EAI Endorsed Transactions
on Energy Web

Research Article EAI.EU

A Survey of Architecture, Framework and Algorithms for Resource Management in Edge Computing

S. Premkumar[1,*], A.N. Sigappi[1]

[1]Department of Computer Science & Engineering, Faculty of Engineering & Technology, Annamalai University
premambal@gmail.com, an.sigappi@gmail.com

Abstract

Internet-based applications predominantly use the existing method of acquiring the computing resources remotely from the cloud data centers. This method of computation is not applicable in future since it is expected that the latencies in communication tend to expand largely due to the internet connectivity among billions of devices. This enormous expansion in latencies induces an adverse impact in the Quality of Service (QoS) and Quality of Experience (QoE) parameters. Edge computing is an imminent computing methodology that deploys the decentralized resources present at the edge of the network to make data processing within the proximity of user devices like smartphones, sensors or wearables. This approach is contrary to the conventional methods of utilizing centralized and distant cloud data centers. Managing the resources becomes a major challenge to be approached due to the diverse and rapidly evolving resources in comparison with the cloud. The lucrative role of Internet of Things (IoT) and Edge, and the challenges posed by the dynamic technologies are presented. This paper presents a survey of the research publications from edge computing from 2013 to 2020, covering the various architectures, frameworks, and the fundamental algorithms involved in resource management in edge computing.

Keywords: Fog computing, Edge computing, Management of resources, Architectures, Frameworks, Algorithms, Smart Agriculture.

Received on 13 October 2020, accepted on 01 December 2020, published on 23 December 2020

doi: 10.4108/eai.23-12-2020.167788

1. Introduction

An appropriate medium for information exchange is facilitated by the growth of Internet-based wireless communication methodologies that has brought more convenience in people's regular life. The worldwide range of mobile terminals is expanding to about 2.8 billion since 2019. With recent advancements in the field of data science and artificial intelligence, the usage of lightweight, smart devices has attained a massive growth. The interconnectivity among devices, known as things, has paved the way for the evolution of Internet-based wireless communication networks [1]. It also indicates the arrival of the Internet of Things (IoT) era bundled with huge sensing components, mobile devices and actuators utilized at the network edge. For every second, these interconnected devices generate a substantial volume of data.

The traditional methods for storage and computing of data are not suitable for the IoT scenario. So, a decentralized big data computation model is required. Cloud computing enables numerous services for examining and storing of data so that the end users can avail the maximum benefit of an IoT infrastructure. Furthermore, cloud computing is facing several challenges in real-time applications. A recent write-up from Cisco states that by 2021 the worldwide monthly mobile data traffic will reach up to approximate 49 exabytes and the consolidated yearly growth rate would be 47% from 2016 to 2021 [2]. There are certain computing tasks in need of timely and context-specific computing like augmented reality, virtual reality and industrial control in which data is generated in enormous amounts by the devices. It results in the computation of huge data traffic that has a key role in the

[*]Corresponding author. Email: premambal@gmail.com

upcoming Internet-based wireless communication systems. The two performance indices, viz increased data rates and decreased delivery latency are the important deciding factors in wireless communication networks based on the Internet. It indicates that high powered processing devices requires the execution of heavy traffic in data, and shifting of data traffic with transmission links having high data rate for the wireless communication networks that rely on the Internet. With a view from wireless networking systems, ultra-dense networks, heavy multiple-input multiple-output (MIMO), high-frequency communications is considered as hopeful means in satisfying the expanding needs of upcoming wireless networking including 5G wireless systems. However, several computing architectures and standards are innovated to enhance the robustness of traffic handing in data from the Internet's perspective. There arises essentiality for dramatic modifications in computer networking, computation, application and storage modes to satisfy the growing needs of the information society. Edge computing for IoT applications requires decentralised cloud and low latency computing, surmounting resource limitations of front-end devices, smart computation techniques, dealing with data explosion and network traffic and sustainable energy consumption for enhance the quality of services in terms of computing and communication specifically for real time application.

1.1. Merits and demerits of edge computing

To mitigate the commonly encountered problems in cloud computing, such as high latencies with poor bandwidth utilization, an auxiliary computing model is devised that connects the sensors and user devices with computing resources for data computation [3, 4]. The setup decreases the volume of data transferred to the cloud, thereby lessening the latencies in communication. The present research trend is towards localizing few processing resources present in huge data centers by sharing the network edge nearer to end-users and sensors. These resources are in either of these two forms: (i) exclusive "micro" data centers situated within public/private infrastructure conveniently and safely or (ii) Internet nodes having intensified computing power in routers, switches and gateways. A computing model utilizing the resources present in the network's edge is denoted as "Edge computing" [5, 6]. The architecture of edge computing is depicted in Figure 1.

When the edge resources and the cloud are utilized altogether, it is mentioned as "fog computing" [7, 8, 9]. Resources at the edge differ from cloud resources in the following ways: (i) Because of constrained resources and a restricted power budget in edge devices. (ii) Heterogeneity among the processors with varying architectures, and (iii) dynamic workload balancing tends to change often and in applications battling for the restricted resources. Accomplishment of IoT use cases like smart city and smart agriculture is challengeable, specifically in managing the resources of fog and edge computing. The salient features of edge and cloud computing are depicted in Figure 2.

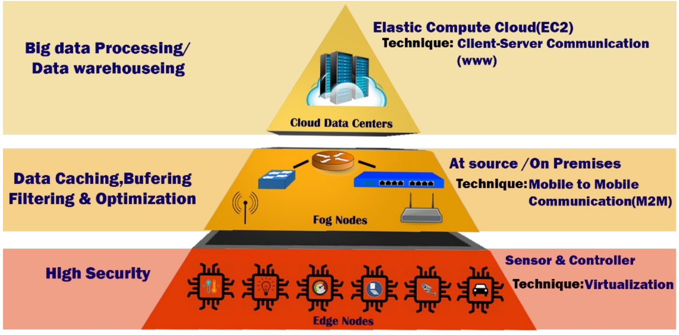

Figure 1. Architecture of Edge computing

The objective of this work is to present a survey that is focused on the fundamental architectures, underlying infrastructure and the algorithms supporting management of resources in fog/edge computing. This article is arranged as follows: Resource management is discussed in Section 2. Architectures, namely the architectures of data flow, control, and tenancy. The infrastructure used for resource management, such as the hardware, device software, and middle ware used, is presented in Section 3. The underlying algorithms, such as discovery, bench marking, load, are illustrated in section 4. Placement, and balancing. Section 5 proposes and ends the article with possible directions.

2. Computational standards

Several computational standards were put forth to deliver resource-effective services on a timely basis to approach the massive data computation in recent years. This section deals with the various computational standards developed in the past few years, such as Manifest Computing (MC), fog computing, Edge Computing, and cloudlet.

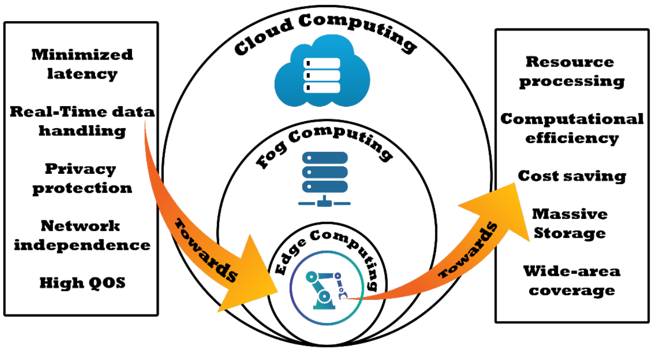

Figure 2. Salient Features of Edge and Cloud Computing

These are confronted with certain common problems with regard to handling and managing a large number of computations, networking and storing tasks caused by heterogeneous devices. They also possess multiple features because of indigenous driving forces, like persuasive computation for MC and IoT applications in fog computing. To alleviate the issues relating to resource management, several computational standards are required to conceal the heterogeneous nature of various devices.

2.1. Manifest Computing

It was developed to split the operating systems and software, from the diversified hardware in IoT devices [10]. It explicitly assists the users by hiding the details of service provisioning [11]. MC qualifies the devices to select on-demand services through networks, without concerning the service provisioning details of promoting and managing the software. MC has the following characteristics:

(1) In a client-server model, MC combines the distributed devices across the network as a single system based on some logic. The system works smartly by delivering services following device capability and network constraints. The server side is entirely dedicated to managing the resources in a centralized manner to provide robust services to users within the network.

(2) MC facilitates the logical separation of software from the hardware by loading and executing of on-demand service in diversified client devices. With the help of high-speed networks, dynamic execution of cross-platform services from servers in remote locations is qualified in client devices.

(3) For utilizing the entire computational resources present in the client devices, MC capacitates the client devices in acquiring on-demand remote services from the server-side. Block-streaming way is used to run them locally. Block-streaming execution is the process of loading and executing the required part of codes alone that is related to the user-specified service instead of loading the entire software on client devices. As a result, energy efficiency and service provisioning delay can be upgraded to a considerable extent.

Recent acknowledgement of MC benefits in the field of IoT enforced a progressive solution to control a large volume of software for highly diversified hardware infrastructure, extending from huge expensive servers, laptops, and smartphones to low-end sensors.

2.2. Fog Computing

In 2012, Fog computing [12] was conceptualized by Cisco that resembles the mobile edge computing concept in some features. Fog is an unique network processing framework that imparts computing capabilities in the network edge [13]. Initially, Fog computing was developed to meet the demands like instant feedback and location awareness in addition to portability support and wireless access for IoT environment. Furthermore, flexibility in services is ensured by fog computing with the help of n-tier architecture. This feature spotlights the entire network devices through the data routing path are capable of providing functionalities like data computation and storage of services for end devices. Physical placement of fog tier amidst the cloud and the IoT devices improves computation, storage and networking resources pooling. This fog tier comprises of a huge range of several nano servers, extending from exclusive components like setup boxes, edge routers to changeable devices like smartphones, high-end sensors, and vehicles [14].

Real-time management and analysis of long-term batch data are needed to implement fog computing in normal application areas. It is mainly performed to enhance the immediate system performance and to acquire awareness about business procedures adjustment that makes the fog servers and the cloud server to interact. Concurrent control challenges the closely packed decentralized fog servers to decrease latency in services, and the batch data analytics comes under the mastery of cloud servers with enormous processing and storage resources. The Openfog Consortium deals with upgrading the evolution of fog computing by the topmost IT firms such as Intel, Microsoft, Cisco, ARM, Dell, and Princeton University. Several options for using fog computing are indicated in [14], extending from intelligent traffic lightweight systems to smart grid and wind farms [13,15]. The enhanced futuristic growth of fog computation is covered in Vaquero and Rodero-Merino‘s work [15]. [16] defines fog computing and some application areas and also focuses on the constraints in designing and implementing fog computing system. [17] declares that fog computing is presumed to be a general platform for demanding IoT applications. [18] explains the benefits of fog computing for IoT and outlines several applicable areas appropriate for fog computing. [19] postulated 10 questions and delivered the essential answers to exhibit the benefits of fog computing in contrasting with other conventional computing platforms. In recent times, a brief set of assessment criteria for fog computing is presented in [13].

2.3. Edge Computing

It is a distributed computing standard to compute and store data nearer to the location where it is required. It enhances response times and saves bandwidth. Content generating networks were formed in the late 1990s, which serve as the origination of edge computing to help in web and video content from edge servers that were utilized nearer to users. The newly emerged edge computing methodologies considerably scale this approach by the usage of virtualization technology that makes it effortless to execute a broader range of applications on the edge servers. The growth of IoT devices at the edge of the network is generating a huge volume of data to be processed at data centers, forcing network bandwidth needs within the range [20]. Even though there are certain improvements in network technology, data centers are unable to assure allowable transferring rates and responding times, which is a crucial need for most applications [21]. In addition to this, edge devices continuously take incoming data from the cloud, pushing the firms to distribute data and service provisioning by creating content delivery networks and thereby accelerating the physical accessibility to the end-user. Edge computation involves the utilization of mobile phones, intelligent objects or network gateways to carry out tasks and provide services on account of the cloud by shifting the computation away from data centers and near the edge of the network [22]. It becomes possible to provide service delivery, content caching, storage, and IoT management by shifting the services to the edge, leading to finer transfer rates and response times. But decentralizing the logic among different network nodes brings out new crisis and demands.

Extensibility

Extensibility in a decentralized network is confronted with various issues. The foremost is it should consider the diverseness of the devices, having varying performance and energy restraints, the ever-changing conditions, and the dependability of the connections, compared to the more dominant framework of cloud data centers. Moreover, security needs may bring more latency in the node communication and resulting in slowing down the scaling process.

Dependability

Restoring a service alive by managing the failovers is a significant task. The users should be able to access a service without interruptions even if a single node gets down and becomes not reachable. In addition to this, edge computing systems must perform certain actions to recover from a failure and intimating the occurrence to the user. To fulfil this aim, every device should sustain the network topology of the whole decentralized system that makes the finding of errors and recovery to be easily applicable. Other features that may impact this aspect are the current technology in use, creating several levels of dependability, and the correctness of the data generated at the edge that could be undependable because of specific environmental conditions. Table 1 shows the advancement of Edge computing in data store technology.

Speed

Edge computing introduces analytical, processing resources nearer to the end-users and thereby aids in leveraging the speed in communication. A well-framed edge platform would exceed a conventional cloud-based system.

Efficiency

Due to the nearness of the logical resources to the end-users, modern analytical and artificial intelligence tools can execute on the edge of the system. This setup at the edge assists in increasing operational effectiveness and becomes more advantageous.

Application

Edge application services reduce the size of shifted data, the resulting traffic, and the distance travelled by the data. It lessens the latency and transmission costs. Computation offloading for current applications, like face recognition algorithms, showed improvements in response times to a greater extent, as depicted in early studies. Detailed research proved that utilizing resource-rich machines called cloudlets near mobile users to provide services generally present in the cloud, created progress in running time when few tasks are offloaded to the edge node. On the other hand, because of transfer times between devices and nodes, offloading each task causes a slowdown. Therefore, an optimal configuration can be framed depending on the workload. A doctorate researcher and data scientist at the University of Oulu in Finland describes that when comparing centralized computing and data storage, moving intelligence to the edge could also fundamentally reduce the cost. In edge computing, specialized chips and cloudlets, primarily micro-clouds or ad hoc clouds that would operate in a home, company, or vehicle, could monitor design complexity [13]. Table 1 summarizes the advancements in edge computing solutions reported in research papers.

2.4. Cloudlet

A research team from Carnegie Mellon University (CMU) first developed the cloudlet concept in 2009 [38]. The term “cloudlet” denotes the proximal placement of micro data centers to mobile users, for instance inside a classroom or a restaurant. One of the motivating factors in forming the cloudlet is for enhancing the interactional performance of mobile apps, with special attention to the ones with firm needs on consecutive latency and jitter which insist on having a responding delay in the range of milliseconds, that raises network complexity in Internet. The proximal placement of cloudlets allows the servers to deliver the mobile users with extremely responsive cloud functionalities.

To provide prompt service, this ordering imposes the owners’ privacy by modifying the private data before issuing to the cloud [39]. The CMU research group released a sequence of works after the innovation of cloudlet that concentrates in two concerns such as finding the most suited cloudlet applications [40] and enhancing user portability by developing virtual machines [41]. [40] proposes cognitive assistance based on real-time. i. e “killer app” like speech and face recognition for the cloudlet, that insists prompt feedback to assist users' attentiveness unnoticeably.[41] states the overlay functioning in discovering the cloudlets and hand-off in the mobile environment based on open-source ecosystem for cloud computing. The interchange of data among the cloudlets is also decreased.

The open edge computing initiative is created by CMU and various topmost industrial firms, involving Intel, Nokia, Deutsche Telekom, Crown Castle and Vodafone due to the potentiality imposed by cloudlets and availability of inherent business opportunities [42], for developing major methodologies encircling cloudlets and managing user acceptance testing. Accessing distant computing services from cloud data centers is the existing framework for several Internet applications. User-generated data like wearables and smartphones, or sensors in a smart city or company, is generally transmitted to geographically far away clouds for processing and storage. This method of computation is not applicable in future since it is expected that the latencies in communication tend to expand largely due to the internet connectivity among billions of devices [43]. This enormous expansion in latencies induces an adverse impact in the Quality-of-Experience and Service parameters [44]. To overcome the problem, an alternative computing model is used to acquire processing resources nearer to sensors and user devices and utilize those [45,46] partially for data computing. It helps in decreasing the volume of data transferred to the cloud, thereby decreasing communication latencies. The present direction in research to realize this computing model is to distribute few processing resources present in enormous data centers by issuing them closely to end-users and sensors in the edge of the network. The dimensions in edge computing resource management is depicted in figure 3.

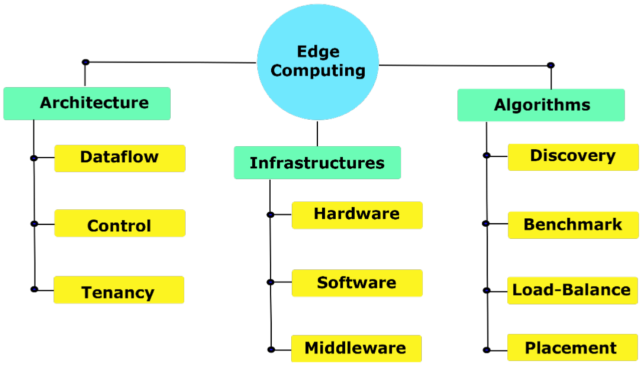

Figure 3. Edge computing resource management

The resources are manifested as (i) exclusive "micro" data centers which are comfortably and securely situated inside public/private framework or (ii) Nodes on the Internet, like gateways, switches, routers increased with potentials to compute. A computation model utilizing the resources present at the network edge is denoted as "edge computing" [47, 48]. A paradigm utilizing the edge resources as well as the cloud is called "fog computing" [49, 50, 51]. In contrast to the resources in the cloud, resources in edge are: (i) resource-restrained - restricted computing resources, as edge devices possess smaller processors and restricted power assignment, (ii) processors are diverse natured - with varying architectures (iii) rapidity in corresponding workloads changes, and resources need to be used by the applications in a restricted manner. Therefore, resource management is the most important issue in both fog and edge computing. According to the time varying demand, the resources are dynamically allocated by the cloud and edge sources. These distributions include networking, workload and the analytical resource distribution with the combination wireless network communication and edge computing in time varying mechanism. The Quality of Service (QoS) metrics in resource allocation has to be met with service differentiation and context awareness to have a better resource utilization in edge and cloud. The elasticity in network resource utilization have a feasible solution named Linear parameter varying modelling of cloud applications combined with set-theoretic controller [50]. The objective of this paper is to analyse and present the architectures, frameworks, and algorithmic methodologies in support of managing the fog/edge computing resources.

3. Architecture classification in Edge computing

The three factors, namely, data flow, control, and tenancy, serve as the basis of classifying the architectures for management of resources in fog/edge computing.

- *Data flow architectures:* It is directed towards the workload's movement and data inside the processing ecosystem. For instance, workloads are portable to the edge nodes from the user devices or cloud servers to the edge nodes.
- *Control architectures:* It is dependent on the way of controlling resources in the processing ecosystem. For instance, various edge nodes are managed by a principal algorithm or a central controller. The distributed approach is another option that can be used.
- *Tenancy architecture:* It relies on the reinforcement given for providing numerous entities in the computing ecosystem. An edge node can host a solo application or a group of applications.

3.1. Data flow in edge computing architecture

The data flow architectures work on the principle of finding the way of transferring the data or workloads inside a fog/edge computing environment. The types of data flow architectures reported in literature are: aggregation, sharing, and offloading. Figure 4 depicts the edge computing data flow model.

Aggregation

It works based on having an edge node acquiring data produced from several end devices. The acquired data is partly processed for filtering or pruning. The objective of aggregation is to minimize the overheads in communication, including the prevention of needless traffic transmitted far away from the edge of the corresponding network. In general, research work on this model can be differentiated based on the following :(i) Modelling and aggregation implementation techniques, (ii) Aggregation improvement techniques.

Table 1. Advancements in edge computing solutions

Description of the Edge Computing Solution	Use case Domain	Advantages	Drawbacks	Reference
The hardware components needed to devise wearable devices and integrating in Internet of Medical Things (IoMT), Textile industry in edge computing are been discussed here.	IoMT , Textile industry	Execution of light weight tasks, data quality, transmission bandwidth, data offloading, data storage, data processing, and IoT management	Technological issues: interoperability, standard, protocol and different manufacturer. security issues: patient privacy, bluetooth does not support encryption	[23]
The design of edge computing framework here is developed to fulfil demanding requirements of IoT-enabled time and mission-critical application namely, early fire detection. Three-level Cyber-Physical Social System (CPSS) is designed in which the middle layer is integrated with dynamic resource scaling mechanism for early fire detection.	Fire identification & detection	Fire detection using computer vison edge, dynamic resource allocation, offloaded requests, QoS performance, energy consumption, and Transmission latency	Deployed in homogeneous servers, not applied in real-time scenario.	[24]
The Multi-Agent System (MAS) MAS-based framework is been developed and proposed to enhance the functioning of the manufacturing domain, whereas edge computing is concerned with load balancing in computing. The proposed model with MAS architecture is been evaluated with throughput in simulation environment proving to be better than cloud architecture.	Manufacturing, Industries	Collaborative operation, low latency communication channel, reduction of service delay, and data congestion is avoided	Distribution of task, computing resources allocation, and low bandwidth allocation	[25]
An affordable, fast early warning system (AFEWS) is proposed with utilizing edge devices. The edge computing device is been installed near the data source to identify and analyse the normal and abnormal functioning using hybrid fault model to predict the working process. Edge computing has been adopted in manufacturing scenario for monitoring the well-being of machine, enhancement of productivity in assembly line and intensifying quality and quantity.	Automobile, Manufacturing	Response time, exact fault identification, CPU usage and lower network delay	Power consumption and task allocation parameter is missing	[26]
Edge computing layer is created not only to analyze the data stream set evolving from a car data stream and also to make a real infrastructure using Raspberry Pi and Node-RED server. Data pre-processing is done in the edge device by detecting an event and filtering through anomaly detection.	Automobile	Processing & aggregating large data volumes, low latency and security protocols	Not able to handle the high streaming of data per millisecond and heating up of raspberry pi	[27]
The architecture is oriented towards Open Source big data technologies to carry out	IoMT	Anomaly detection in	Better detection is dependent on the	[28]

real-time analysis of wearable sensor data streams. The proposed architecture is comprised of 4 differnt layers: a sensing layer, a pre-processing layer (Raspberry Pi), a cluster processing layer (Kafka's broker and Flink's mini-cluster) and a persistence layer (Cassandra database).		real-time, technology independent communication, replacing framework, and adding new nodes	processing memory of local edge	
The proposed system was deployed into the field using tools such as Arduino, Raspberry Pi 3 Model B, Python, Django Web Framework, and Standard 433 MHz RF link.	Agriculture	Easy setup, cost-effective and low-level implementation	Half-duplex architecture, cannot fulfil immediate response, low bandwidth, single real-time thread execution	[29]
The proposed system is a base line to manage the soilless culture requirements in recirculation having moderately saline water with 3-tier architecture. Cyber-physical system interacts with the crop, and the edge layer takes care of connection and failures. The cloud module FIREWARE takes care of data.	Greenhouse, Hydroponics	Effective control of environmental data, water consumption, nutrients validation, and crop cycle estimation	System is costly and needs regular maintenance	[30]
Two perspectives considered are: a device (or Thing) and a functional perspective of users with virtual devices are enabled to aggregate the capabilities of IoT devices, derive new services by inference, delegate requests/responses and generate events.	Virtual Devices	Siloization, network bandwidth, latency, speed, semantic interoperability	No real-time implementation	[31]
An Edge-IoT platform and a Social Computing framework is proposed to build a system aimed at smart energy efficiency in a public building scenario.	Smart Energy	High computing, network resource cost, and energy efficiency	Security, traceability and data integrity	[32]
The architecture consists of three modules: (i) the Sensor Interface Module (SIM), (ii) the Communication Module (CM) and (iii) the Measuring Engine Module (MEM).	Engine	Software overhead was negligible with less CPU busy time	Fewer trails are considered	[33]
A concerted tracking methodology for manufacturing resources was proposed by appropriating IoT and edge computing technology. It facilitates automatic tracking and tracing of the location underpinned by Bluetooth signals and edge gateways. The entire system was executed in an air conditioner manufacturing unit situated in an industrial park to validate the feasibility and efficiency.	Manufacturing, Industries	Accuracy and maintain effectiveness in decision making at edge, lower transmission delay, low latency	Data tracking is limited	[34]
The experiment reveals that refining raw IoT data at the edge devices is efficient with respect to latency and imparts circumstantial awareness for the smart city decision-framers in a logical manner.	Smart City	Low latency, effective decision-maker	Huge data analysis	[35]
The architecture to implement the active maintenance contains device, network, data and application domain which acts on connection failure and mapping of equipment.	Manufacturing	Information fusion, cooperative mechanism, bandwidth optimization and response latency	Security mechanism is required	[36]
A data collection framework in Wireless Sensor Network (WSN) using edge computing was proposed for agriculture in which for each task allocation, double	Agriculture	Quality data, data collection time, low Latency, energy consumption,	Not compared with IoT scenario	[37]

selection strategy has been applied to filter out the node and sensor data.	low waiting time

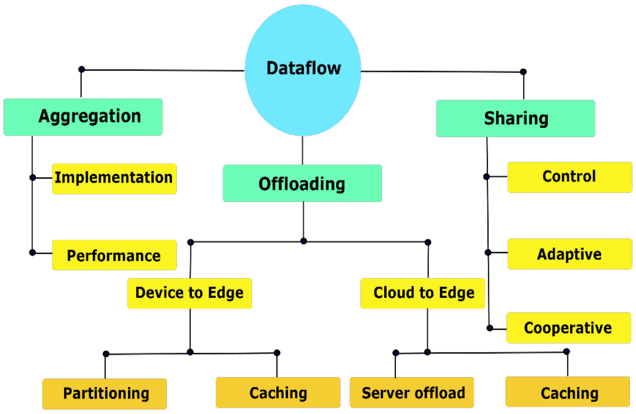

Figure 4. Edge Computing data flow Model

i. Modelling and aggregation implementation techniques

The fundamental methods executed for reinforcing aggregation formed a major portion in Wireless Networking [52] and distributed processing of data stream [53]. Dense and large-scale sensor networks are unable to route the entire sensor generated data to a centralized server. In-network data aggregation is explained as utilizing the intermediary nodes to aggregate the data along the data path [54].

a) Techniques based on Graphs

Directed graph-based and Tree-based methods used for data aggregation is explained here.

(1) Techniques based on Tree concept

Tree-based techniques encompass two forms such as Spatial index trees and Data Aggregation Trees (DATs). DATs are utilized for aggregation of WSN using Probabilistic Network Models (PNMs) or Deterministic Network Models (DNMs). Recently, the usage of PNMs over DNMs is drawing the attention in research for making logical conclusions of lossy links in the network using tree-oriented methods in balancing the load [55]. Spatial index trees are applied in making query inside networks but also addressed for aggregation in recent times. An energy-effective index tree named EGF-tree is applied for collecting and aggregating the data [56]. Table 2 presents some of the applications of edge computing.

Table 2. Applications of Edge Computing

Reference	Description
[57]	mF2C is a European project to design, execute and evaluate the unique management for a synchronized fog and cloud computing system
[58,59]	Nippon Telegraph and Telephone Corporation (NTT) created an edge accelerated web platform to decrease the response time for cloud applications
[60,61]	Android apps with cognitive assistance to be more efficient to decrease the response time for cloud applications
[62]	Cisco Kinetic platform provides fog processing module to process IoT data
[63]	Vortex platform allows connecting each vortex edge device to an IoT device to construct global shared fog domain
[64]	Cisco IOx platform offers edge features to develop IoT applications
[65]	FogSim simulator allows the manipulation of edge features
[66]	ParaDrop is an edge computing platform for wireless gateways and applications
[67]	Implementation of cloudlet based on Software Defined Networking (SDN) for patient monitoring
[68]	A converged wireless access architecture based on edge servers for heterogeneous cloud access networks
[69]	Managing and configuring heterogeneous IoT networks using UbiFlow framework

(2) Methods based on Directed Graph

The programming model in dataflow and WSN applications use a directed graph. Distributed Dataflow (DDF) programming model used in fog computing [70] works on the Message Queuing Telemetry Transport (MQTT) protocol, which makes the workflow easier in various nodes by assuming the diversity in devices [71].

b) Techniques based on Graphs

The programming model in dataflow and WSN applications use a directed graph. Distributed Dataflow (DDF) programming model used in fog computing [70] works on the Message Queuing Telemetry Transport (MQTT) protocol, which makes the workflow easier in various nodes by assuming the diversity in devices [71].

c) Methods based on Clusters

Effectiveness in energy is a major aspect of node clustering in cloud-based methods. A node from each cluster is selected as the cluster head. It is accountable for making aggregation process in each cluster locally and transferring the aggregated data to other node. Data aggregation is made energy-effective by clustering methods is also stated [72].

d) Methods based on Petri Net

Recent studies draw the attention of using the High-Level Petri Net (HLPN), called as RedEdge, for modelling the aggregation in Edge-oriented systems [73]. HLPN makes heterogeneity easier, and it is authenticated by verifying fulfilment by an automated solver. This framework was analyzed for a smart city implementation and validated for different effectiveness metrics like power, latency, and memory usage.

e) Decoupled methods

The typical aggregation methods discussed above generally possess heavy incorrectness if data is lost in the network. The aggregation method finds the routing data path. Synopsis Diffusion (SD) is a method to decouple the routing process from aggregation and thereby optimizing separately to improve correctness [74].

f) Batch methods

Batch method is an algorithm which is used in computing for external data stream. The data produced from different sources is transferred to a node where it is clustered periodically to a batch job and gets executed on the node.

g) Hybrid methods

Hybrid methods relate one or more of the techniques mentioned above. It decreases the loss rate and presents only a few communication errors while enhancing the network's overall effectiveness.

ii. Aggregation improvement techniques

Aggregation is applied to reform various goals in the computational environment. The efficiency of communication is expressed concerning the latency, bandwidth and energy restraints. Therefore, the goals range from the efficiency in communication to the definite quality of aggregation (or analytics) which is carried out on the edge node. After exploring the conventional works on methods to improve aggregation, the following methods are obtained and explained below.

a) Efficiency-aware approaches

Efficiency-aware approaches are classified into three methods: 1) For optimization of bandwidth 2) For minimization of latency 3) For reduction of energy utilization.

(1) Optimization of bandwidth

Three modules are present in the Bandwidth Efficient Cluster-based Data Aggregation (BECDA) algorithm [75]. First module concerns with arranging the distributed nodes into many clusters. The second module deals with choosing a cluster head for each cluster and then aggregating data from within the cluster. The last module makes every cluster head to promote for intra-cluster aggregation. It uses bandwidth effectively for aggregating the network data and systematic than forerunner techniques.

(2) Latency-conscious

Latency is one of the significant benchmarks frequently examined in edge-based systems for aggregation [76, 77]. It can be reduced by proposing an intermediary architecture concerning data services [78].

(3) Energy-conscious

Recent studies in the energy effectiveness of data aggregation concentrate in reducing the network's power consumption by creating separate nodes that are energy effective through software and hardware methods. The data aggregation algorithm which is used a consecutive interference cancellation method to reaches lower limits of latency and maintains constant energy at the same time. RedEdge is an edge computing paradigm that works on mobile devices in which the consumption of energy for data transfer is reduced [73].

b) Quality-conscious Techniques

An approach where end devices data are compulsorily transferred for decreasing overheads to a node is referred to as selective forwarding. "Quality-aware" is the creation of functional decisions to improve predictive analytics quality in selective forwarding [79].

c) Security-aware Techniques

Security and privacy in identity are the important factors for the process of aggregation in the edge node between a public cloud and user devices. An Anonymous and Secure Aggregation (ASAS) scheme [80] utilizing elliptic curve public-key cryptography, a linearly homomorphic cryptosystem and bilinear pairings termed as the Castagnos-Laguillaumie cryptosystem [81], is created inside fog environment.

d) Heterogeneity-conscious Techniques

Diverseness is a natural aspect of edge-oriented environments. [82]. Diverseness of resources is a source to various kinds of fog/edge nodes, comprising architectures of CPU, a union of allocated micro data centers (CPU-oriented systems), and traffic control devices like routers, switches, base stations at the Edge of the network. [128]. There is a possibility of using diversified resources in a unique fog/edge computing model at the same time, and there exists a little proof of such type of completely implemented system.

Sharing model

It is normally deployed if the workload is shared among peers in contrast to the aggregation model. This model fulfils the processing needs of a workload by directly offloading in battery powered peer devices rather than offloading inside the cloud. As a result, network with dynamicity is established, provided that devices tend to connect and quit the network without an intimation. Pragmatically achievable schemes developed for coordinated task execution requires to be innately energy-conscious. Studies in sharing techniques can be categorized into three methods, as depicted in figure 4.

i. Control oriented

Centralized and distributed control differentiate the sharing model in mobile edge devices.

a) Centralized Controller

It is dedicated in managing the workload on every edge device in a network. Directed Acyclic Graph (DAG)-based workflow is used by the devices at the edge. The cooperative execution of tasks is rested in a cloud controller [83, 84]. SDN is used as the basis for implementing Software-Defined Cooperative Offloading Model (SDCOM) [85]. A controller is used to enhance the coordination among mobile devices linked to the controller on a Packet Delivery Network (PDN) gateway. The controller focuses on decreasing the gateway's traffic and assuring impartiality in energy utilization among mobile devices. The actively arriving tasks are managed by Online Task Scheduling (OTS) algorithm.

b) Distributed Controller

A game-theoretic method was utilized as a decentralized method for attaining the Nash equilibrium in coordinated devices [86]. The Nash equilibrium in the sharing model is concerned deeper to create the Congestion Game (COG)-based sharing and Multi-item Auction (MIA) model [87].

ii. Adaptive Techniques' oriented

It is nature-stimulated and provide a solution for multi-objective optimization problems [88]. Sharing model is used concerning various goals in a system. For instance, the battlefield scenario implements the sharing model at the edge [89]. In this environment, latencies are expected to be in a decreased value, and the energy consumption of the devices is also expected to be in optimal value. The adaptive methods discussed in existing literature is summarized below:

a) Connectivity-conscious

The sharing model requires the details about the connectivity among devices, for instance, a battlefield scenario. A mobile device increases its processing when peer devices come inside its proximity of communication [89]. A probabilistic forecasting model in an inherently scheduled task can complete the execution process promptly on a peer device when it is in the scope of the device. Connectivity-conscious techniques can be of single or multi-hop, or opportunistic [90].

(1) Single Hop Techniques

In this method of single hopping, a fully interconnected network is formed by a device after getting a list of its neighbours. If a device shares a workload, then the workload is automatically shared with other devices that are in direct connection with the respective device [90,].

(2) Multi-hop Techniques

Every device calculates the shortest path to each other node in the network which is capable of workload sharing. The workload is generally distributed with devices that may decrease the entire energy footprint. When compared to single-hop techniques, the advantage of using the multi-hop technique in the sharing mode is, a huge supply of resources can be drawn into workloads that need thorough computations. The total running time of a distributed workload is decreased using a task-sharing approach implementing a greedy algorithm [91].

(3) Opportunistic Techniques

A device in need to distribute its workload in these methods confirms whether its peers can run a task when it is inside the proximity of communication. It is forecasted through contextual profiling of how long a device was inside the communication proximity of its peers. In the IoT environment for data-centric services, a group of mobile devices creates a mobile cloud through opportunistic networking to process the multiple IoT sensors requests [92].

b) Heterogeneity-conscious

Generally, edge devices are diversely natured at all levels in a mobile cloud. Hence, the operating system, processor architecture, and workload deployment confront various issues in easing coordination [93]. Solutions for handling diverseness concerned problems in mobile networks are developed. For instance, Honeybee, a work-sharing method was developed in which rotations of diversified mobile devices are utilized to share the workload among the edge devices [94].

c) Security-conscious

Honeypot which identifies and separates malicious attacks present in a device employed in the sharing model is proposed [95,151]. A malevolent device is found in twenty minutes. If a device is found to be malevolent, then it is separated for safeguarding the network.

d) Fairness-conscious

It is a multi-objective optimization issue. The goals are to decrease the battery drain in mobile devices to extend the lifetime of the network as well as enhance the performance gain of the workload shared among devices [96].

iii. Cooperative sharing oriented

Devices at edge can distribute workloads either (a) in a less framed environment based on ad hoc coordination or (b) in a more tightly coupled surrounding to ease coordination. Facilitating device-to-device communication by using ad hoc networks is an existing research area. This coordination is encountered for Mobile cloud computing (MCC) in the sharing model environment for the edge [97]. The term "transient clouds," is named by a research work where neighbouring mobile devices form an ad hoc cloud, and the fundamental algorithm for managing the tasks is relied on an alternative of the Hungarian method [98].

a) Infrastructure- based Cooperative sharing

Federation of devices is being analysed at the edge of the network to ease the coordination [99]. This produces tightly coupled coalitions when compared to ad hoc clouds and cost-effective than exclusive micro cloud deployment.

Offloading

Here, a server, an application, and its associated data are transferred to the Edge of the network. It either increases the processing needs of a single or a group of user devices or conducts cloud services to process device requests nearer to the source. Analysis based on Offloading can be distinguished in the following two ways.

i. Offloading from User Device to Edge

It increases processing in user devices by utilizing edge nodes (usually a single hop away). Two major methods used for offloading to edge from user device are application partitioning and caching mechanisms.

a) Partitioning of application

GigaSight architecture is an instance for offloading from devices to the Edge through application partitioning [76] in which Cloudlet Virtual Machines (VMs) helps in processing videos streamed from numerous mobile devices [100]. The Cloudlet VM is utilized for denaturing, a process of removing user related content for protecting privacy. The framework involved is presented as a Content Delivering Network (CDN) inversely. In this section methods and models used for application partitioning are discussed.

1) Methods

Brute Force, Greedy Heuristic, Simulated Annealing and Fuzzy Logic are the four methods discussed in the context of application partitioning.

Brute force approach: The entire feasible combinations of offloading plans (taking the cloud, edge nodes, and user devices) are analyzed [101]. The scheme with the minimal running time for a task is then selected. It is simply not an exact practical solution given the time required to obtain a scheme, but instead could provide awareness into the search space.

Greedy Heuristic approach: It concentrates on reducing the time taken for finishing the execution of a task on the mobile device [101]. Offloading framework utilizes a greedy heuristic approach for obtaining offloading plans during the implementation of prototype models [102].

Simulated Annealing approach: The search space is dependent on the usage of fog and cloud nodes, total costs, and the finishing time of an application to derive an offloading plan that reduces the costs and the task accomplishment time [103].

Fuzzy logic: The objective is to enhance the QoE given by several parameters like resource requirements, sensitivity and service access rate for delay in data processing. It is utilized in ranking every application placement request by taking the nodes' processing abilities [104].

2) Models

Graph-oriented, component-oriented and neural network-oriented are the three fundamental models used for partition the application from devices to the edge.

Graph-oriented: It is used in the automated partitioning of application in Clone Cloud [105]. In the execution time, it translates to moving the application thread onto the clone, then it is returned onto the first mobile device.

Component-oriented: The various capabilities of an application (a web browser) executing on a device are formed as components separated from the device and edge server [106].

Neural Network-oriented: A deep neural network's latency is decreased for latency-critical applications without transmitting images/videos distantly from the source. Many layers are present in deep networks that are shared over various nodes. The infrastructure of Neurosurgeon models the separation among layers that is both energy and latency effective from end-to-end [107].

b) Caching Mechanisms

A global cache is formed in a way that it is accessible on an edge node that plays as a distributed memory for numerous devices that require to interact and the caching mechanisms are:

1) Chunking and Aggregation based

The multi Radio Access Technology (multi-RAT) was an architectural model for upload caching. Here, in this mechanism, VMs are present at the edge of the network, and a user device uploads chunks of a large file onto them concurrently [108]. After aggregation VM integrates these chunks, and transmitted to a cloud server.

2) *Reverse Auction Game-based*

A caching mechanism on coordinating the edge nodes was stated as an alternate mechanism in [109]. The user-generated videos are distributed among the users through caching the edge. A reversible auction game is utilized to stimulate caching.

ii. Offloading to the Edge from the cloud

Here, the data flow direction is reversed when compared to the above mechanism by moving the workload to the edge from the cloud. Server offloading, caching mechanisms and web programming are the different mechanisms.

a) Server Offloading

Replication or partitioning is used to offload the server executing in the cloud to edge. Replication is a method to create server replications on edge.

1) Database Cloning

An application's database tends to be made replica and shared by multiple applications or users at the edge of the network [111].

2) Application-specific Data Replication

This approach enables a particular application to fetch the relevant data to the users in the edge for executing the application in a consistent way which is contrasting to database cloning [110]. However, these methods are not viable in resource-constrained edge scenarios because of the assumption that edge nodes are not storage restricted by the cloning and application-concerned data replication methods.

3) Partitioning

The partitioning factors such as functionality aware, latency-aware and geography-aware in the server are concerned in offloading to the edge from the cloud. *Functionality-aware:* Google Glass comes under the cognitive assistance applications category in which the cloud alone does not provide the processing required for such type of latency-critical applications. To fulfil the latency and processing demands of applications based on cognitive assistance, the research is ongoing to offload the required processing into the Cloudlet VMs [112].

Latency-aware:[114] explained the backend partitioning with an application logic traditionally present in clouds for servicing the application requests concurrently. The hybrid edge-assisted execution model for LTE networks works by fulfilling the application requests by the cloud and also the edge networks oriented on latency needs. It varies with the Edge Nohyde Resource Management (ENORM) framework by way of partitioning the server along with geographical requirements [112]. *Geography-aware:* Online gaming application such as Pokemon request the services which are normally transferred to a cloud server from user devices. The ENORM framework segregates the game server and utilizes in an edge node instead of transmitting the entire traffic to data centers [113]. Geographically related data to a particular location is then made obtainable in an edge node. The users form the cloud to Edge from the relevant offloading. Functionality aware, latency-aware and geography-aware are the factors involved with partitioning.

b) Caching Mechanisms

It is found that two methods, such as multi-layer caching and content popularity and forms the methodologies in caching.

1) Based on Content Popularity

ISP-based caching and CDN mitigate the network congestion problems at the time of downloading apps in user devices. However, considerable issues emerge due to increased growth of apps and devices.

2) Caching in Multi-layers

It is a wireless networking method used for delivering the contents [115]. It is assumed that a global cache caches the data from data centers is available at a base station, and localized caches are accessible on edge nodes. Uncoded caching and coded caching are the two methods deployed in this caching. Every node is unknown about the cached content of other nodes, and hence data cooperation is not needed in uncoded caching. The coded caching method deals with coding the cached content provided that all other edge nodes are needed to encode the content before sending to users.

3.2. Control

Controlling the resources is another method for categorizing architectures in accomplishing the management of various resources in an edge scenario. This research survey identified two architectures, such as centralized and distributed control architectures. Centralized control is the usage of a controller which decides the processing, communication, networks of the edge resources. In contrast to this, the architecture can be called as distributed if the process of making decisions is distributed across the edge nodes. The control techniques used in edge computing are summarized in Figure 5 and discussed in this section.

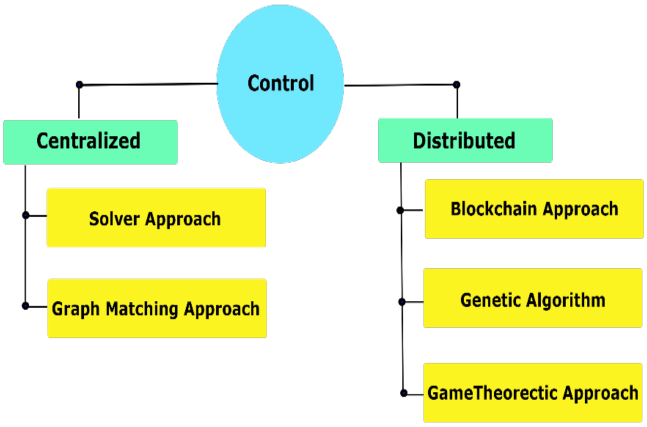

Figure 5. Resource control of edge computing model

Centralized

The two centralized architectures commonly used in edge computing are: (i) solver-approach and (ii) graph matching–approach.

i. Solver-approach

Solver approach is based on mathematical solvers which are generally employed in initiating deploying and redeploying plans for distribution of workloads in clouds, clusters, and grids. Approaches similar to the above is fixed for edge scenario [116]. For instance, a Least Processing Cost First (LPCF) method was developed for controlling the assignment of tasks in edge nodes. Solver approach is focussed at processing costs and reducing network.

ii. Graph Matching-approach

An offloading infrastructure considering device-to-device and cloud offloading techniques were stated. Offloading of tasks through a 3-layer graph-matching algorithm formed by getting the offloading space (mobiles, edge nodes, and the cloud) into consideration. The issue in reducing the running time of the whole task is plotted to the minimal weight-matching problem in the 3-layer graph. A consolidated method utilizing the Blossom algorithm was used to produce an offloading plan [117].

Distributed

The three different types of distributed architectures are (i) game-theoretic approach (ii) blockchain approach (iii) genetic algorithm

i. Game-theoretic approach

It is used to attain control in a decentralized manner for tasks offloading in the mobile-edge cloud computing having a diverse-channel in wireless environment [119]. It is obtaining an ideal solution using centralized methods that results in NP-hard. Hence the game-theoretic method is found to be more suitable in those type of environments. The Nash equilibrium was accomplished for offloading in a decentralized scenario. In contrast, two metrics, such as the number of benefited users in the cloud and the overheads in computation, were analyzed to substantiate the viability of the game-theoretic method over other centralized methods.

ii. Blockchain approach

It serves as a supporting method to implement decentralized management in edge computing systems [118] and enables the feasibility in developing a decentralized peer-to-peer network in the absence of intermediaries. Therefore, imparting the blockchain to edge computing model leads to communication among the nodes present at the edge without the need of intermediates. A decentralized ledger platform termed as Hyper ledger Fabric is utilized for execution and promotion of intelligent contracts defined by the user securely.

iii. Genetic algorithm

Generally, the sensors are the end devices which transmits the data to a computing node through a network in IoT systems and the several features regarding communication, computation and networking ae decided by it. This decision making is shared along with several edge nodes by using the Edge Mesh approach, which utilizes a processing overlay network also with a genetic algorithm [120]. This combination is used for mapping the task graph to the communication network for minimizing the utilization of energy. The variables concerned in this genetic algorithm are the Generation Gap for mutation rate, crossover methods, and size of the population [121].

3.3. Tenancy

Tenancy is one of the techniques used in the management of resources by providing a classification of architectures in an edge-oriented environment. It can be defined as the possibility of sharing the fundamental resources of hardware among various entities to achieve optimized usage of resources and effectiveness in energy consumption. The complete utilization of hardware resource by a single entity can be called a single-tenant system. Contrarily, the utilization of the same resource by various entities constitutes the multi-tenant system. Multi-tenancy is the essential attribute for an absolute distributed system to be accessed publicly. A single-tenant scheme applies to an entity's access to a private hardware. Conversely, a multi-tenant scheme refers to the sharing of the same resource by many individuals. Multi-tenant requires to be an optimal distributed system that is publicly available to allow the efficient use of resources in edge computing with minimal costs.The different possibilities in the classification are as follows:

i. Single Application, Single User (SASU)

This method makes the edge capable of executing only one application, and only the user can alone get connected to the application. The user and application utilize the hardware resources.

ii. Single Application, Multiple User (SAMU)

It provides the edge with the capability to execute one application that supports multiple users. Although the fundamental hardware resources are not shared among applications, there is a higher degree of sharing than SASU, since the edge node services multiple user requests.

iii. Multiple Application, Single User (MASU)

It makes the edge node to host multiple applications, but each application can only support one user. This mode of the tenancy is used for experimental purposes (or stress-testing the system) during the development of an ideal infrastructure.

iv. Multiple Application, Multiple User (MAMU)

It makes the edge node to host multiple applications, and many users are connected to an individual application. This is a perfect infrastructure in representing the publicly accessible infrastructure.

Multi-tenancy is supported by two methods termed as system virtualization and network slicing.

1) *System Virtualization*

It is a method supporting multi-tenancy at the system level. Different virtualization techniques are presently obtainable, like containers (considered in Section 3.2.1) and traditional VMs. VMs possess a wider resource footprint in comparison to containers. Hence edge computing uses lightweight virtualization presently [112, 122, 124] that eases the isolation of resources for separate applications made feasible by virtualization, where the users may access the virtualized application—for instance, simultaneous hosting of numerous applications with various containers on an edge node.

2) *Network Slicing*

Several logical networks are made to execute on the physical network's top layer while considering the network level. Therefore, the same physical network is used for communication by different networks with various throughput and latency requirements [124].

4. Infrastructure

The fog/edge computing framework gives functionalities with software and hardware to control the computation, storage resources and network [53] for utilities using the fog/edge. The three classes of infrastructure involved in effective management of resources in fog/edge computing is given below:

- **Hardware:** Research in edge computing proposes utilizing small-form-factor devices like Wi-Fi set-top boxes, Access Points (APs), network gateways, small home servers, drones, cars, and even edge ISP servers, as computing servers forming effectiveness in resources [127]. Single-board computers (SBCs) are fixed in these devices, offering substantial computing functionalities. Products like smartphones, laptops, desktops are also used by edge computing.
- **System software:** It executes edge hardware resources directly on the CPU, memory, and network devices. It controls and share the resources to the edge applications. Operating systems and virtualization software are some kinds of system software.
- **Middleware:** It works by executing on an operating system (OS) and providing compatible functionalities not underpinned by the system software. The middleware correlates decentralized process nodes. Containers or virtual machines are deployed for each edge node.

4.1. Hardware

Hardware forms the core unit of any framework for handling the functioning and controlling of computation on a real-time basis. The grouping of such small computing servers enhances the environment in cloud computing and it can be accelerated by a well-defined set of applications implementing cyber-physical systems (CPS) and IoT data. The hardware in fog/edge computing is categorized into two methods, as depicted in Table 3.

4.2. Software

In fog/edge platform, system software functions directly with fog/edge devices and controls the processing, storage and networking of the devices. Some examples are VMs and containers. In fog/edge computing, the usage of system software is categorized as virtualization concerning system and network as depicted in Table 4.

4.3. Middleware

Along with delivering interrelated services to system software, middleware involves in monitoring the performance, coordinating & organizing, providing communication utilities, and protocols to fog/edge computing. There are four classes about the usage of middleware in fog/edge computing is divided into the following, as presented in Table 5.

Table 3. Hardware for IoT Infrastructure

Hardware	Devices	Technique used	Application areas	Reference
Computation Devices	Single-board Computers - Raspberry Pi	Efficient data aggregation	Connected vehicles, Drones, cloud system	[129,146]
	Commodity Products-desktops, laptops, and smartphones	Rendering service using under-utilized resources	classrooms, movie theatres, and cafes	[130]
Network Devices	Gateways and Routers	Network bandwidth utilization	Establish a data path	[131]
	Wi-Fi Access point	Wireless communication	Gateways to WSN	[121]
	Racks in edge	Network Innovations in a Global Environment	Smart cities	[132]

Table 4. System Software for IoT Infrastructure

System Software	Virtual machine concept used	Technique used	Application areas	Reference
System Virtualization	Virtual machine hypervisor, cloudlet	Partition the resources among multi-tenants	Remote surveillance	[133,134]
	Containers, Namespaces	Platform-as-a-Service (PaaS)	Wi-Fi access point, gateway	[135,136]
	VM Container Migration, Data centres	Proactive techniques	Load balancing and fault tolerance	[137,138]
Network Virtualization	Network Function Virtualization (NFV), SDN, Hybrid Fog and cloud (HFC)	Routing, firewall	Managing network through software	[77,139,140,150]
	Overlay Network, Peer to peer	Virtual link	Decentralized Edge computing	[141]

Table 5. Middleware for IoT Infrastructure

Middleware	Edge Types	Technique used	Application areas	Reference
Volunteer Edge Computing	Data Store Nebula	Location-aware data processing	Network performance monitoring	[142,143]
Hierarchical Fog/Edge Computing	Mobile Fog facilitates Distributed processes	Load-balance workloads	Scaling capabilities during runtime	[144,145]
Mobile Fog/Edge Computing	Execution Prediction, FemtoCloud	Offload computation-intensive task	Achieving satisfactory latency and bandwidth	[83,104]
Cloud Orchestration Management	Inter-device coordination Micro Cloud	Exploiting resources	Image processing	[146]

5. Algorithms

The four fundamental algorithms enabling fog/edge computing are: (i) discovery - finding edge resources inside the network to be utilized for distributed processing, (ii) benchmarking - acquiring resources' performance to make decisions in expanding the performance concerning implementation, (iii) load-balancing - distributing workloads among various resources with some bases like fairness, priorities and (iv) placement - finding resources necessary for the deployment of workload. Table 6 presents the techniques used and applications based on the four algorithms. Edge computing can minimise the network and bandwidth issues by localising cluster-based networks and associated with moving large amounts of data to or from IoT devices and reduce reliance on the network. This reduced reliance of networks enables the intractability among the edge devices.

Table 6. Types of Algorithms

Algorithms	Techniques used	Services	Application areas	Reference
Discovery	Programming frameworks, protocols for handshaking and message passing	Foglets, Edge-as-a-Service (EaaS), Internet.	Public resource distribution. Online game. simulation-based validation	[82,147,148]
Benchmarking	Basic conventions for assessing functional properties, benchmarking based on application and integrated benchmarking	Spark, Cloud Sim	IoT application	[149,153]
Load-balancing	Particle Swarm Optimization - Constrained Optimization	Internet of vehicles, Cooperative load-balancing model (CooLoad), Breadth-First Search (BFS)	Effective load balancing authenticated data center.	[152,154,155]
Placement	Dynamic condition-aware iterative techniques	Priced timed Petrinets (PTPNs) Fog-to-Cloud (F2C)	Dynamic behaviour on performance, time and cost. Real-time applications	[156,157]

6. Use Case Scenarios

The goal of this Section is to explain the most important functional requirements and features of edge computing platforms from a use case. There are many use cases where the edge computing is implemented such as Smart Agriculture, Smart Healthcare, Smart Transportation and Smart Waste Managements. Edge computing is the subset of the cloud computing to enrich the cloud services for all kinds of applications in IoT environment various techniques is reviewed under the hardware, algorithm and infrastructure for efficient resource management. A typical use case scenario of smart agriculture is explained in this section.

6.1. Smart agriculture

The advent of Information and Communication Technologies (ICT) has a significant part in the agricultural sector by delivering computer-based services and functionalities [123]. These systems are unable to satisfy the needs of the current generation because of the factors like processing of massive data, lack of major functionalities such as storage space, processing speed, dependability, availability and extensibility. It also lacks an effective utilization of resources in computer-based agricultural systems. Big data features are presented in the form of Agriculture-as-a-Service (AaaS) applications. Cloud-based service can easily control various kinds of agriculture based-data and serves as the solution for ever-growing volume of agricultural data. As per the government report, the amount of agriculture data sets acquired by Open Government Data Platform India records to a size of 3.5GB [126]. The data becomes diverse, specifically in the form of images. Figure 6 depicts the edge computing based smart agriculture.

The status of agricultural crops affected due to insects, weather, and other factors is acquired through the IoT sensors and satellites in the form of heterogeneous, huge volume of big data. Data sets expand with a velocity of

80.72 KB/ minute or more [126] due to the continuous data collection. Cloud-based service is a key solution for solving this problem which works on various aspects (soil, crop, pest, weather, fertilizer, cattle, productivity, irrigation, and equipment) following the steps given below.

- gathering of data from multiple sensors through predefined devices,
- categorizing the collected data into several classes after analysis,
- storing the categorized information in cloud repository for upcoming use and
- automatic diagnosing of the status in agriculture.

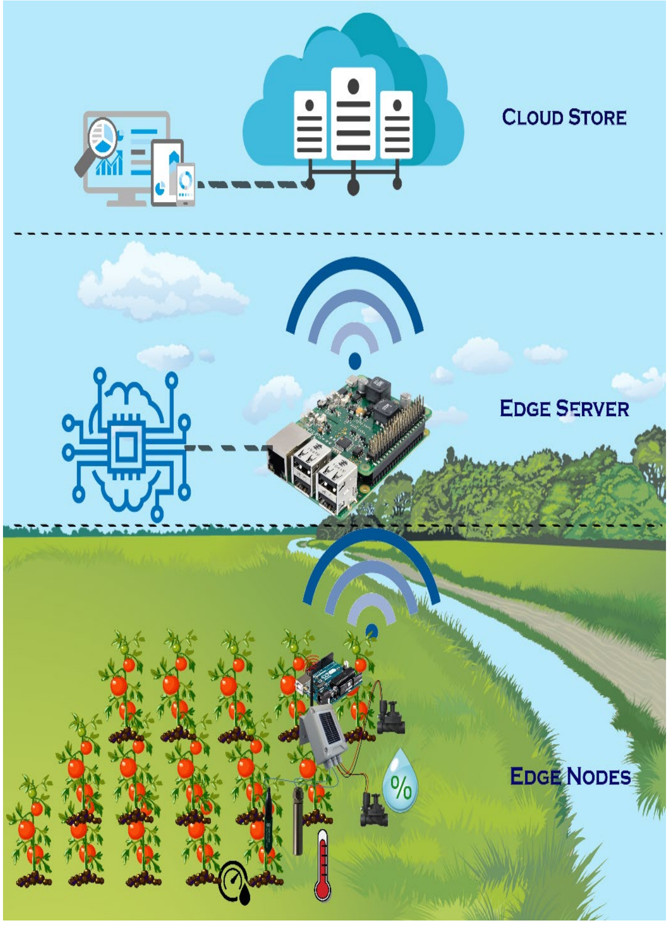

Figure 6. Edge Computing scenario for intelligent agriculture

The number of users in agriculture systems and the volume of data sets is directly proportional to each other. Therefore, Cloud computing serves as a solution for providing highly extendable and resilient distributed computing environments. The cloud-based information system can find the QoS needs from user requests, and resources should be allocated effectively to run the user request. There arises a need to track and control the ground level factors, which is then automated and predicted by communicating to the cloud. These agricultural factors are handled in the field level itself, where IoT plays a major role. where Edge computing arrives into the picture. Edge computing decreases the computation burden on the cloud for better results in smart farming.

In accordance with architecture in resource management, [125] the edge-server layered method named DART (Data analysis in real time) scheme is proposed in tomato harvest yield prediction. The layered method involves the separation of edge layer and server layer. An edge trained and tested devices are used to analyse the collecting and pre-processing of growth measurement values and environment sensor values by the IoT gateway. The server analyses after collecting cumulative data from the edges for predicting the complete tomato yield which includes the process of server training and server testing. At this point, infrastructure in resource management comes into the picture where the system deploys sensor devices to communicate with MCU using UART interface. REST api is used for sending the data from MCU to the edge device (IoT gateway by the Raspberry Pi 3) systematically. A mini MySQL database is employed in the edge device for storing the data to implement the DART methodology. In the meanwhile, the edge device executes the Smart Farm Edge application that runs on the apex of the DART framework. Appropriate usage of algorithm handles the computation analytics at the edge of this system where the nonlinear polynomial model aids the server in predicting the subsequent harvest yield and gross fruit yield. The system was analysed on the edges with different machine learning models, such as Long Short-Term Memory Recurrent Neural Networks (LSTM), multi-layer perceptron and ridge regression as a comparison with the proposed algorithm.

The various perspectives in resource management of edge computing are effectively addressed in this article for tomato yield prediction.

7. Conclusion

An extensive range of technical challenges regarding management of resources in fog/edge computing is approached in this survey work. Although some demanding issues prevail with respect to the capacities and performance factors in fog/edge computing, few future research directions are discussed to confront the remaining issues. Edge computing has drawn heed over the last decade as an alternate methodology to traditional centralized cloud computing model. Latency in communication is reduced and network bandwidth is used effectively by bringing the processing devices within the proximity of mobile and IoT devices. The various techniques in resource management are presented in this paper to find and differentiate the major benefactions in terms of architectures, frameworks and algorithms. Evolution from prototype models to usable platforms in daily life is still in its infancy. Edge computing systems are expected to be the most important links for effective processing of information. Methodical studies in this domain are expanding over these years and several resource optimization and management schemes are proposed. A wide survey of edge-based computing systems presented in this paper is expected to provide an intuitive understanding for both academicians and researchers.

References

[1] Ju Ren, Yaoxue Zhang, Ruilong Deng, Ning Zhang, Deyu Zhang, and Xuemin Shen. 2016. Joint channel access and sampling rate control in energy harvesting cognitive radio sensor networks. IEEE Trans. Emerg. Top. Comput. 7, 1(2019), 149–161. DOI:10.1109/TETC.2016.2555806.

[2] Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2016-2021. 2017. Retrieved from https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white paperc11- 520862.html.

[3] Weisong Shi, Jie Cao, Quan Zhang, Youhuizi Li, and Lanyu Xu. 2016. Edge computing: Vision and challenges. IEEEInternet Things J. 3, 5 (2016), 637–646.

[4] Blesson Varghese, NanWang, Sakil Barbhuiya, Peter Kilpatrick, and Dimitrios S. Nikolopoulos. 2016. Challenges and opportunities in edge computing. In Proceedings of the IEEE International Conference on Smart Cloud (SmartCloud'16) IEEE, 20–26

[5] Mahadev Satyanarayanan. 2017. The emergence of edge computing. Computer 50, 1 (2017), 30–39.

[6] Mahadev Satyanarayanan, Paramvir Bahl, Raman Caceres, and Nigel Davies. 2009. The case for VM-based cloudlets in mobile computing. IEEE Pervas. Comput. 8, 4, (2009).

[7] Flavio Bonomi, Rodolfo Milito, Jiang Zhu, and Sateesh Addepalli. 2012. Fog computing and its role in the Internet of things. In Proceedings of the 1st Edition of the MCC Workshop on Mobile Cloud Computing. ACM, 13–16.

[8] Amir Vahid Dastjerdi and Rajkumar Buyya. 2016. Fog computing: Helping the Internet of things realize its potential. Computer 49, 8 (2016), 112–116.

[9] Ashkan Yousefpour, Caleb Fung, Tam Nguyen, Krishna Kadiyala, Fatemeh Jalali, Amirreza Niakanlahiji, Jian Kong, and Jason P. Jue. 2019. All one needs to know about fog computing and related edge computing paradigms: A complete survey. J. Syst. Archit. (2019).

[10] Yaoxue Zhang. 2004. Transparence computing: Concept, architecture and example. Chin. J. Electr. 32, S1, Article 169 (2004), 5 pages.

[11] Ju Ren, Hui Guo, Chugui Xu, and Yaoxue Zhang. 2017. Serving at the Edge: A scalable IoT architecture based on transparent computing. IEEE Netw. 31, 5 (2017), 96–105.

[12] Flavio Bonomi, Rodolfo Milito, Jiang Zhu, and Sateesh Addepalli. 2012. Fog computing and its role in the Internet of Things. In Proceedings of the Workshop on Mobile Cloud Computing (MCC'12). ACM, New York, NY, 13–16.

[13] Sami Yangui Roch H. Glitho Monique J. Morrow Carla Mouradian, Diala Naboulsi and Paul A. Polakos. 2018. A comprehensive survey on fog computing: State-of-the-art and research challenges. IEEE Commun. Surv. Tutor. 20, 1 (2018), 416–464.

[14] Flavio Bonomi, Rodolfo Milito, Preethi Natarajan, and Jiang Zhu. 2014. Fog Computing: A Platform for Internet of Things and Analytics. Springer International Publishing, Cham, 169–186. DOI:https://doi.org/10.1007/ 978-3-319-05029-4_7

[15] Luis M. Vaquero and Luis Rodero-Merino. 2014. Finding your way in the Fog: Towards a comprehensive definition of fog computing. ACM SIGCOMM Comput. Commun. Rev. 44, 5 (2014), 27–32.

[16] Shanhe Yi, Cheng Li, and Qun Li. 2015. A survey of fog computing: Concepts, applications and issues. In Proceedings of the 2015 ACM Workshop on Mobile Big Data (Mobidata'15). ACM, 37–42.

[17] M. Yannuzzi, R. Milito, R. Serral-Gracia, D. Montero, and M. Nemirovsky. 2014. Key ingredients in an IoT recipe: Fog computing, cloud computing, and more fog computing. In Proceedings of the IEEE International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD'14). 325–329. DOI:https://doi. org/10.1109/CAMAD.2014.7033259

[18] AmirVahidDastjerdiandRajkumarBuyya.2016.Fogcomputing:HelpingtheInternetofThingsrealizeitspotential. Computer 49, 8 (2016), 112–116.

[19] Chiang Mung, Ha Sangtae, I Chih-lin, Risso Fulvio, and Zhang Tao. 2017. Clarifying fog computing and networking: 10 questions and answers. IEEE Commun. Mag. 55, 4 (Apr. 2017), 18–20. DOI:https://doi.org/10.1109/MCOM.2017. 7901470

[20] OpenFog Reference Architecture for Fog Computing. 2017. Retrieved from https://www.openfogconsortium.org/wp-content/uploads/OpenFog_Reference_Architecture_2_09_17-FINAL.pdf.

[21] Hasti Ahlehagh and Sujit Dey. 2014. Video-aware scheduling and caching in the radio access network. IEEE/ACM Trans. Netw. 22, 5 (2014), 1444–1462.

[22] Arif Ahmed and Ejaz Ahmed. 2016. A survey on mobile edge computing. In Proceedings of the 2016 10th International Conference on Intelligent Systems and Control (ISCO'16) DOI: https://doi.org/10.1109/ISCO.2016.7727082.

[23] F. Qureshi and S. Krishnan, "Wearable Hardware Design for the Internet of Medical Things (IoMT)," Sensors, vol. 18, no. 11, p. 3812, Nov. 2018, doi: 10.3390/s18113812.

[24] M. Avgeris, D. Spatharakis, D. Dechouniotis, N. Kalatzis, I. Roussaki, and S. Papavassiliou, "Where There Is Fire There Is SMOKE: A Scalable Edge Computing Framework for Early Fire Detection," Sensors, vol. 19, no. 3, p. 639, Feb. 2019, doi: 10.3390/s19030639.

[25] X. Zhang, S. Tang, X. Liu, R. Malekian, and Z. Li, "A Novel Multi-Agent-Based Collaborative Virtual Manufacturing Environment Integrated with Edge Computing Technique," Energies, vol. 12, no. 14, p. 2815, Jul. 2019, doi: 10.3390/en12142815.

[26] M. Syafrudin, N. Fitriyani, G. Alfian, and J. Rhee, "An Affordable Fast Early Warning System for Edge Computing in Assembly Line," Appl. Sci., vol. 9, no. 1, p. 84, Dec. 2018, doi: 10.3390/app9010084.

[27] F. Xhafa, B. Kilic, and P. Krause, "Evaluation of IoT stream processing at edge computing layer for semantic data enrichment," Future Gener. Comput. Syst., vol. 105, pp. 730–736, Apr. 2020, doi: 10.1016/j.future.2019.12.031.

[28] L. Greco, P. Ritrovato, and F. Xhafa, "An edge-stream computing infrastructure for real-time analysis of wearable sensors data," Future Gener. Comput. Syst., vol. 93, pp. 515–528, Apr. 2019, doi: 10.1016/j.future.2018.10.058.

[29] M. R. Suma and P. Madhumathy, "Acquisition and Mining of Agricultural Data Using Ubiquitous Sensors with Internet of Things," in International Conference on Computer Networks and Communication Technologies, vol. 15, S. Smys, R. Bestak, J. I.-Z. Chen, and I. Kotuliak, Eds. Singapore: Springer Singapore, 2019, pp. 249–261.

[30] M. A. Zamora-Izquierdo, J. Santa, J. A. Martínez, V. Martínez, and A. F. Skarmeta, "Smart farming IoT platform based on edge and cloud computing," Biosyst. Eng., vol. 177, pp. 4–17, Jan. 2019, doi: 10.1016/j.biosystemseng.2018.10.014.

[31] K. Sahlmann and T. Schwotzer, "Ontology-based virtual IoT devices for edge computing," in Proceedings of the 8th International Conference on the Internet of Things, Santa Barbara California USA, Oct. 2018, pp. 1–7, doi: 10.1145/3277593.3277597.

[32] I. Sittón-Candanedo, R. S. Alonso, Ó. García, L. Muñoz, and S. Rodríguez-González, "Edge Computing, IoT and Social Computing in Smart Energy Scenarios," Sensors, vol. 19, no. 15, p. 3353, Jul. 2019, doi: 10.3390/s19153353.

[33] E. Cordelli, G. Pennazza, M. Sabatini, M. Santonico, and L. Vollero, "An Open-Source Smart Sensor Architecture for Edge Computing in IoT Applications," Proceedings, vol. 2, no. 13, p. 955, Nov. 2018, doi: 10.3390/proceedings2130955.

[34] Z. Zhao, P. Lin, L. Shen, M. Zhang, and G. Q. Huang, "IoT edge computing-enabled collaborative tracking system for manufacturing resources in industrial park," Adv. Eng. Inform., vol. 43, p. 101044, Jan. 2020, doi: 10.1016/j.aei.2020.101044.

[35] S. Alamgir Hossain, Md. Anisur Rahman, and M. A. Hossain, "Edge computing framework for enabling situation awareness in IoT based smart city," J. Parallel Distrib. Comput., vol. 122, pp. 226–237, Dec. 2018, doi: 10.1016/j.jpdc.2018.08.009.

[36] B. Chen, J. Wan, A. Celesti, D. Li, H. Abbas, and Q. Zhang, "Edge Computing in IoT-Based Manufacturing," IEEE Commun. Mag., vol. 56, no. 9, pp. 103–109, Sep. 2018, doi: 10.1109/MCOM.2018.1701231.

[37] X. Li, L. Zhu, X. Chu, and H. Fu, "Edge Computing-Enabled Wireless Sensor Networks for Multiple Data Collection Tasks in Smart Agriculture," J. Sens., vol. 2020, pp. 1–9, Feb. 2020, doi: 10.1155/2020/4398061.

[38] Mahadev Satyanarayanan, Paramvir Bahl, Ramón Cáceres, and Nigel Davies. 2009. The case for VM-based cloudlets in mobile computing. IEEE Perv. Comput. 8, 4 (2009), 14–23.

[39] Mahadev Satyanarayanan, Pieter Simoens, Yu Xiao, Padmanabhan Pillai, Zhuo Chen, Kiryong Ha, Wenlu Hu, and Brandon Amos. 2015. Edge analytics in the Internet of Things. IEEE Perv. Comput. 14, 2 (2015), 24–31.

[40] Mahadev Satyanarayanan. 2017. The emergence of edge computing. Computer 50, 1 (2017), 30–39.

[41] Kiryong Ha and Mahadev Satyanarayanan. 2015. OpenStack++ for Cloudlet Deployment. Technical Report. http:// elijah.cs.cmu.edu/DOCS/CMU-CS-15-113.pdf

[42] OECI. 2017. Retrieved from http://openedgecomputing.org/index.html.

[43] Blesson Varghese and Rajkumar Buyya. 2018. Next generation cloud computing: New trends and research directions. Fut. Gen. Comput. Syst. 79 (2018), 849–861.

[44] Cheol-Ho Hong, Kyungwoon Lee, Minkoo Kang, and Chuck Yoo. 2018. qCon: QoS-Aware network resource management for fog computing. Sensors 18, 10 (2018), 3444.

[45] Weisong Shi, Jie Cao, Quan Zhang, Youhuizi Li, and Lanyu Xu. 2016. Edge computing: Vision and challenges. IEEE Internet Things J. 3, 5 (2016), 637–646.

[46] Blesson Varghese, Nan Wang, Sakil Barbhuiya, Peter Kilpatrick, and Dimitrios S. Nikolopoulos. 2016. Challenges and opportunities in edge computing. In Proceedings of the IEEE International Conference on Smart Cloud (SmartCloud'16). IEEE, 20–26

[47] Mahadev Satyanarayanan. 2017. The emergence of edge computing. Computer 50, 1 (2017), 30–39.

[48] Mahadev Satyanarayanan, Paramvir Bahl, Ramón Caceres, and Nigel Davies. 2009. The case for VM-based cloudlets in mobile computing. IEEE Pervas. Comput. 8, 4 (2009).

[49] Flavio Bonomi, Rodolfo Milito, Jiang Zhu, and Sateesh Addepalli. 2012. Fog computing and its role in the Internet of things. In Proceedings of the 1st Edition of the MCC Workshop on Mobile Cloud Computing. ACM, 13–16.

[50] Amir Vahid Dastjerdi and Rajkumar Buyya. 2016. Fog computing: Helping the Internet of things realize its potential. Computer 49, 8 (2016), 112–116.

[51] Ashkan Yousefpour, Caleb Fung, Tam Nguyen, Krishna Kadiyala, Fatemeh Jalali, Amirreza Niakanlahiji, Jian Kong, and Jason P. Jue. 2019. All one needs to know about fog computing and related edge computing paradigms: A complete survey. J. Syst. Archit. (2019)

[52] Ramesh Rajagopalan and Pramod K. Varshney. 2006. Data-aggregation techniques in sensor networks: A survey. IEEE Commun. Surv. Tutor. 8, 4 (2006), 48–63.

[53] Marcos Dias de Assunção, Alexandre da Silva Veith, and Rajkumar Buyya. 2018. Distributed data stream processing and edge computing: A survey on resource elasticity and future directions. J. Netw. Comput. Appl. 103 (2018), 1–17.

[54] Elena Fasolo, Michele Rossi, Jorg Widmer, and Michele Zorzi. 2007. In-network aggregation techniques for wireless sensor networks: A survey. IEEE Wirel. Commun. 14, 2 (2007), 70–87.

[55] Jing He, Shouling Ji, Yi Pan, and Yingshu Li. 2014. Constructing load-balanced data aggregation trees in probabilistic wireless sensor networks. IEEE Trans. Parallel Distrib. Syst. 25, 7 (2014), 1681–1690

[56] Jine Tang, ZhangBing Zhou, Jianwei Niu, and Qun Wang. 2013. EGF-tree: An energy efficient index tree for facilitating multi-region query aggregation in the Internet of things. In Proceedings of the IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing. IEEE, 370–377.

[57] MF2C. Available online: http://www.mf2c-project.eu/ (accessed on 1 October 2018).

[58] Accelerating Innovation and Collaboration for the Next Stage. Available online: http://www.ntt.co.jp/news2013/1311ehzt/pdf/xgxf131108d_all.pdf (accessed on 1 October 2018).

[59] Announcing the “Edge Computing” Concept and the “Edge Accelerated Web Platform” Prototype to Improve Response Time of Cloud Applications. Available online: http://www.ntt.co.jp/news2014/1401e/ 140123a.html (accessed on 1 October 2018).

[60] Ha, K.; Chen, Z.; Hu, W.; Richter, W.; Pillai, P.; Satyanarayanan, M. Towards wearable cognitive assistance. In Proceedings of the 12th Annual International Conference on Mobile Systems, Applications, and Services, Bretton Woods, NH, USA, 16–19 June 2014; pp. 68–81.

[61] Orsini, G.; Bade, D.; Lamersdorf, W. Computing at the mobile edge: Designing elastic android applications for computation offloading. In Proceedings of the 2015 8th IFIP Wireless and Mobile Networking Conference (WMNC), Munich, Germany, 5–7 October 2015; pp. 112–119.

[62] Cisco Kinetic. Available online: https://www.cisco.com/c/en/us/solutions/internet-of-things/iot-kinetic. html#~stickynav=1 (accessed on 1 October 2018).

[63] Vortex. Available online: http://www.prismtech.com/vortex (accessed on 1 October 2018).

[64] Cisco IOx. Available online: https://www.cisco.com/c/en/us/products/cloud-systems-management/iox/ index.html (accessed on 1 October 2018).

[65] Garcıa, M.; Fuentes, P.; Odriozola, M.; Vallejo, E.; Beivide, R. FOGSim Interconnection Network Simulator; University of Cantabria: Cantabria, Spain, 2014.

[66] Banerjee, S.; Liu, P.; Patro, A.; Willis, D. ParaDrop: An Edge Computing Platform in Home Gateways. In Fog for 5G and IoT; Wiley: Hoboken, NJ, USA, 2017; p. 13.

[67] El Amraoui, A.; Sethom, K. Cloudlet softwarization for pervasive healthcare. In Proceedings of the 2016 30th International Conference on Advanced Information Networking and Applications Workshops (WAINA), Crans-Montana, Switzerland, 23–25 March 2016; pp. 628–632.

[68] Monfared, S.; Bannazadeh, H.; Leon-Garcia, A. Software defined wireless access for a two-tier cloud system.In Proceedings of the 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM),Ottawa, ON, Canada, 11–15 May 2015; pp. 566–571.

[69] Wu, D.; Arkhipov, D.I.; Asmare, E.; Qin, Z.; McCann, J.A. UbiFlow: Mobility management in urban-scale software defined IoT. In Proceedings of the 2015 IEEE Conference on Computer Communications (INFOCOM), Hong Kong, China, 26 April–1 May 2015; pp. 208–216.

[70] Nam Ky Giang, Michael Blackstock, Rodger Lea, and Victor C. M. Leung. 2015. Developing IoT applications in the Fog: A distributed dataflow approach. In Proceedings of the 5th International Conference on the Internet of Things (IOT’15). IEEE, 155–162.

[71] Yiming Xu, V. Mahendran, and Sridhar Radhakrishnan. 2016. Towards SDN-based fog computing: MQTT broker virtualization for effective and reliable delivery. In Proceedings of the 8th International Conference on Communication Systems and Networks (COMSNETS’16). IEEE, 1–6.

[72] Hongbo Jiang, Shudong Jin, and Chonggang Wang. 2011. Prediction or not? An energy-efficient framework for clustering-based data collection in wireless sensor networks. IEEE Trans. Parallel Distrib. Syst. 22, 6 (2011), 1064– 1071

[73] Muhammad Habib ur Rehman, Prem Prakash Jayaraman, Saif ur Rehman Malik, Atta ur Rehman Khan, and Mohamed Medhat Gaber. 2017. RedEdge: A novel architecture for big data processing in mobile edge computing environments. J. Sens. Act. Netw. 6, 3 (2017).

[74] Suman Nath, Phillip B. Gibbons, Srinivasan Seshan, and Zachary R. Anderson. 2004. Synopsis diffusion for robust aggregation in sensor networks. In Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems. 250–262.

[75] Dnyaneshwar S. Mantri, Neeli Rashmi Prasad, and Ramjee Prasad. 2015. Bandwidth efficient cluster-based data aggregation for wireless sensor network. Comput. Electr. Eng. 41, C (Jan. 2015), 256–264.

[76] Luca Becchetti, Alberto Marchetti-Spaccamela, Andrea Vitaletti, Peter Korteweg, Martin Skutella, and Leen Stougie. 2009. Latency-constrained aggregation in sensor networks. ACM Trans. Algor. 6, 1, 13.

[77] Hongxing Li, Chuan Wu, Qiang-Sheng Hua, and Francis C. M. Lau. 2014. Latency-minimizing data aggregation in wireless sensor networks under physical interference model. Ad Hoc Networks 12 (Jan. 2014), 52–68.

[78] Stephan Reiff-Marganiec, Marcel Tilly, and Helge Janicke. 2014. Low-latency service data aggregation using policy obligations. In Proceedings of the IEEE International Conference on Web Services. IEEE, 526–533.

[79] Christos Anagnostopoulos. 2014. Time-optimized contextual information forwarding in mobile sensor networks. J. Parallel and Distrib. Comput. 74, 5 (2014), 2317–2332.

[80] Huaqun Wang, Zhiwei Wang, and Josep Domingo-Ferrer. 2018. Anonymous and secure aggregation scheme in fogbased public cloud computing. Fut. Gen. Comput. Syst. 78 (2018), 712–719.

[81] Guilhem Castagnos and Fabien Laguillaumie. 2015. Linearly homomorphic encryption from DDH. In Proceedings of the Cryptographers' Track at the RSA Conference. Springer, 487–505.

[82] Blesson Varghese, Nan Wang, Dimitrios S. Nikolopoulos, and Rajkumar Buyya. 2017. Feasibility of fog computing. Retrieved from: https://arxiv.org/abs/1701.05451.

[83] Karim Habak, Mostafa Ammar, Khaled A. Harras, and Ellen Zegura. 2015. Femto clouds: Leveraging mobile devices to provide cloud service at the Edge. In Proceedings of the IEEE 8th International Conference on Cloud Computing. 9–16.

[84] Karim Habak, Ellen W. Zegura, Mostafa Ammar, and Khaled A. Harras. 2017. Workload management for dynamic mobile device clusters in edge femtoclouds. In Proceedings of the 2nd ACM/IEEE Symposium on Edge Computing. ACM, 6.

[85] Yong Cui, Jian Song, Kui Ren, Minming Li, Zongpeng Li, Qingmei Ren, and Yangjun Zhang. 2017. Software defined cooperative offloading for mobile cloudlets. IEEE/ACM Trans. Netw. 25, 3 (2017), 1746–1760.

[86] Xu Chen. 2015. Decentralized computation offloading game for mobile cloud computing. IEEE Trans. Parallel Distrib. Syst. 26, 4 (2015), 974–983.

[87] Xiao Ma, Chuang Lin, Xudong Xiang, and Congjie Chen. 2015. Game-theoretic analysis of computation offloading for cloudlet-based mobile cloud computing. In Proceedings of the 18th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems. ACM, 271–278.

[88] Dragi Kimovski, Humaira Ijaz, Nishant Saurabh, and Radu Prodan. 2018. Adaptive nature-inspired fog architecture. In Proceedings of the IEEE 2nd International Conference on Fog and Edge Computing (ICFEC'18). IEEE, 1–8.

[89] Wei Gao. 2014. Opportunistic peer-to-peer mobile cloud computing at the tactical Edge. In Proceedings of the IEEE Military Communications Conference. IEEE, 1614–1620.

[90] Abderrahmen Mtibaa, Afnan Fahim, Khaled A. Harras, and Mostafa H. Ammar. 2013. Towards resource sharing in mobile device clouds: Power balancing across mobile devices. ACM SIGCOMM Comput. Commun. Rev. 43, 4 (Aug. 2013), 51–56.

[91] Colin Funai, Cristiano Tapparello, and Wendi Heinzelman. 2016. Mobile to mobile computational offloading in multi-hop cooperative networks. In Proceedings of the IEEE Global Communications Conference (GLOBECOM'16). IEEE, 1–7

[92] Eleonora Borgia, Raffaele Bruno, Marco Conti, Davide Mascitti, and Andrea Passarella. 2016. Mobile edge clouds for information-centric IoT services. In Proceedings of the IEEE Symposium on Computers and Communication (ISCC'16). IEEE, 422–428.

[93] Zohreh Sanaei, Saeid Abolfazli, Abdullah Gani, and Rajkumar Buyya. 2014. Heterogeneity in mobile cloud computing: Taxonomy and open challenges. IEEE Commun. Surv. Tutor. 16, 1 (2014), 369–392.

[94] Niroshinie Fernando, Seng W. Loke, and Wenny Rahayu. 2019. Computing with nearby mobile devices: A work sharing algorithm for mobile edge-clouds. IEEE Trans. Cloud Comput. 7, 2 (2019)

[95] Abderrahmen Mtibaa, Khaled Harras, and Hussein Alnuweiri. 2015. Friend or foe? detecting and isolating malicious nodes in mobile edge computing platforms. In Proceedings of the IEEE 7th International Conference on

Cloud Computing Technology and Science (CloudCom'15). IEEE, 42–49.

[96] Hariharasudhan Viswanathan, Eun Kyung Lee, and Dario Pompili. 2016. A multi-objective approach to real-time in-situ processing of mobile-application workflows. IEEE Trans. Parallel Distrib. Syst. 27, 11 (2016), 3116–3130.

[97] John Panneerselvam, James Hardy, Lu Liu, Bo Yuan, and Nick Antonopoulos. 2016. Mobilouds: An energy efficient MCC collaborative framework with extended mobile participation for next generation networks. IEEE Access 4 (2016), 9129–9144.

[98] Terry Penner, Alison Johnson, Brandon Van Slyke, Mina Guirguis, and Qijun Gu. 2014. Transient clouds: Assignment and collaborative execution of tasks on mobile devices. In Proceedings of the IEEE Global Communications Conference. IEEE, 2801–2806.

[99] Ivan Farris, Leonardo Militano, Michele Nitti, Luigi Atzori, and Antonio Iera. 2017. MIFaaS: A mobile-IoT-federationas-a-service model for dynamic cooperation of IoT cloud providers. Fut. Gen. Comput. Syst. 70 (2017), 126–137.

[100]Pieter Simoens, Yu Xiao, Padmanabhan Pillai, Zhuo Chen, Kiryong Ha, and Mahadev Satyanarayanan. 2013. Scalable crowd-sourcing of video from mobile devices. In Proceedings of the 11th International Conference on Mobile Systems, Applications, and Services. ACM, 139–152.

[101]Long Chen, Jigang Wu, Xin Long, and Zikai Zhang. 2017. ENGINE: Cost Effective Offloading in Mobile Edge Computing with Fog-Cloud Cooperation. Retrieved from: https://arxiv.org/abs/1711.01683.

[102]Antonio Brogi and Stefano Forti. 2017. QoS-aware deployment of IoT applications through the Fog. IEEE Int. Things J. 4, 5 (2017), 1185–1192.

[103]Redowan Mahmud, Satish Narayana Srirama, Kotagiri Ramamohanarao, and Rajkumar Buyya. 2019. Quality of experience (QoE)-aware placement of applications in fog computing environments. J. Parallel Distrib. Comput. 132 (Oct. 2019).https://www.sciencedirect.com/science/article/abs/pii/S0743731518301771.

[104]Byung-Gon Chun, Sunghwan Ihm, Petros Maniatis, Mayur Naik, and Ashwin Patti. 2011. Clonecloud: Elastic execution between mobile device and cloud. In Proceedings of the 6th Conference on Computer Systems. ACM, 301–314.

[105]Noriyuki Takahashi, Hiroyuki Tanaka, and Ryutaro Kawamura. 2015. Analysis of process assignment in multi-tier mobile cloud computing and application to Edge accelerated web browsing. In Proceedings of the 3rd IEEE International Conference on Mobile Cloud Computing, Services, and Engineering. IEEE, 233–234.

[106]Yiping Kang, Johann Hauswald, Cao Gao, Austin Rovinski, Trevor Mudge, Jason Mars, and Lingjia Tang. 2017. Neurosurgeon: Collaborative intelligence between the cloud and mobile Edge. ACM Sigplan Notices 52, 4, 615–629.

[107]Kazuhiro Tokunaga, Kenichi Kawamura, and Naoki Takaya. 2016. High-speed uploading architecture using distributed edge servers on multi-RAT heterogeneous networks. In Proceedings of the IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN'16). IEEE, 1–2

[108]Sokol Kosta, Andrius Aucinas, Pan Hui, Richard Mortier, and Xinwen Zhang. 2012. Thinkair: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading. In Proceedings of the IEEE Conference on Computer Communications (INFOCOM'12). IEEE, 945–953.

[109]Lei Gao, Mike Dahlin, Amol Nayate, Jiandan Zheng, and Arun Iyengar. 2003. Application specific data replication for edge services. In Proceedings of the 12th International Conference on World Wide Web. ACM, 449–460.

[110]Yi Lin, Bettina Kemme, Marta Patino-Martinez, and Ricardo Jimenez-Peris. 2007. Enhancing edge computing with database replication. In Proceedings of the 26th IEEE International Symposium on Reliable Distributed Systems (SRDS'07). IEEE, 45–54.

[111]Zhuo Chen, Lu Jiang, Wenlu Hu, Kiryong Ha, Brandon Amos, Padmanabhan Pillai, Alex Hauptmann, and Mahadev Satyanarayanan. 2015. Early implementation experience with wearable cognitive assistance applications. In Proceedings of the Workshop on Wearable Systems and Applications. ACM, 33–38.

[112]Nan Wang, Blesson Varghese, Michail Matthaiou, and Dimitrios S. Nikolopoulos. 2017. ENORM: A framework for edge node resource management. IEEE Trans. Serv. Comput. (2017).

[113]Miguel Báguena, George Samaras, Andreas Pamboris, Mihail L. Sichitiu, Peter Pietzuch, and Pietro Manzoni. 2016. Towards enabling hyper-responsive mobile apps through network edge assistance. In Proceedings of the 13th IEEE Consumer Communications & Networking Conference (CCNC'16). IEEE, 399–404.

[114]Thang X. Vu, Symeon Chatzinotas, and B. Ottersten. 2017. Energy-efficient design for edge-caching wireless networks: When is coded-caching beneficial? In Proceedings of the IEEE 18th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC'17). 1–5. DOI:10.1109/SPAWC.2017.8227689

[115]Nitinder Mohan and Jussi Kangasharju. 2016. Edge-fog cloud: A distributed cloud for Internet of things computations. In Proceedings of the Cloudification of the Internet of Things (CIoT'16). IEEE, 1–6

[116]Xu Chen and Junshan Zhang. 2017. When D2D meets cloud: Hybrid mobile task offloadings in fog computing. In

Proceedings of the IEEE International Conference on Communications (ICC'17). IEEE, 1–6.

[117]Alexandru Stanciu. 2017. Blockchain based distributed control system for edge computing. In Proceedings of the 21st International Conference on Control Systems and Computer Science (CSCS'17). IEEE, 667–671

[118]Xu Chen, Lei Jiao, Wenzhong Li, and Xiaoming Fu. 2016. Efficient multi-user computation offloading for mobile-edge cloud computing. IEEE/ACM Trans. Netw. 24, 5 (2016), 2795–2808.

[119]Yuvraj Sahni, Jiannong Cao, Shigeng Zhang, and Lei Yang. 2017. Edge mesh: A new paradigm to enable distributed intelligence in Internet of things. IEEE Access 5 (2017), 16441–16458.

[120]Floris Van den Abeele, Jeroen Hoebeke, Girum Ketema Teklemariam, Ingrid Moerman, and Piet Demeester. 2015. Sensor function virtualization to support distributed intelligence in the internet of things. Wirel. Person. Commun. 81, 4 (2015), 1415–1436.

[121]Peng Liu, Dale Willis, and Suman Banerjee. 2016. Paradrop: Enabling lightweight multi-tenancy at the network's extreme Edge. In Proceedings of the IEEE/ACM Symposium on Edge Computing (SEC'16). IEEE, 1–13

[122]Roberto Morabito, Vittorio Cozzolino, Aaron Yi Ding, Nicklas Beijar, and Jorg Ott. 2018. Consolidate IoT edge computing with lightweight virtualization. IEEE Network 32, 1 (2018), 102–111.

[123]Singh, S., & Chana, I. (2015). QoS-aware Autonomic Resource Management in Cloud Computing: A Systematic Review. ACM Computing Surveys, 48(3), 1–46. doi:10.1145/2843889

[124]Oriol Sallent, Jordi Perez-Romero, Ramon Ferrus, and Ramon Agusti. 2017. On radio access network slicing from a radio resource management perspective. IEEE Wirel. Commun. 24, 5 (2017), 166–174.

[125]J. Park, J.-H. Choi, Y.-J. Lee, and O. Min, "A Layered Features Analysis in Smart Farm Environments," in Proceedings of the International Conference on Big Data and Internet of Thing - BDIOT2017, London, United Kingdom, 2017, pp. 169–173, doi: 10.1145/3175684.3175720.

[126]India Agriculture and Climate Data Set. (n. d.). Retrieved from https://ipl.econ.duke.edu/dthomas/dev_data/ datafiles/india_agric_climate.htm

[127]Ivan Stojmenovic. 2014. Fog Computing: A cloud to the ground support for smart things and machine-to-machine networks. In Proceedings of the Australasian Telecommunication Networks and Applications Conference. 117–122.

[128]Yu Xiao, Marius Noreikis, and Antti Ylä-Jaäiski. 2017. QoS-oriented capacity planning for edge computing. In Proceedings of the IEEE International Conference on Communications (ICC'17). IEEE, 1–6.

[129]Paolo Bellavista and Alessandro Zanni. 2017. Feasibility of fog computing deployment based on Docker containerization over Raspberry Pi. In Proceedings of the 18th International Conference on Distributed Computing and Networking.ACM, 16.

[130]Hua-Jun Hong, Jo-Chi Chuang, and Cheng-Hsin Hsu. 2016. Animation rendering on multimedia fog computing platforms. In Proceedings of the IEEE International Conference on Cloud Computing Technology and Science. IEEE,336–343.

[131]Mohammad Aazam and Eui-Nam Huh. 2014. Fog computing and smart gateway based communication for cloud of things. In Proceedings of the International Conference on Future Internet of Things and Cloud (FiCloud'14). IEEE,464–470.

[132]Abhimanyu Gosain, Mark Berman, Marshall Brinn, Thomas Mitchell, Chuan Li, Yuehua Wang, Hai Jin, Jing Hua, and Hongwei Zhang. 2016. Enabling campus edge computing using GENI racks and mobile resources. In Proceedings of the IEEE/ACM Symposium on Edge Computing (SEC'16). 41–50.

[133]Paul Barham, Boris Dragovic, Keir Fraser, Steven Hand, Tim Harris, Alex Ho, Rolf Neugebauer, Ian Pratt, and Andrew Warfield. 2003. Xen and the art of virtualization. In ACM SIGOPS Op. Syst. Rev., Vol. 37. 164–177.

[134]Avi Kivity, Yaniv Kamay, Dor Laor, Uri Lublin, and Anthony Liguori. 2007. KVM: The Linux virtual machine monitor. In Proceedings of the Linux Symposium, Vol. 1. 225–230.

[135]Nicholas Haydel, Sandra Gesing, Ian Taylor, Gregory Madey, Abdul Dakkak, Simon Garcia De Gonzalo, and WenMei W. Hwu. 2015. Enhancing the usability and utilization of accelerated architectures via Docker. In Proceedings of the IEEE/ACM 8th International Conference on Utility and Cloud Computing. 361–367.

[136]Stephen J. Vaughan-Nichols. 2006. New approach to virtualization is a lightweight. Computer 39, 11 (2006).

[137]Raja Wasim Ahmad, Abdullah Gani, Siti Hafizah Ab Hamid, Muhammad Shiraz, Abdullah Yousafzai, and Feng Xia. 2015. A survey on virtual machine migration and server consolidation frameworks for cloud data centers. J. Netw. Comput. Appl. 52 (2015), 11–25.

[138]Mattias Forsman, Andreas Glad, Lars Lundberg, and Dragos Ilie. 2015. Algorithms for automated live migration of virtual machines. J. Syst. Softw. 101 (2015), 110–126.

[139]Diego Kreutz, Fernando M. V. Ramos, Paulo Esteves Verissimo, Christian Esteve Rothenberg, Siamak Azodolmolky, and Steve Uhlig. 2015. Software-defined networking: A comprehensive survey. Proc. IEEE 103, 1 (2015), 14–76.

[140]Stephan Reiff-Marganiec, Marcel Tilly, and Helge Janicke. 2014. Low-latency service data aggregation using policy

obligations. In Proceedings of the IEEE International Conference on Web Services. IEEE, 526–533.

[141]Genc Tato, Marin Bertier, and Cédric Tedeschi. 2017. Designing overlay networks for decentralized clouds. In Proceedings of the IEEE International Conference on Cloud Computing Technology and Science (CloudCom'17). IEEE,391–396.

[142]Abhishek Chandra, Jon Weissman, and Benjamin Heintz. 2013. Decentralized edge clouds. IEEE Internet Comput. 17,5 (2013), 70–73.

[143]Mathew Ryden, Kwangsung Oh, Abhishek Chandra, and Jon Weissman. 2014. Nebula: Distributed edge cloud for data intensive computing. In Proceedings of the IEEE International Conference on Cloud Engineering. 57–66.

[144]Kirak Hong, David Lillethun, Umakishore Ramachandran, Beate Ottenwälder, and Boris Koldehofe. 2013. Mobilefog: A programming model for large-scale applications on the internet of things. In Proceedings of the 2nd ACM SIGCOMM Workshop on Mobile Cloud Computing. 15–20.

[145]Stefan Nastic, Thomas Rausch, Ognjen Scekic, Schahram Dustdar, Marjan Gusev, Bojana Koteska, Magdalena Kostoska, Boro Jakimovski, Sasko Ristov, and Radu Prodan. 2017. A serverless real-time data analytics platform for edge computing. IEEE Internet Comput. 21, 4 (2017), 64–71.

[146]Brian Amento, Bharath Balasubramanian, Robert J. Hall, Kaustubh Joshi, Gueyoung Jung, and K. Hal Purdy. 2016. FocusStack: Orchestrating edge clouds using location-based focus of attention. In Proceedings of the IEEE/ACM Symposium on Edge Computing (SEC'16). IEEE, 179–191.

[147]Enrique Saurez, Kirak Hong, Dave Lillethun, Umakishore Ramachandran, and Beate Ottenwälder. 2016. Incremental deployment and migration of geo-distributed situation awareness applications in the fog. In Proceedings of the 10th ACM International Conference on Distributed and Event-based Systems. ACM, 258–269.

[148]Roman Kolcun, David Boyle, and Julie A. McCann. 2015. Optimal processing node discovery algorithm for distributed computing in IoT. In Proceedings of the 5th International Conference on the Internet of Things (IOT'15). IEEE,72–79.

[149]Roberto Morabito. 2017. Virtualization on internet of things edge devices with container technologies: A performance evaluation. IEEE Access 5 (2017), 8835–8850.

[150]Velliangiri, S., Rajagopal, R., & Karthikeyan, P. (2019). Trust Factor Based Key Distribution Protocol in Hybrid Cloud Environment. Scalable Computing: Practice and Experience, 20(2), 349-364.

[151]Velliangiri, S., Karthikeyan, P., Xavier, V. A., & Baswaraj, D. (2020). Hybrid electro search with genetic algorithm for task scheduling in cloud computing. Ain Shams Engineering Journal.

[152]Manikannan K and Nagarajan V. (2020). Optimized Mobility Management for RPL / 6LoWPAN based IoT network architecture using the firefly algorithm, Microprocessors and Microsystems, 2020, vol. 77, pp.820.

[153]Alexandr Krylovskiy. 2015. Internet of things gateways meet Linux containers: Performance evaluation and discussion. In Proceedings of the IEEE 2nd World Forum on Internet of Things (WF-IoT'15). IEEE, 222–227.

[154]Konstantinos E. Parsopoulos, Michael N. Vrahatis et al. 2002. Particle swarm optimization method for constrained optimization problems. Intell. Technol.—Theor. Appl.: New Trends Intell. Technol. 76, 1 (2002), 214–220.

[155]Xiuli He, Zhiyuan Ren, Chenhua Shi, and Jian Fang. 2016. A novel load balancing strategy of software-defined cloud/fog networking in the internet of vehicles. China Commun. 13, 2 (2016), 140–149.

[156]V. B. Souza, Xavier Masip-Bruin, Eva Marín-Tordera, Sergio Sànchez-López, Jordi Garcia, Guang-Jie Ren, Admela Jukan, and A. Juan Ferrer. 2018. Towards a proper service placement in combined Fog-to-Cloud (F2C) architectures.Fut. Gen. Comput. Syst. 87 (2018), 1–15.

[157]Mohit Taneja and Alan Davy. 2017. Resource aware placement of IoT application modules in Fog-Cloud computing paradigm. In Proceedings of the IFIP/IEEE Symposium on Integrated Network and Service Management (IM'17). IEEE,1222–1228.