has been found.

V. (MAGIC FLOWER POT) AN AR GAME FOR LEARNING ABOUT PLANTS

It is a game designed to promote learning about plants. During the game, players gather and nurture flowers in the local environment through their smart devices, and then form virtual gardens, as shown in Fig.2 [15]. The game works in three stages:

1) Collect plant seeds.
2) Seeding and planting.
3) Forming a virtual garden (AR Garden)

Fig. 1 Augmented Reality Planet

The first stage is played outdoors in nature. The aim of it is to collect the seeds of plants for planting at home, where the player searches for the plants they want to grow, takes pictures of the plant using their smartphone and uploads the pictures to the game server. The system identifies the plant from the image. It generates a virtual seed (Virtual Seed) that contains the ID of the digital model that represents the plant and information about the ideal environmental conditions for its growth.

The second stage is played at home. The aim is to grow the virtual seed collected in the previous stage. The player sows the seed in the Augmented Smart Pot. In order for the planting and growth process to succeed, the player must control the environmental changes to ensure that the plant gets enough light, heat, and water every day.

In order for the player to monitor the growth of the plant, they can use their smartphone or use a virtual reality glasses (Holo Lens HMD) [16] that display a virtual representation of the plant and its current state within the flower pot, providing information on the current conditions of the environment (sun - clouds - humidity). The requirements for the work of this stage:

1) The enhanced smart flowerpot with controller and three sensors of heat, light and air humidity.

2) A program that works on the controller and manages the plant growth process, as it collects the values from the sensors to determine the state of the plant.

3) Virtual reality glasses or a smartphone with an application that supports the game operation.

The proposed scenario for this system is to find the best application placement policy that identifies the values captured from the sensors and draws the image of the plant using augmented reality on the mobile phone screen.

VI. RESULTS

I. Average Processing Time:

Figure 3 shows the average processing time in milliseconds, which is the time needed to process sensors' data and the formation of the plant's three-dimensional shape for different application placement policies. Through the figure, we find that cloud placement records the minimum processing time, due to its high resources where the waiting and processing time is small. When implementing the Edge by Latency policy, the average processing time was 800 milliseconds for 10,000 devices. Still, we note a gradual decrease in this time with the increase in the number of devices. The reason for this is that after the network congestion with messages and a lot of tasks await processing on the less-delay nodes, a group of nodes that have high processing specifications and are far from devices become less delay and process tasks faster than closer nodes.

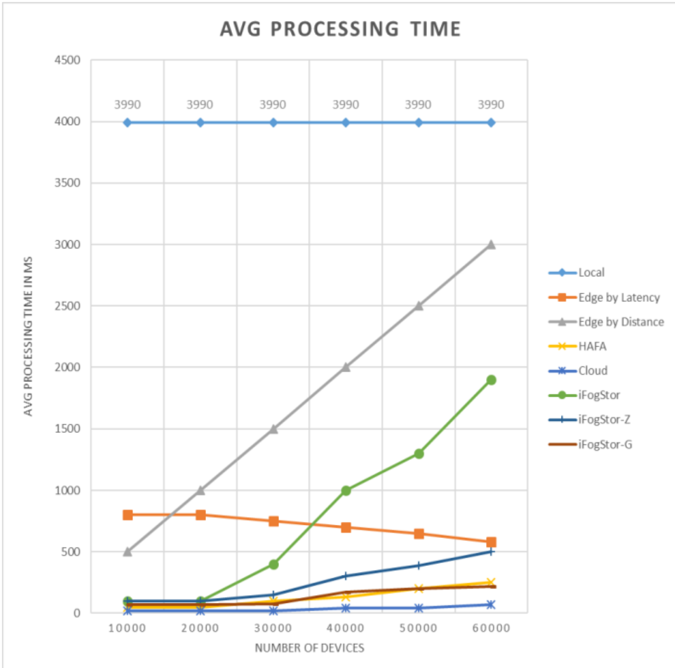

Fig 3 Avg processing time

While the policy of "Edge by Distance" was very expensive, as the average maximum processing time was 3 seconds for 60 thousand devices, although it was not significant for 10 thousand devices. The reason behind this case is the focus of treatment on the nearby nodes. Through the figure, we also find that the local placement of the application data on the mobile devices was costly and it took 4 seconds for each device, due to the weak hardware resources and its inability to handle such type of applications. (HAFA) and (iFogStor-G) application placement methods recorded excellent times. The average processing time for 60 thousand devices reached approximately 250 milliseconds. This is due to the independence of management and control of the model of (HAFA) and thus better task scheduling and a split policy in the iFogStor-G model that finds the fastest node to process the task in a balanced manner across the network. The average maximum processing time in iFogStor-Z policy was 500 milliseconds, which is a good time due to the algorithm dividing the network into Zones and implementing the iFogStor algorithm for each region separately.

II. Average Network Latency:

Figure 4 shows the average network delay time in milliseconds, which is the time needed to transfer the sensor values to the processing node and then send the generated shape to the smartphone device. Through the figure, we find that the cloud placement policy had a higher delay rate than others since all devices send and receive their data to and from the cloud, which generated high load and congestion on the cloud servers. The (Edge By Distance) policy also recorded a high delay of three seconds. The reason for this is that the processing was only the nodes near only the smartphone.

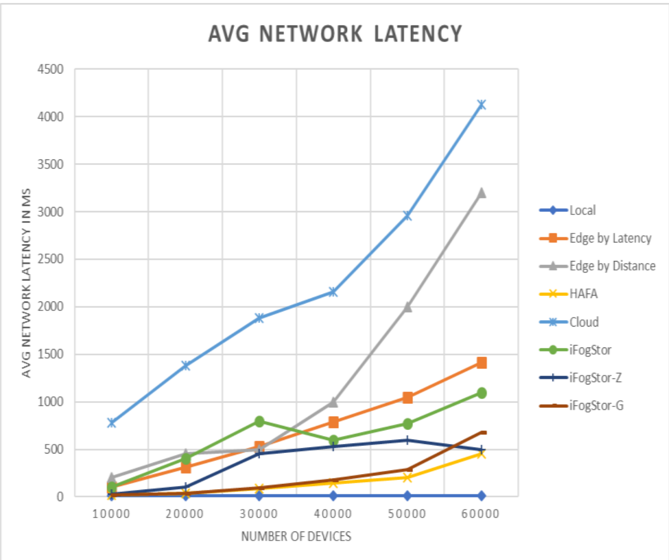

Fig 4 Avg Network Latency

When using the pattern (Edge By Latency), the average maximum delay time was 1500 milliseconds, thanks to the nature of the algorithm that always searches for the node that has the least network delay. As for the placement policies (iFogStor-G), (iFogStor-z) and (HAFA) the times were close, and all of them were less than one second. (HAFA) policy recorded the best time between them because of the independence of management and control of Puddles in this algorithm. On the other hand, the use of local placement (Local) produced the least delay among all policies, because the communication between the device and the sensor was within the local network of each device.

III. Average Service Time

Service time is the time required to process the device request in addition to the time required to send the sensor values and the transmission time of the shape generated by the node to the device, and therefore the service time is the processing time + network delay time.

Figure 5 shows the average service time in milliseconds for different application placement policies. Through the figure, we find that (HAFA) had the best service time of 700 milliseconds for 60 thousand connected devices, followed by (iFogStor-G) and then (iFogStor-z) where the average service time was less than a second for the greatest number of devices. As for Cloud and Local, the average service time was approximately 4 seconds. Edge by distance policy recorded the highest time by approximately 6 seconds.

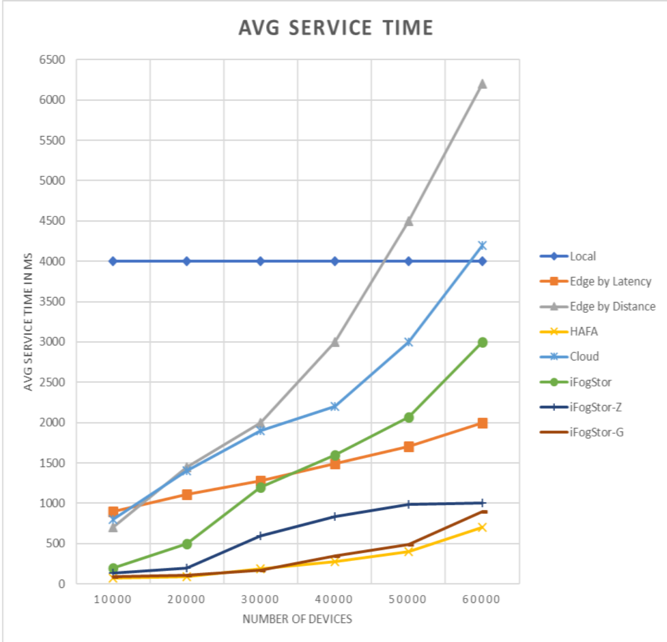

Fig 5 Average Service Time

IV. AVG Cloud VM Utilization:

For each policy, a request is sent to the cloud to calculate the best placement for the application according to the used algorithm. This leads to the occupation of the processors of the cloud and the virtual machines that are working on them. Figure 6 shows the average use of virtual machines for different placement policies. For local placement, the use was 0% because processing takes place directly on a smartphone without the need to send a request to the cloud. While (HAFA) recorded the best occupancy, the average was 10% for 60 thousand devices, due to the independent management of the Puddles. In (iFogStor-G) mode, the average utilization was 18% for the maximum number of devices. As for the policies (Edge by Distance), (Edge by Latency) and (iFogStor-z), the usage was about 35%, due to the processing capabilities required to calculate the best positioning. Whereas, the (iFogStor) and (Cloud) policies recorded the highest use of virtual machines and machines, due to the exponential increase in processing in the iFogStor policy and the need to process all requests in (Cloud) policy.

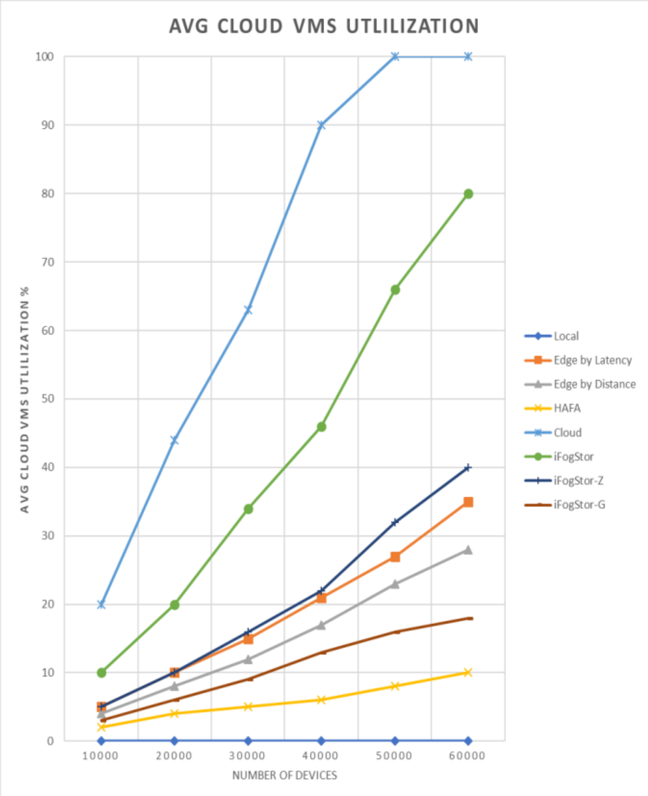

Fig 6 AVG Cloud VM Utilization

VII. CONCLUSIONS

This paper demonstrated that the use of (HAFA) and (iFogStor-G) is the best approach for the application of augmented reality in terms of service time and consumption of cloud processors and virtual machines. These policies provided the best approach for real-time application, providing the user with output within a second despite the network congestion and the increase in the number of nodes. Furthermore, these two policies reduced dependence on the cloud in processing, thus taking better advantage of the existing processing capabilities to serve other applications.

On the other hand, using the policies (Cloud), (Local) and (Edge by distance) did not yield good results due to an average service time of 5 seconds that leads to bad user experience and prevents the application from working correctly.

ACKNOWLEDGMENT

The authors wish to acknowledge the Faculty of Information Engineering at Tishreen University for their support of this research.

REFRENCES

[1] Cisco, Inc. http://www.cisco.com Accessed on Jan 6 2020.

[2] Cisco white paper,Fog Computing and the Internet of Things: Extend the Cloud to Where the Things Are. 2016.

[3] S. Shaik , S. Baskiyar. Hierarchical and Autonomous Fog Architecture. In Proceedings of the 47th International Conference on Parallel Processing Companion (ICPP '18). Association for Computing Machinery, New York, NY, USA,2018, Article 18, 1–8.

[4] M. I. Naas, L. Lemarchand, J. Boukhobza, P. Raipin, "A graph partitioning-based heuristic for runtime IoT data placement strategies in a fog infrastructure", Proc. Symp. Appl. Comput. (SAC), pp. 767-774, 2018.

[5] Thakare1, Miss. Pranita P. Deshmukh2, Miss. Rupali A. Meshram3, Miss. Komal R. Hole4, Miss. Rutuja A. Gulhane5, Miss. Nilaja A, A Review: The Internet of Things Using Fog Computing Miss. Yugandhara. International Research Journal of Engineering and Technology, March 2017.

[6] T. PATEL, K. JARIWALA, FOG COMPUTING IN IOT. Proceedings of ARSSS International Conference, 04th February, 2018, Pune, India.

[7] R. Mahmud – F. LuizKoch – R. Buyya, Cloud-Fog Interoperability in IoT-enabled Healthcare Solutions. proceedings of the 19th International Conference on Distributed Computing and Networking - ICDCN '18 – 2018.

[8] M. Mahmoud, J. Rodrigues, K.Saleem,Towards energy-aware fog-enabled cloud of things for healthcare. Computers & Electrical Engineering, Volume 67,2018,Pages 58-69.

[9] Duan, K.; Fong, S.; Zhuang, Y.; Song, W. Carbon Oxides Gases for Occupancy Counting and Emergency Control in Fog Environment. Symmetry 2018, 10, 66.

[10] M. I. Naas, P. R. Parvedy, J. Boukhobza and L. Lemarchand, "iFogStor: An IoT Data Placement Strategy for Fog Infrastructure," 2017 IEEE 1st International Conference on Fog and Edge Computing (ICFEC), Madrid, 2017, pp. 97-104.

[11] A. Buyya, S. Srirama, "Fog and Edge Computing: Principles and Paradigms", Jan 2019, John Wiley & Sons, Inc.

[12] A.Ahmed, H.Arkian,D.Battulga, Ali J. Fahs, M.Farhadi,D.Giouroukis, A.Gougeon, F. Gutierrez, G. Pierre, Paulo R. Souza Jr, M. Tamiru, Li Wu, "Fog Computing Applications: Taxonomy and Requirements", eprint arXiv:1907.11621, Jul 2019.

[13] W. Qian, "PFogSim: A Simulator for Evaluating Dynamic and Layered Fog Computing Environments", Master Thesis, Auburn University, USA, 2019.

[14] Data set du Barcelona, https://opendataajuntament.barcelona.cat/data/en/dataset, Accessed on Oct 2019.

[15] T. Zarraonandia, A. Montero, P. Diaz, and I. Aedo. 2019. "Magic Flowerpot": An AR Game for Learning about Plants. In Extended Abstracts of the Annual Symposium on Computer-Human Interaction in Play Companion Extended Abstracts (CHI PLAY '19 Extended Abstracts). Association for Computing Machinery, New York, NY, USA, 813–819, October 2019.
[16] Microsoft, https://www.microsoft.com/en-us/hololens, Access on 6 Jan 2020.