International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2020 IJSRCSEIT | Volume 6 | Issue 2 | ISSN : 2456-3307
DOI : https://doi.org/10.32628/CSEIT206221

# Computational Offloading in FOG computing using Machine Learning Approaches

**Najmus Saqib[1], Nadeem Yousuf Khanday[2]**
[1]Research Intern, National Institute of Technology, Srinagar Jammu and Kashmir, India
[2]Research Scholar, National Institute of Technology, Srinagar Jammu and Kashmir, India

## ABSTRACT

Computation offloading is a prominent exposition for the mobile devices that lack the computational power to execute applications that require a high computational cost. There are several criteria on which computational offloading can be performed. The common measures' being load harmonizing at the servers on which task is to be computed, energy management, security and privacy of tasks to be offloaded and the most important being the computational requirement of the task. That being said more and more solutions for offloading use various machine learning (ML) and deep learning (DL) algorithms for predicting the best nodes off to which task is to be offloaded improving the performance of offloading by reducing the delay in computing the tasks. We present various computational offloading techniques which use ML and DL. Also, we describe numerous middleware technologies and the criteria's that are crucial for offloading in specific developments.

**Keywords :** Machine Learning, Deep Learning, Computational Offloading, Edge Computing, IoT

## I. INTRODUCTION

We live in the age of technology which keeps on refining over time such as hardware technologies which communicate like smartphones, tablet computers, smart gear like smart glasses and smartwatches etc. Also, communication technologies are advancing like any other technology with rapid pace examples being 4G, long-term evolution (LTE), and wireless broadband (WiBro) and more recently developed 5G. New paradigms of computing are also being developed and deployed like cloud computing, fog/edge computing, Internet of Things (IoT) and semantic web all of which in tandem are responsible for the development of smart and intelligent applications. Artificial Intelligence (AI) and ML are also infused in everyday technologies.

The Internet of Things (IoT) paradigm promises to make "things" including consumer electronic devices or home appliances, such as medical devices, fridge, cameras, and sensors, part of the Internet environment. IoT as an emerging paradigm supports integration, transfer, and analytics of data generated by smart devices (e.g. sensors).

Furthermore, another paradigm namely Mobile Edge Computing (MEC) is on the rise. In 2014, more than 1 billion smartphones were sold worldwide, and more than 2 billion consumers are expected to get a smartphone by 2016 [1]. Smartphones nowadays have enough computing capacity to process various computing tasks. On average by 2020 there are going to be 6.58 connected devices per person. There are also more than a dozen sensors on current smartphones giving rise to various smart applications/services lately like cloud storage, healthcare, IoT, instant messaging(IM), augmented reality, multimedia streaming and so on[2-4]. Often the computation required for such varied tasks generated by billions of mobile users is far more than the capability of mobile computing power. Mobile phones as we know as much capable as they are, still lack resources for compute-heavy tasks also, the device usage can be constrained by limited battery life and network connectivity. Therefore, we can consider utilizing the highly available edge of fog cloud resources in addition to the device and with their mobility attribute thus making necessary to offload certain tasks to the cloud. The architecture of fog computing is shown in figure 1.

According to [5], computational offloading is the resourceful process that depends on peripheral infrastructure to execute a computational task generated by a less powered device. It is not as simple as it seems to offload a task from the device to another device (for/edge node). That is the reason the help of many ML and DL algorithms are incorporated in present offloading techniques to get optimal performance in terms of usage Network bandwidth, prediction of optimal nodes, determination of what task data size is eligible for offloading and what not.

Modern IoT applications are becoming more and more reliable on AI and ML for them to be more autonomous and intelligent. Thus making the application more sophisticated and intelligent but also increase the complexity of the application tasks. Therefore we need to efficiently execute the tasks based on the computation requirement of the application. For the said reasons it is all the more beneficial to use ML and DL for efficient offloading of tasks.

The main contributions of this work are:

- Describe numerous measures on which computational offloading is performed by current IoT middleware technologies.
- Present the middleware technologies which assist in computational offloading.
- We give an overview of various machine learning and deep learning algorithms used by lately proposed computational offloading techniques in the FOG computing paradigm.

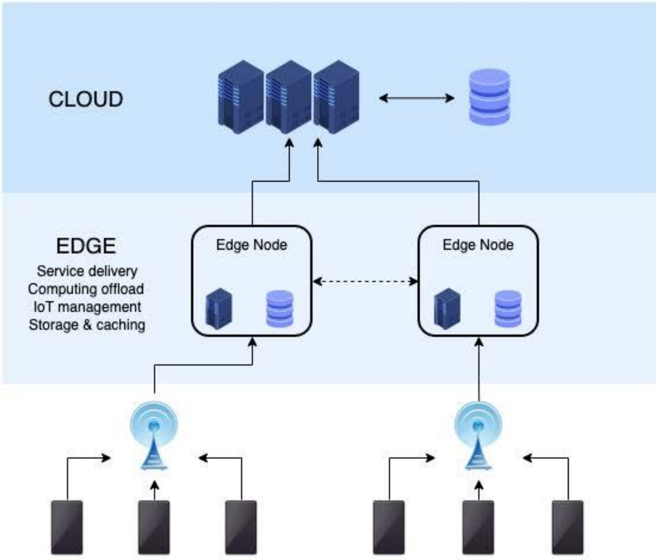


**Figure 1**. Computational offloading in fog computing.

## II. Criteria's on which computation offloading is done in the FOG computing paradigm

With the help of middleware technologies, decisions are made to offload a certain task based on various criteria's which are discussed below.

### A. Tasks requiring excessive computation

When we have a resource-hungry application running on a mobile device with low computation power, it becomes necessary to offload the task to a device which has more resources [6]. For example map applications on every device use satellite data to update the map with the help of the cloud.

### B. Delay Sensitivity

Applications which are sensitive to latency can have an adverse effect on their performance if the tasks are not executed swiftly. In such scenarios, the closest possible computing node to the receiving node must be used to offload and execute the tasks with low latency [7]. For example, a video streaming service provided by a cloud may offload and cache the content at an edge node which is close to the user equipment (UE)

### C. Load Sharing

Distribution of tasks generated by the UE's is necessary when one of the servers on which the tasks are executing is overwhelmed by the huge number of tasks within the service provider's ecosystem [8]

### D. Data management and organization

Since in fog computing data is moved frequently moved from one UE to another, it becomes important to manage and organize the user data. As the data which is seldom used by the user may not be in the immediate use at the native UE but the data may still be required in the future and as such, there is a need to save the data somehow. In such a case it is better to offload the data from the UE to a cloud [9, 10].

### E. Privacy and security

Because of privacy and security concern offloading is required if the data is secretive.For example, patients data from a hospital's local machines may be moved to a cloud which is present at the hospital and is insecure as the access is easy as compared to the data stored in the cloud. Similarly, personal data of a user may be moved to a personal mobile edge cloud [11]

## III. MIDDLEWARE TECHNOLOGIES USED FOR OFFLOADING

The decision making of offloading tasks in fog computing paradigm is done with the help of numerous middleware technologies especially for reducing energy consumption and latency. See figure 2 which shows the working of a smart gateway used for task offloading. Some of the middleware technologies are presented below.

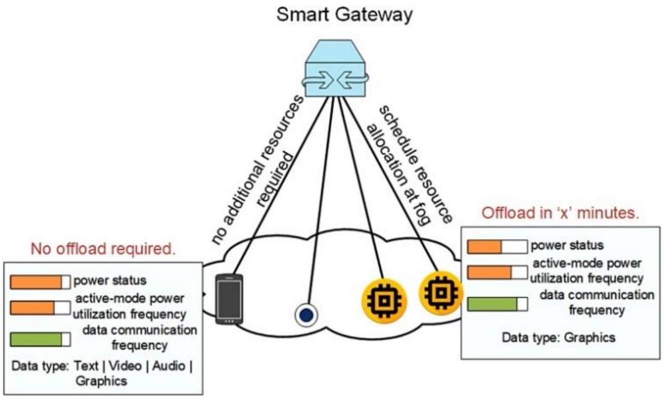


**Figure 2.** The smart gateway decides if an underlying node requires task offloading.

### A. Cloudlet

The main job of the cloudlets [12] is to deliver the much-needed computing and storage resources to the mobile devices nearby. They are highly distributed and disseminated infrastructures. Cloudlets are generally one hop away from mobile devices thus aim to bridge the gap between the cloud and the mobile device and in turn help the UE to receive the task offloading service to save battery, computation, storage etc. Other problems are aimed to be solved by cloudlets like wide-area network (WAN) latency, jitter and packet loss.

### B. Mobile edge computing

The term mobile edge computing (MEC) was first coined by the European Telecommunications Standard Institute (ETSI). It is defined by the ETSI as a resource which provides multiple services like computation and memory to a mobile device which is in close proximity it and specifically within the radio access network (RAN) [13]. The advantages of MEC are reduced latency, lower bandwidth consumption and as a result better quality-of-experience (QoE).

### C. Micro datacenter

As the name suggests these are the smaller versions of large scale data centres by being immensely portable and containerized [14, 15].The architecture of micro datacentre (MDC) usually consists of a small number of servers per rack e.g. up to 4 servers or less. Portability is the biggest plus of MDC as server capacity can be increased or decreased as per the requirement of the application in mobile devices. All these features help in setting up an MDC either indoors and outdoors and in tough terrains.

### D. Nano datacenter

Nano datacenters (NaDa or nDCs) are just like MDCs but with more capabilities in terms of memory and computation. They are always-on gateways. There are multiple applications such as video-on-demand (VoD) and gaming that can use the benefits of NaDa [16]. There are many benefits of NaDa like they minimize the traffic on the backbone and are comparatively much cheaper to set up. They deliver localized services and thus become flexible for the users and are highly scalable. The only downside of NaDa is that they have limited uplink bandwidth.

### E. Femto cloud

To utilize the computational and storage capabilities a Femto cloud is used [17]. It is an arrangement of co-located devices. The computational power of mobile devices is under-utilized much of the time. Thus offloading to these devices in areas like classrooms, public transit and so on, can yield better performance and save the cost.

## IV. Offloading techniques which use ML and Deep learning in the fog computing paradigm

The simplest offloading technique is to offload the task to a nearby cloud but this may lead to overloading of servers and cause latency [18]. In this section, we review some of the proposed offloading techniques in fog which use ML and deep learning.

Ghoneim et el.[19] discussed LSTSM powered MEC computational offloading as it has low communication delay and thus overcomes the problem of high communication delay that of a cloud. The authors stated that in the present scenario the MEC architecture has gained ground for computational offloading. They further added that due to increasing requirements and services of user equipment's simple offloading technique is naïve at best. In this context, the authors proposed a deep learning-based intelligent computation offloading based architecture which uses a multilevel LSTM algorithm for computation task prediction and depends on the size of the data. In this approach the authors assumed that the number of the device for computation offloading is done is k. The tasks are offloaded to a node which belongs to the cellular network of the device. The author's defined the data size as $V_k \in \{V_1, V_2, V_K\}$, $W_f$, $W_C$, $b_f$, and $b_C$ are used as weights and biases for forget and input gates with activation functions as σ and tanh. The hidden layer gives the output as s $h_k = \sigma( W_o \cdot [h_k{-}1, V_k] + b_o) * \tanh(C_k)$. Finally output $\tilde{V}k \in \{\tilde{V}1, \tilde{V}2, \ldots, \tilde{V}K\}$ is produced by the final layer. Thus the prediction of the task data size is the goal of the LSTM algorithm which it achieves by improving the accuracy ($|\tilde{V}_k - V_k| \propto 0$).

**Table 1:** Tabulates the characteristics of the discussed IoT middleware technologies

| Middleware | Target Nodes | Distance | Core Purpose | Scalability |
|---|---|---|---|---|
| Cloudlet | mobile | 1-hop | latency management | med-high |
| Mobile edge computing | mobile and IoT | 1-hop | latency management | med-high |
| Micro datacenter | mobile and IoT | 1 to 2 hops | latency management, task offloading | high |
| Nano datacenter | mobile and IoT | 1-hop | latency management, task offloading | low |
| Femto cloud | mobile and IoT | 0-hop | task offloading | low |

Hassan et el.[20] proposed a Deep reinforcement Q-learning model for the offloading of a task in fog computing to address the problem of far-end networks which exists between the devices and the edge cloud The far-end network produces which makes it undesirable for real-time IoT applications. The authors argued that instead of far-end networks, a near-end network would produce low latency and thus is far better equipped for offloading in fog computing. The flexibility, diversity and topographical spreading of mobile devices create a variety of problems for computational offloading in fog. Thus the authors used a deep Q-learning based autonomic framework for offloading the tasks modelling the problem into a Markov decision process (MDP). This technique was able to make better offloading decisions by minimizing the latency of service computing. This Q-learning approach for offloading also consumed less power thus making the model energy-efficient.

Junior et el.[21] tackled the issues which arise due to disregarding contextual information during computational offloading. The authors argue that when a single reasoner is used for task offloading decisions, due to the dynamic nature of the fog computing it becomes difficult to predict the accuracy of the offloading technique. Thus the whole system becomes unstable and with meagre performance. The authors provide a solution which uses contextual information to significantly improve the computational offloading in fog computing with the help of classification ML algorithms. The authors proposed a Context-Sensitive Offloading System (CSOS). The proposed system relies on ML reasoning algorithms and robust profiling to provide accurate offloading decisions. The JRIP and J48 classifiers reached an accuracy of 95% during evaluation under the simulated database of tasks. The system was also assessed under real scenarios where contextual information keeps on changing. CSOS made better-offloading decisions with low energy consumption and performance gains

Alelaiwi et el.[22] argued that in today's data-rich world mobile devices are not equipped to perform high computation processes. Far too long, due to the high computing capacity cloud platform was considered the solution for handling the compute-heavy tasks of mobile devices. The cloud data centres, being far end networks for mobile devices, increase the network delay and affects the mobile network. To overcome such issue alelaiwi introduced a deep learning algorithm for predicting whether the task is to be offloaded to a fog node in proximity or an edge node. The author used restricted Boltzmann machines learning algorithm to model a response-time-prediction framework to predict and detect the haphazardness in the resource availability thereby improving the offloading of tasks as the framework could predict the best possible edge or fog node for as offloading location.

## V. CONCLUSION

Mobile computing, IoT devices and the rapid development of communication technologies together with machine and deep learning techniques have paved the way for more interactive and autonomous services. Today billions of mobile devices providing computation are easily and cheaply accessible with the only downside being the availability in small pieces. Therefore to compute certain tasks which require higher computation power offloading is done. ML and DL techniques are also applied to better predict where to offload the tasks. Computational offloading can be implemented on various criteria's such as load balancing, delay or latency in the network, security, privacy, data and energy management and so on. In this paper, we described many criteria's on which computational offloading is performed in many pieces of literature in fog computing. We reviewed some of the IoT middleware technologies that are being been used for offloading the tasks. Finally, we discussed several ML and DL powered offloading techniques in fog computing.

## VI. REFERENCES


[1]. Buyya, Rajkumar, and Amir Vahid Dastjerdi, eds. Internet of Things: Principles and paradigms. Elsevier, 2016.

[2]. Abdelnasser, Heba, Khaled A. Harras, and Moustafa Youssef. "UbiBreathe: A ubiquitous non-invasive WiFi-based breathing estimator." Proceedings of the 16th ACM International Symposium on Mobile Ad Hoc Networking and Computing. 2015.

[3]. Abdelnasser, Heba, Moustafa Youssef, and Khaled A. Harras. "Wigest: A ubiquitous wifi-based gesture recognition system." 2015 IEEE Conference on Computer Communications (INFOCOM). IEEE, 2015.

[4]. Abdelnasser, Heba, Moustafa Youssef, and Khaled A. Harras. "Magboard: Magnetic-based ubiquitous homomorphic off-the-shelf keyboard." 2016 13th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON). IEEE, 2016.

[5]. Flores, Huber, et al. "Social-aware hybrid mobile offloading." Pervasive and Mobile Computing 36 (2017): 25-43.

[6]. Masip-Bruin, Xavi, et al. "Foggy clouds and cloudy fogs: a real need for coordinated management of fog-to-cloud computing systems." IEEE Wireless Communications 23.5 (2016): 120-128.

[7]. Huang, Cheng, Rongxing Lu, and Kim-Kwang Raymond Choo. "Vehicular fog computing: architecture, use case, and security and forensic challenges." IEEE Communications Magazine 55.11 (2017): 105-111.

[8]. Osanaiye, Opeyemi, et al. "From cloud to fog computing: A review and a conceptual live VM migration framework." IEEE Access 5 (2017): 8284-8300.

[9]. Elgazar, Ali, et al. "Towards intelligent edge storage management: Determining and predicting mobile file popularity." 2018 6th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud). IEEE, 2018.

[10]. Aazam, Mohammad, Eui-Nam Huh, and Marc St-Hilaire. "Towards media inter-cloud standardization–evaluating impact of cloud storage heterogeneity." Journal of Grid Computing 16.3 (2018): 425-443.

[11]. Aazam, Mohammad, Eui-Nam Huh, and Marc St-Hilaire. "Towards media inter-cloud standardization–evaluating impact of cloud storage heterogeneity." Journal of Grid Computing 16.3 (2018): 425-443.

[12]. Satyanarayanan, Mahadev, et al. "The case for vm-based cloudlets in mobile computing." IEEE pervasive Computing 8.4 (2009): 14-23.

[13]. Al Agha, Khaldoun, Guy Pujolle, and Tara Ali Yahiya. Mobile and wireless networks. John Wiley & Sons, 2016.

[14]. Alsaffar, Aymen, et al. "An architecture of IPTV service based on PVR-Micro data center and PMIPv6 in cloud computing." Multimedia Tools and Applications 76.20 (2017): 21579-21612.

[15]. Aazam, Mohammad, and Eui-Nam Huh. "Dynamic resource provisioning through fog micro datacenter." 2015 IEEE international conference on pervasive computing and communication workshops (PerCom workshops). IEEE, 2015.

[16]. Valancius, Vytautas, et al. "Greening the internet with nano data centers." Proceedings of the 5th international conference on Emerging networking experiments and technologies. 2009.

[17]. Habak, Karim, et al. "Femto clouds: Leveraging mobile devices to provide cloud service at the edge." 2015 IEEE 8th international conference on cloud computing. IEEE, 2015.

[18]. Tong, Liang, Yong Li, and Wei Gao. "A hierarchical edge cloud architecture for mobile computing." IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications. IEEE, 2016.

[19]. Miao, Yiming, et al. "Intelligent task prediction and computation offloading based on mobile-edge cloud computing." Future Generation Computer Systems 102 (2020): 925-931.

[20]. Alam, Md Golam Rabiul, et al. "Autonomic computation offloading in mobile edge for IoT applications." Future Generation Computer Systems 90 (2019): 149-157.

[21]. Junior, Warley, et al. "A context-sensitive offloading system using machine-learning classification algorithms for mobile cloud environment." Future Generation Computer Systems 90 (2019): 503-520.

[22]. Alelaiwi, Abdulhameed. "An efficient method of computation offloading in an edge cloud platform." Journal of Parallel and Distributed Computing 127 (2019): 58-64.

**Cite this article as :**

Najmus Saqib, Nadeem Yousuf Khanday, "Computational Offloading in FOG computing using Machine Learning Approaches", International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN : 2456-3307, Volume 6 Issue 2, pp. 82-88, March-April 2020. Available at doi : https://doi.org/10.32628/CSEIT206221
Journal URL : http://ijsrcseit.com/CSEIT206221