Journal of Engineering and Technology

ISSN 2180-3811 eISSN 2289-814X https://jet.utem.edu.my/jet/index

FOG COMPUTING WITH INTERNET OF THINGS: AN OVERVIEW OF ARCHITECTURE, ALGORITHMS, CHALLENGES AND APPLICATIONS

K. K. Vaigandla*[1] and M. Siluveru[1]

[1] Electronics & Communication Engineering, Balaji Institute of Technology and Science, Telangana, India.

corresponding_vkvaigandla@gmail.com

Article history:
Received Date: 25 January 2023
Revised Date: 7 April 2023
Accepted Date: 2 May 2023

Keywords: Fog Computing (FC), Cloud Computing (CC), Communication, Networking,

Abstract— Fog is an early design for data storage, computing, and application control. It includes both control and data planes and is useful in wired and wireless contexts. As a 5G, Internet of Things (IoT), and integrated artificial intelligence (AI) architecture, it supports an expanding variety of applications. Performance, security, latency, and network failure are just a few of the problems that integrated cloud computing (CC) must contend with as IoT applications continue to develop. The evolution of fog computing (FC) offers a solution to these issues by putting CC closer to the IoT. The main purpose of the fog is to deliver the data produced by the edge IoT devices. Instead

This is an open-access journal that the content is freely available without charge to the user or corresponding institution licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International (CC BY-NC-ND 4.0).

Internet of Things (IoT), IoT Devices

of sending the data to a cloud server, local processing and data storage are carried out at the fog node (FN). When compared to the cloud, FC offers high-quality, quick-response services. FC may thus be the best choice for enabling the IoT to provide a reliable and highly secure service to a large number of IoT customers. In this article, we present the characteristics, architecture, integration with the IoT, and problem of FCs.

I. Introduction

Fog is essentially a structural architecture that distributes the processing, communications, controls, and storage facilities to the end users along the many cloud-to-things continuum options. The phrases "fogs" and "edges," despite the fog is larger than the typical idea of edges, are frequently used synonymously [1,2]. Moving processing, control, and data storage onto the cloud has been popular during the past ten years. The main tasks that are moved to centralized data centers, backbone IP networks, and cellular core networks include computation, storage, and network management [3-7]. However, as the IoT develops, there are more and more needs that cloud computing must satisfy. At the same time, a wide range of potent end-user, network edge, and access devices have proliferated, including smart phones, tablets, smart home appliances, small cellular base stations, edge routers, traffic control cabinets along the roadside, connected vehicles, smart meters and energy controllers in a smart power grid, smart building controllers, and manufacturing control systems, to name a few. Following closely after are several other smart clients and edge devices, including information-transmitting light bulbs, portable laptops, and button-sized Radio Frequency tuners [3-7].

A common characteristic of mobile communications is the five-generation (5G) architecture. Mobile device development has been driven by a variety of variables, some of which are communications-related, like providing high-speed mobile connectivity to densely populated places, and others which are less communication-related, such having a battery life of over 10 years, among others. Traffic increase is influenced by a few factors, such as the rising demand for enhanced mobile broadband (eMBB), ultra-reliable and low latency transport modes (URLLC), and the needs posed by massive machine communications (MMTC) and the massive Internet of Things (MIoT). Thanks to 5G, mobile communications will become more and more significant in the Industrial IoT, particularly in terms of needs for ultra-reliable and low-latency connection [8-10].

As time goes on, more and more businesses and people rely on desktop computers and intelligent gadgets to do daily tasks. These sophisticated systems produce information using a variety of applications and sensors. As a result, businesses continually produce and store vast amounts of information [11]. After the development of the IoT, the amount of information produced by various types of sensors is increasing. Big data analytics has received excessive attention recently because of the quick increase in the volume of information being produced and the inability of traditional databases to manage various types of structured and disorganized information. Numerous firms are analyzing the data gathered from numerous gadgets to derive the necessary understanding for making important choices [12]. Many companies nowadays require a robust cloud-based infrastructure since everything is moving to the cloud due to its pay-per-use, scalability, and accessibility capabilities. Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service are the most popular cloud services provided by CC (SaaS). Everything as a Service (XaaS) is the direction that all these cloud services are moving in [13]. However, the data generated by these countless millions of sensors, known as "Big Data," cannot be fully analyzed and sent to the cloud as this may significantly increase latency. Additionally, few IoT applications demand faster processing than CC's current capabilities. FC, which connects the processing power of smart devices near the client to support the use of networking, computing, and storage close to the edge, can solve this problem [14]. Fog with IoT can reduce the amount of data transferred to CC for archiving, processing, analysis, and performance improvement. Fog is a fully distributed computing solution; unlike CC [15-16], it is not wholly dependent on any integrated component. By making use of the idle resources of several nearby devices, fog may be exploited to solve the CC's latency problem. However, most of the work is delegated to CC. Fog, in contrast to CC, is a distributed computing solution in which several devices close to the clients employ computing capabilities with less features but strong processing capacity with multiple cores. As a result, a number of smart devices are installed with storage and processing capability that may function as FC devices, including switches, base stations, routers, smart phones, and network device management [3-7]. Global connection and different organizational structures have led to the evolution of several research issues related to FC. The main problems with the FC idea are its needs and deployable environment. This explains why there are several computing methods used in the FC field.

II. IoT for new Architectures: Challenges

A digital development that can facilitate data collection and exchange is the "IoT" Never before has this been attempted. It makes communication possible and more securely reports user data [3]. By 2020, it is anticipated that there will be more than 40 billion smart devices or gadgets connected to the internet. IoT has been the subject of several findings to track problems and difficulties relating to its design and architecture [3-7]. Many new difficulties brought on by the developing IoT cannot be fully handled by the Cloud and host computing paradigms of today. Here, we go over a number of these core difficulties.

A. Stringent Latency Requirements

End-to-end latencies between the sensor and the control node must frequently remain under a few milliseconds in many industrial control systems, including manufacturing systems, smart grids, oil and gas systems, and commodities packaging systems [17]. Many more IoT applications may need latencies below a few tens of milliseconds, including vehicle-to-vehicle (V2V) communications, vehicle-to-roadside communications, drone flight control apps, virtual reality applications, gaming applications, and real-time financial trading applications. These requirements are well above what the bulk of Cloud services can handle.

B. Limitations on Network Bandwidth

Data creation is accelerating due to the enormous and quickly expanding number of linked items [18]. It is anticipated that the US smart grid would produce 1000 petabytes of data annually. 2.4 petabytes of data were produced by the US Library of Congress each month, one petabyte was used by Google each month, and 200 petabytes were used by AT&T's network in 2010 [19]. The amount of network bandwidth needed to send all the data to the cloud will be unreasonably large. Because of rules and worries about data privacy, it is frequently unneeded and occasionally even illegal. 90% of the data produced by the endpoints, according to ABI Research's estimate, will be processed and kept locally rather than in the cloud [18].

C. Devices Limited by Resources

The resources of many IoT devices will be quite constrained. Examples include sensors, data collectors, actuators, controllers, security cameras, transportation systems like trains and autos as well as implanted medical equipment in people. The limited resources of many resource-constrained devices will not be sufficient to meet all their computational requirements. Because these connections frequently call for resource-intensive processing and complicated protocols, requiring all of them to interface directly with the Cloud will be impracticable and expensive.

D. Cyber-Physical Systems

The pendulum between the "brick" and the "click" is starting to swing back toward the "brick" once more as more cyber-physical systems are connected to the IoT. In this scenario, interactions, and frequently close integrations, among both cyber systems and physical systems are becoming increasingly relevant and introduce additional business priorities and operational requirements. Cyber-physical systems examples include networked automobiles and railways, smart cities, and industrial control systems. Continuous and secure functioning is frequently given top emphasis in such systems. Because taking a system offline for any reason might result in major financial loss or unbearable consumer annoyance, it is sometimes necessary to plan days, weeks, or even months [20].

E. Security Challenges

Existing perimeter-based security measures for the Internet have been the focus of cyber security solutions for today's business networks, data centers, and consumer gadgets. To prevent security threats from breaching the secured perimeters, a system or individual device under protection is specifically put behind firewalls that collaborate with intrusion detection and prevention systems. Additionally, certain laborintensive security operations are being shifted to the cloud. Existing cloud-based security services continue to priorities perimeter-based security, such as forwarding access control requests to the clouds for authentication and authorization processing and email and web traffic to the clouds for threat detection.

F. Protecting Devices with Limited Resources

Many IoT devices with limited resources won't have the resources to fully secure themselves. These gadgets could last a very long time, and updating their hardware and software might not be feasible. These gadgets must, nevertheless, maintain their security during their lengthy lifespan. For instance, changing any hardware on a car that has already been sold to a customer can be very inconvenient for the owner and result in high costs and reputational harm to the automaker. However, due to the lengthy lifespan of an automobile, which is 11.4 years on average [21], security risks will become substantially more sophisticated, a lot of new threats will emerge, and the defenses needed to counter these threats will need to be improved and upgraded in accordance.

III. Fog Computing: Overview

The definition of FC is "an very virtualized environment that offers networking, storage, and compute capabilities across obsolete CC information centers, typically, but not totally positioned at the network edge" [22]. Numerous edge nodes with limited processing capabilities, also known as fog nodes (FNs), are present in a fog structure. These FNs have less processing and storage facilities. When there are several servers and an edge in a fog network, they are referred to as cloudlets [23-24] and they participate in the shared computing environment rather than being on the network edge. The clients may be able to get a real-time response for delicate latency applications by employing this fog device.

A. Features

Adaptability: The surrounding environment is monitored by a vast network of sensors. The fog provides distributed processing and storage capabilities that can work with such a wide range of end devices.

Real time communications: When using cloud batch analysis, FC solicitations offer simultaneous communication between fog nodes.

Physical distribution: In contrast to the integrated cloud, fog offers decentralized applications and services that may be hosted anywhere.

Less latency and position awareness: Since fog is close to edge devices, there is reduced waiting time for processing edge devices' information. Additionally, it helps with position responsiveness by allowing fog nodes to be hosted everywhere.

Compatibility: Through a variety of service providers, fog modules can adapt to and work with many platforms that are not the same.

Provisions for web-based analytics and integration with cloud: To play a crucial role in the speed and computation of the information close to the edge devices, fog is placed between edge devices and cloud.

Heterogeneity: Since they are created by different firms and have different origins, edge devices or fog nodes require hosting according to where they will be used. Fog may therefore adapt to different systems.

Provision for flexibility: The ability of fog solicitations to connect directly to devices like mobiles and so enabling flexibility approaches, such Locator ID Separation Protocol (LISP), which requires a distributed indexed system, is one of its key characteristics.

The feature of FC is depicted in Figure 1.

B. Architecture

FC is a technique that moves a few data centre processes to the network edge. As part of a shared strategy between CC data centers and end devices, it provides less storage, processing, and service network. The primary goal of FC is to provide less unpredictable latency for time sensitive IoT tasks [25]. Numerous researchers have created numerous reference architectures for fog. The fog computing framework is shown in Figure 2.

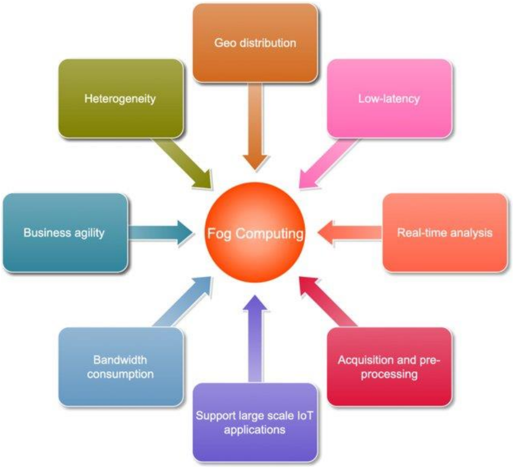

Figure 1: Features of FC

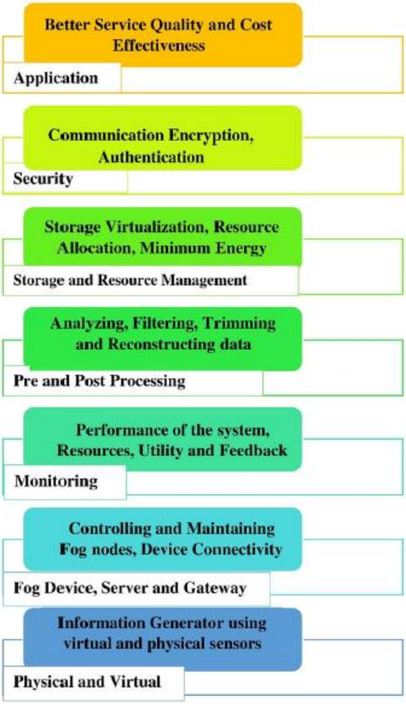

Figure 2: Fog computing framework

Physical and virtual sensors: The fundamental information generator of FC is the many types of information generated by the sensors [26]. This data may be generated by a variety of devices, including smart houses and appliances, CCTV and traffic surveillance systems, automated driving cars, humidity, and temperature sensors, and more. For instance, a constant traffic status of all the routes must be obtained from various sensors, devices for pathways, and roadside CCTV monitoring to manage traffic lights in an intelligent traffic surveillance system. It is crucial to foresee future traffic needs by compiling data from various GPS sensors. When there is a traffic collision, virtual sensors, in addition to physical ones, are crucial [27].

Fog device, server, and gateway: A fog server, fog device, or gateway might be an IoT or standalone device [28-29]. The fog server, however, as it manages more fog devices, must have a better setup than the fog gateway and devices. Numerous parameters, including hardware setup, network connectivity, the ability to control devices, etc., are necessary for the fog server to function. The fog server's function is established in accordance with its IoT component. Networked to fog devices is a group of real and virtual sensors. The fog server will also be networked to a cluster of fog devices. In comparison to the fog device, it should also have more sophisticated calculation and storage capabilities. When necessary, a specific set of fog devices connected to the same server can broadcast with one another. Few applications computations in an intelligent transport use case could be decided by different fog groups.

Monitoring: The monitoring level, together with utility and feedback, keeps tabs on the performance of the system and resources. The appropriate resources are chosen as operating system monitoring tools. In situations with smart transportation systems, several processes take place. There is a chance that the resources won't be available in a situation where calculations or storage on a fog device are required. Similar events might take place on the fog server side. The devices and servers on the fog side will seek assistance from various peers to deal with such conditions. Therefore, these will be efficiently decided by the system monitoring components. The component of resource demand examines the available resources and forecasts future resources based on user behaviour and consumption.

Pre and post processing: It has several components and analyses both simple and complex data. This level's function is to collect the data by analyzing, filtering, trimming, and reconstructing it as needed. After the data has been processed, the data flow component decides where it should be stored, either locally at the fog or in the cloud for long-term storage [28]. FC's severe problem is that data is processed at the edge and only a minimal amount of data needs to be stored. The fundamental concept is to transfer data that will be needed often to fog servers and data that will be utilized seldom or for an extended length of time to the cloud. A smart transportation application generates data from several sensors. To obtain the created information, this data will be examined and processed. This produced data might not be of any value.

Storage and resource management: Data storage via storage virtualization is the responsibility of the storage module. Data backup is the component in charge of preventing data loss and ensuring data availability. The idea of storage virtualization involves a collection of hardware that manages storage over a network and functions as a single storage device. This personal storage unit is simple to operate and maintain. The key advantage of storage virtualization is that it improves corporate functionality while keeping hardware and storage costs low. Additionally, it reduces storage complexity. Data backups are essential since there is a potential that the storage system might malfunction [30]. The data backup module is in charge of regularly customizing data backup plans. The components at the resource management level deal with concerns related to energy conservation as well as resource allocation and scheduling. Reliability is a component that ensures the application's scheduling and resource allocation are reliable. The scalability of the fog resources is guaranteed at times of heavy resource demand. The cloud platform succeeds in achieving horizontal scalability, while the fog platform concentrates on achieving both vertical and horizontal scalability [31]. When storage is to be carried out in a system of distributed resources, a major issue in the allocation process of the distributed resources occurs.

Security: All security-related concerns, such as communication encryption and secure information storage, are maintained at the security level. The information of the users of the fog is likewise secured at this level. It is suggested that the fog environment be established as a utility system like a cloud environment. In a cloud environment, users connect to the cloud to request all services, but in a fog environment, users connect to the fog system to request all services, with the fog middleware managing and maintaining all connections with the cloud. As a result, the user who wants to connect to a service must be approved. Therefore, the validation component is in charge of sending authentication requests to each and every user in the fog [32]. It is crucial to maintain security by using encryption between various communications to prevent the entry of harmful users. The encryption component can encrypt different connections made between the cloud and IoT devices. Since wireless connections connect most of the fog components, security must be preserved.

Application: FC was initially developed to enable IoT applications [33]. Since then, several WSN-based applications have started to support FC. Nearly all apps that struggle with latency began to benefit from the fog environment. Any utility service that might work with fog to improve performance and cut costs was included in this category. The fog environment can accommodate the needs of processing in real-time while employing augmented reality, which can result in a sustained improvement in several augmented reality services.

A comparison between FC and CC is tabulated in Table 1.

Table 1: FC vs CC

S. No.	Parameters	Fog Computing	Cloud Computing
1	Hosting dispersed integrated	Hosting dispersed integrated	Hosting dispersed integrated
2	Positing of server nodes	At the confined network edge	Within the web
3	Delay	Less	More
4	Range amidst server	Single hop	Many hops
5	Hardware	Less computing energy and storage	Expending computing energy and storage
6	Information	Attack more likeliness	Less likeliness
7	Position knowledge	Accepted	Denied

IV. FC with the IoT

Different problems are being encountered with the current integrated CC framework for IoT applications. For instance, it is not possible to implement time-sensitive requirements like augmented reality, audiovisual streaming, and gaming [34]. In addition, because it is an integrated prototype, it lacks position responsiveness. FC addresses these problems.

The fog can effectively supply immediate transmission for a variety of IoT requests, such as connected autos, etc. For applications that have less waiting time requirements, such as augmented reality, audiovisual streaming, and gaming, FC is well-suited [35]. IoT integration with FC will bring various benefits to various IoT demands. FC enables immediate communication between IoT devices to reduce waiting times, especially for time-sensitive IoT queries. Additionally, one of the important features of FC is its capacity to support massive sensor networks.

As seen in Figure 3, FC might benefit various IoT applications in a number of ways. The many research publications on the connections between IoT and FC are discussed in this section. In this section, we look at recent publications that address how IoT and FC are combined in various applications. Numerous linked research papers discuss various aspects of FC. In [36], FC and its immediate applications are discussed. It demonstrates data produced by IoT devices that fog can process. Additionally, it lists the issues with congestion and waiting times that fog can alleviate. The paper also demonstrates how FC may help a smart platform manage decentralized IoT setups that are continuously evolving and enhance unique services at the edge network, with the goal of establishing various business concepts and opportunities for network operators. The integration of IoT and fog is described in [37] in general terms. The authors began by discussing several issues with creating IoT systems and how difficult it is to identify these issues using current networking and computing architectures. Additionally, they identified and specified three circumstances in which flexibility support is essential for FC and the IoT. A survey of an FC reference framework prototype was conducted in [14]. The prototype supports the need for IoT to be localized in the fog rather than utilizing the cloud. The core fog services are in a software-specified resource control layer in the reference framework [38]. This provides a middleware that runs in the cloud and prevents fog clusters from acting independently. Instead, cloud-based middleware is used to analyze, track, and organize fog cells. Additionally, the integration of fog with IoT was explored in [39], which included key details on the fog's characteristics and how it disseminated CC by enhancing it through the fog. The FC paradigm's proposed model has been investigated in[40]. The authors claimed that their prototype was the best way to support IoT devices that had resource constraints. To demonstrate the versatility of their fog rules, they also provide three impulsive states, including wireless sensor networks (WSNs), intelligent vehicles, and an intelligent grid. Additionally, in [41] studied fog mobile, which can distribute IoT applications across several devices in an organized architecture from the edge network to CC. To model service delay, comprehend, and evaluate fog-IoT-CC application scenarios, an architecture has been proposed in [42]. For the fog nodes that tend to minimize latency services for IoT nodes, the authors have proposed a way for lowering latency. They recommended a technique that creates fog to fog transmission to distribute the load and reduce service delays. The approach for offloading computing considers several demand categories with varying calculation times in addition to the queue's overall length. A use scenario including a man-in-the-middle attack was employed to examine how memory and the central processor unit (CPU) of fog devices are utilized. In their taxonomy of FC, Mahmud et al. [43] discussed the differences between mobile cloud computing, edge computing, and FC. Additionally, the design of the fog node, various FC settings, and fog networking devices were studied.

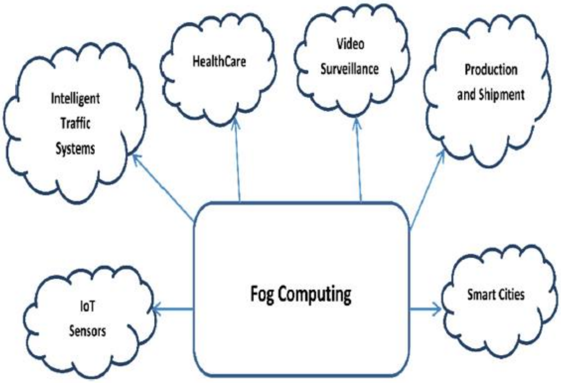

Figure 3: IoT applications produce a number of advantages provided by FC

V. Computing Algorithms Fog System

A basic overview of fog system techniques, including heterogeneity points, QoS management, scalability, mobility, federation, and interoperability were provided in [44]. Fog components may execute a variety of tasks independent of the environment they are placed in terms of computing and storage requirements because to heterogeneity. Fog systems improve user experience by lowering latency and processing time through QoS management. Scalability is the key to implementing FC on a broad horizontal and vertical spectrum in terms of user and application numbers. This element needs to be flexible because there might not always be a necessity for a lot of fog components. Mobility allows fog components to be moved and repositioned as needed to carry out a task while under operation. V2V communication is an example of this [45]. Fog infrastructure may be regarded from an architectural and computational perspective. In the same way that architecture may be classified into application-agnostic architectures and application-specific architectures, fog algorithms can be further divided into computation, content storage and delivery, energy consumption, and application-specific algorithms. This paper discusses the work that has been done in the algorithmic environment about computing in fog systems since the topic of FC is quite broad. System efficiency may be increased by node pairing in the same fog domain using a utility-based technique once nodes in a fog network have been analyzed and paired. To do this, Irving's matching algorithm is improved, and a model for a one-sided, stable matching game with a quota is developed [46]. The creation of a utility-based list considers transmission power, transmission distance, and cost. By creating preference lists for each node to many other nodes in the fog domain, the utilities couple up the nodes. FC applications can improve user experience by lowering power consumption or latency for each user by allowing radio access clustering. The computation might be split among a few cells to increase the performance of the fog components and lessen network traffic. To do this, the computational burden is distributed across nearby radio access points having computing capabilities. Forming a cluster to complete a job with the least amount of delay and another to use less energy are two methods for generating clusters [47]. Clusters are scalable and have internal communication and resource allocation. A mathematical framework for resource sharing is built by mapping certain parameters to temporal resources [48]. Allocating resources amongst nodes in a fog system is the key problem. Performing mathematical computations or downloading data is a straightforward way, but it ignores the service being provided. This may lead to an increase in latency for applications that require a lot of delay, such augmented reality. Thus, processing is reduced by executing task-oriented sharing. This is accomplished by developing a uniform framework to distribute resources by mapping resources such as power, bandwidth, and latency. Convex optimization techniques are used to build optimization problems from these parameters.

A. Computation Management

Effective control of computing activity at FNs prolongs their life and improves their performance. FNs are to be given the resource-demanding tasks performed by end devices with restricted resources. This method is sometimes referred to as unloading. Offloading advantages can be realized, nevertheless, if the best timing, location, content, and method are chosen [49]. Offloading algorithms are created to determine the location of the offloaded work as well as whether the task should be offloaded completely or in part. Meng et al. [50]'s decision to choose cloud-only, cloud-first, fog-only, or fog-first processing reduces overall energy usage. However, their plan does not consider other crucial QoS elements like availability, secrecy, and authentication. Under the restrictions of authentication, secrecy, and availability as well as integrity, capacity, speed, and cost in the mobile IoT environment, Rahbari and Nickray [51] also choose the optimum location for offloading. Their plan does not include fault tolerance, and it may perform better if machine learning techniques were added. The classification of the fog algorithms is shown in Figure 4.

Application Placement: To decrease application response time, deadline violations, and cost, [52] explains the precise optimization technique, a greedy first-fit heuristic, and a genetic algorithm to find the location. Their plan makes no mention of the movable fog landscape's fault tolerance. Fuzzy logic-based methods are used by Mahmud et al. [53] to calculate the Rating of Expectation, or the priority value of the request, and the Capacity Class Score, or the status of FN. They can connect the placement requests to a fog instance with that linearly optimized mapping. Their plan has not yet been translated and put to the test in a real-world setting. Xia et al. [54] present two backtracking algorithms, two heuristics, and two heuristics to decrease the average reaction time and enhance the placement policy's quality and scalability.

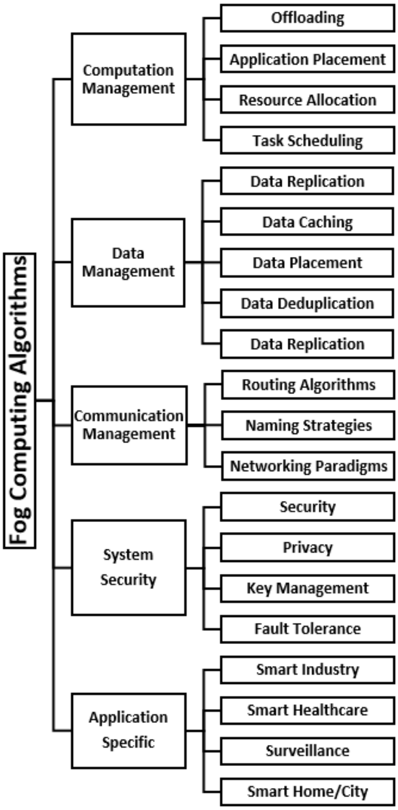

Figure 4: Fog Algorithms: Classification

Resource Allocation: The techniques are suggested to distribute the scarce fog resources to users in an efficient manner. FC and/or fog networking resources have been the subject of much study. The pricing resource is established using a model of the Stackelberg games, and many-to-many matching is used to distribute the resources, according to Zhang et al. Two bidding schemes, a continuous demand scheme and a multi-demand scheme are suggested in the offline auction-based mechanism put out by Jiao et al. [55] to maximize social welfare. To improve cost effectiveness, Jia et al. [56] suggested a double matching method based on a Deferred Algorithm for a double two-sided matching game. The effectiveness of the plan will be evaluated considering the use of UAVs and additional architectural hierarchy levels [57]. AHP is used by Abedin et al. [58] to order the service requirements. A one-to-many matching game is used to find a solution to the defined college admissions dilemma to optimize QoS. After that, the best fit RA technique is used to maintain user association stability.

Task Scheduling: The task, the job, or the process may all be scheduled using task scheduling approaches. The smallest autonomous unit that cannot be split further is called a task. A job can be broken down into several parts. Workflow is a collection of interdependent tasks, each of which depends on the completion of the one before it. Tychalas and Karatza [59] suggest workload-based destination selection for the bag of tasks to cut costs. Their plan barely slows down the reaction time. Virtual Machines (VMs) with low, medium, and high capabilities conduct low, medium, and high priority tasks, respectively, in Aladwani's work [60]. However, their plan ignores location awareness and security concerns while choosing VMs. The challenge of minimizing the maximum job completion time is resolved by Zeng et al. [61] utilizing the linear programming relaxation approach.

B. Data Management

Fog receives a tonne of data from related IoT sensors. Fog's data management becomes crucial for the timely and efficient use of the data that has been saved. By using a single integer programme to get the precise answer, Naas et al. [62] overcome the data placement problem. The extensive fog infrastructure was used to test the combinatorial explosion. To shorten the solving time, a geo-partitioning based heuristic approach was presented. The strategy, however, ignores the interdependence of data flow and task balance in geo-partitioned subparts. To decrease the latency of data transfer from the storage node to the consumer node using a k-way graph partitioning, Naas et al. [63] propose a divide and conquer based strategy. The model of multi replica data insertion is suggested by Huang et al. [64]. To select a solution with the least amount of delay, their programme uses the pruning technique. Their job entails balancing replica count and performance in the reduction of total delay. Wang and Wu [65] investigate multiple data placement with a financial challenge. For a specific total budget, the authors attempt to reduce the overall data access latency. Their plan fails to consider various requests of various proportions. The research in [66] employs FC for storage expansion and frictional linear programming to identify the best places to deposit data.

C. Communication Management

A hash-based naming approach applicable for the Fog-cloud environment was developed by G'omez-C'ardenas et al. [67]. Content-Centric Network is used by Guibert et al. [68] to increase the effectiveness of communication and local storage. With a deadline and bandwidth restriction, Kadhim and Seno [69] introduce EEMSFV, an Energy Efficient Multicast Routing Protocol for Vehicular Networks based on Software Defined Networks and FC. A hierarchical routing technique that is energy-efficient is proposed by Abidoye and Kabaso [70]. They employ an ant colony optimization to reduce the overall volume of broadcast packets. Data transmission within the network, however, is vulnerable to many threats and requires suitable security measures.

D. Algorithms for Security

By keeping the identification of the authorized user, Hu et al. [71] prevent illegal access. To create session keys between a fog node and a user, a Diffie-Helmen key agreement method is utilized. The data is encrypted using a cutting-edge encryption protocol. Implementing the Secure Hash Algorithm allows for integrity checks. For IoT devices with limited resources, Wazid et al. [72] construct a lightweight one-way cryptographic function and bitwise XOR operation to ensure safe key management and user authentication. For resource-intensive devices and fog servers, elliptic curve point multiplication and a biometrics fuzzy extractor approach are utilized.

E. Application Specific Algorithms

When identifying cancers, Xu et al. [73] employ FC using a modified, semi-supervised Fuzzy C Means method. By utilizing improvised particle swarm optimization, Wan et al. work's [74] schedules the task on the equipment using FC in the context of a smart industrial environment. The fog was utilized by Vijayakumar et al. [75] to identify infections caused by mosquitoes early. Users are divided into classes of infected and uninfected users using the similarity coefficient. A random projection and structural similarity index-based light background removal technique for motion detection is proposed by Siddharth and Aghila [76].

VI. Research Challenges in Fog Computing

The discipline of FC has developed from CC as a means of providing clients with computing resources on an economic and product level. IoT device development has recently followed trends that result in low hardware costs. The calculation was carried out quite close to the edge, which eventually lowers the costs associated with computations and data offloading in the cloud and offers security and privacy options for the data at the user's end. Nevertheless, there are several network, security, device, and integration of fog with IoT difficulties in computing at the edge that are now being researched. The dispersed context in which FC operates needs several considerations. This section is a brief explanation of the difficulties in solving issues for fog as shown in Figure 5.

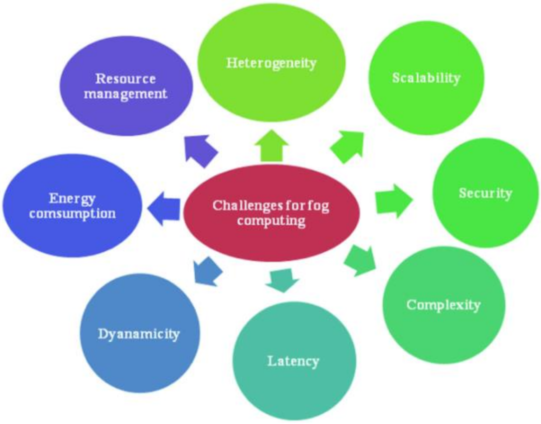

Figure 5: Challenges in FC

A. Device and Network Issues

Decentralized framework: The decentralized nature of the FC architecture results in a redundant structure. On the network's edge devices, the exact same code is repeated [77-78]. The fog environment should thus concentrate on minimizing redundancy in the decentralized design.

Networking resources: The network resources are dispersed at random in the fog architecture. This makes connecting more challenging. It is possible to manage a suitable network that contains middleware to keep a common set of resources at the edge available for the needed application.

Device heterogeneity: The heterogeneous end devices in the fog architecture. The structure is now more diverse due to the nature of heterogeneity [39]. This characteristic of the heterogeneity at the device and network should also be considered by the apps that are created employing fog.

B. Computational Challenges

The following aspects of computations at various levels and how to divide computing resources are examined, making computation at the fog level rather difficult.

Computations at different levels: Cloud servers should always communicate with the fog system. The major goal of the fog system should be to reply to users at the lower level within a certain time frame and to send the necessary calculations to the cloud, which will take more time. When compared to other portions of the calculation, which will be done close to the edge with minimal computing costs, very few computations are offloaded to the clouds since they are not limited by reaction time or processing capacity. It can be difficult to determine which computations should be carried out in the cloud and which should be performed in the fog.

Distribute computation resources: The necessary resources might not be available for computation at the edge, other FNs are a source of these materials. Due to this demand, it is now more important than ever to distribute resource calculations among several FNs. This calls for a strategy that creates a common pool by combining memory, computations, and networking resources. Numerous applications can access the pool and reserve resources based on their needs [79-80]. Instead of employing edge computing devices, the current study emphasizes the necessity to create a shared pool that contains resources.

Mobilization challenge: In terms of mobilization, Open Fog is designated as an N-level environment. However, the sharp increase in the number of fog level levels may result in long-term problems with the recently emerging fog paradigm. As a result, the case study's level count must remain unwavering. Results of mobilization will be approved if they are based on criteria such as the tasks performed at each level, the total number of sensors employed, the proficiency of fog devices, and the dependability and latency of fog devices. However, it is crucial to consider how these requests will be complied with.

Utilization of resource challenge: Because the gadgets are varied in type and readily available, resource utilization is at its most colorful and unique in an environment like a fog. Each of these fog devices is responsible for carrying out the application independently.

Fog device breakdown challenge: Because the fog devices are dispersed and their control is decentralized, the likelihood that they may malfunction is always increasing. Therefore, the fog devices may malfunction for several reasons, including end user behaviour, hardware failure, and software crashing. Aside from these problems, a few more that could be quite important include the power source, connection, and adaptability. Given that many fog devices may be connected by Wi-Fi, it seems sense that Wi-Fi connectivity is occasionally unreliable.

The bulk of them are wirelessly linked mobile devices, thus they could often switch locations to other clusters.

Complexity challenge: FC uses a variety of IoT sensors and devices that have been produced by several firms, making it challenging to choose the best mechanism due to varied hardware configurations, software setups, and individual needs. Additionally, in a few cases, the requirements for maximum security application entail the use of specified protocols and devices, which increases the process's drawback.

C. Security challenges

The heterogeneous devices that make up fog architecture are numerous. They could be exposed to various assaults. Discuss the man-in-the-middle assault that occurred in the fog in [81]. Data and network are the two key problems that need to be addressed in the fog environment, other elements in the cloud datacenters also have a role. The fog's equipment is set up in a less-than-secure area, making it simple to carry out any kind of physical attack. The equipment at the edge must be operated safely in the fog.

VII. Conclusion

The typical IoT-CC architecture has difficulties due to high response latency, increased bandwidth consumption, and large storage needs as data generation rates and volumes rise. By putting processing, storage, and networking closer to end users, FC provides improved services to them. With the requirement for real-time bandwidth-efficient, latency-sensitive, and secure services for end IoT devices with limited resources, the value of FC has become clear. The IoT has currently captured the attention of both industry and researchers. This attraction has changed how we live and is now necessary in today's society. It may link to everything in our environment. IoT devices are powerful yet have limited processing and storage power. However, there are several issues with the conventional integrated CC, including network failure and increased latency. FC has emerged as a solution to these issues. It is an extension of CC, but close to IoT devices. Complete information computation will be done at FNs, reducing waiting times, especially for critical applications. FN and IoT together provide significant benefits for various IoT applications. In this article, we provide several computing paradigms, FC features, an extensive architecture of FC with its multiple levels, and a thorough study of fog with IoT. The focus of the conversation was on various fog system methods and difficulties in the field of FC. In general, the goal of this research project was to offer a study to examine recent contributions in research on FC and IoT in the modern world as well as to indicate forthcoming study and open issues surrounding merging fog with IoT.

VIII. References

[1] Aazam M, Zeadally S, Harras KA. Offloading in fog computing for IoT: Review, enabling technologies, and research opportunities. Futur Gener Comput Syst. 2018.

[2] Haouari F, Faraj R, Alja'Am JM. Fog Computing Potentials, Applications, and Challenges. In: 2018 International Conference on Computer and Applications, ICCA 2018. 2018.

[3] K. K. Vaigandla, "Communication Technologies and Challenges on 6G Networks for the Internet: Internet of Things (IoT) Based Analysis," 2022 2nd International Conference on Innovative Practices in Technology and Management (ICIPTM), 2022, pp. 27-31, doi: 10.1109/ICIPTM54933.2022.9753990.

[4] Dr. Nookala Venu, Dr. A. Arun Kumar and Karthik Kumar Vaigandla. Review of Internet of Things (IoT) for Future Generation Wireless Communications. International Journal for Modern Trends in Science and Technology 2022, 8(03), pp. 01-08. https://doi.org/10.46501/IJMTST0803001

[5] Karthik Kumar Vaigandla , Radha Krishna Karne , Allanki Sanyasi Rao, " A Study on IoT Technologies, Standards and Protocols", IBM RD's Journal of Management & Research, Volume 10, Issue 2, September 2021, Print ISSN : 2277-7830, Online ISSN: 2348- 5922, DOI: 10.17697/ibmrd/2021/v10i2/166798.

[6] KarthikKumar Vaigandla, Nilofar Azmi, RadhaKrishna Karne,

"Investigation on Intrusion Detection Systems (IDSs) in IoT," International Journal of Emerging Trends in Engineering Research, Volume 10. No.3, March 2022, https://doi.org/10.30534/ijeter/2022/041032022

[7] Dr. Nookala Venu, Dr. A. ArunKumar, Karthik Kumar Vaigandla, "Investigation on Internet of Things(IoT) : Technologies, Challenges and Applications in Healthcare," International Journal of Research, Volume XI, Issue II, February/2022, pp.143-153

[8] Karthik Kumar Vaigandla and Dr.N.Venu, "A Survey on Future Generation Wireless Communications - 5G : Multiple Access Techniques, Physical Layer Security, Beamforming Approach", Journal of Information and Computational Science, Volume 11 Issue 9,2021, pp. 449-474.

[9] Karthik Kumar Vaigandla, SandyaRani Bolla , RadhaKrishna Karne, "A Survey on Future Generation Wireless Communications-6G: Requirements, Technologies, Challenges and Applications", International Journal of Advanced Trends in Computer Science and Engineering, Volume 10, No.5, September - October 2021, pp.3067-3076, https://doi.org/10.30534/ijatcse/2021/211052021

[10] Karthik Kumar Vaigandla, Nilofar Azmi, Podila Ramya, Radhakrishna Karne, "A Survey On Wireless Communications : 6g And 7g," International Journal Of Science, Technology & Management, Vol. 2 No. 6 (2021): November 2021, pp. 2018-2025. https://doi.org/10.46729/ijstm.v2i6.379

[11] M.D. Assuncao, R.N. Calheiros, S. Bianchi, M.A.S. Netto, R. Buyya, Big data computing and clouds: Trends and future directions, J. Parallel Distrib. Comput. 79–80 (2015) 3–15.

[12] J. Chen, et al., Big data challenge: A data management perspective, Front. Comput. Sci. 7 (2) (2013) 157–164.

[13] F. Alhaddadin, W. Liu, J.A. Gutierrez, A user prole-aware policy-based management framework for greening the cloud, in: Proc. IEEE 4th Int. Conf. Big Data Cloud Comput. (BdCloud), 2014, pp. 682–687.

[14] A.V. Dastjerdi, H. Gupta, R.N. Calheiros, S.K. Ghosh, R. Buyya, Fog computing: Principles, architectures, and applications, in: Internet of Things: Principle.

[15] R. Mahmud, R. Kotagiri, R. Buyya, Fog computing: A taxonomy, survey and future directions, in: Internet of Everything, Springer, Singapore, 2018, pp. 103–130.

[16] L. Gao, T.H. Luan, S. Yu, W. Zhou, B. Liu, FogRoute: DTN-based data dissemination model in fog computing, IEEE Internet

Things J. 4 (1) (2017) 225–235.

[17] M. Weiner, M. Jorgovanovic, A. Sahai, and B. Nikoli´c, “Design of a Low-Latency, High-Reliability Wireless Communication System for Control Applications”, 2014 IEEE International Conference on Communications (ICC), pp. 3829–3835. IEEE (2014).

[18] R. Kelly, “Internet of Things Data to Top 1.6 Zettabytes by 2022” <https://campustechnology.com/articles/2015/04/15/internet-of-things-data-to-top-1-6-zettabytes-by-2020.aspx> [Available: April 7, 2016].

[19] N. Cochrane, “US smart grid to generate 1000 petabytes of data a year” <http://www.itnews.com.au/news/us-smart-grid-to-generate-1000-petabytes-of-data-a-year-170290#ixzz458VaITi6> [Published: March 23, 2010] [Available: April 7, 2016].

[20] G. Gan, Z. Lu, and J. Jiang, “Internet of Things Security Analysis”, 2011 International Conference on Internet Technology and Applications (iTAP), August 16-18, 2011.

[21] U.S. Department of Transportation, Bureau of Transportation Statistics. <http://www.rita.dot.gov/bts/sites/rita.dot.gov.bts/files/publications/national_transportation_statistics/html/table_01_26.html_mfd> [Available: March 2, 2016].

[22] F. Bonomi, R. Milito, J. Zhu, S. Addepalli, Fog computing and its role in the internet of things, in: Proceedings of the 2012 ACM First Edition of the MCC Workshop on Mobile Cloud Computing, ACM, 2012, pp. 13–16.

[23] M. Whaiduzzaman, A. Naveed, A. Gani, MobiCoRE: Mobile device based cloudlet resource enhancement for optimal task response, IEEE Trans. Serv. Comput. (2016).

[24] Y. Chen, Y. Chen, Q. Cao, X. Yang, PacketCloud: A cloudletbased open platform for in-network services, IEEE Trans. Parallel Distrib. Syst. 27 (4) (2016) 1146–1159.

[25] Y. Shi, G. Ding, H. Wang, H.E. Roman, S. Lu, The fog computing service for healthcare, in: Proceedings of the 2015 2nd International Symposium on Future Information and Communication Technologies for Ubiquitous HealthCare (Ubi-HealthTech), Beijing, China, 28–30 May 2015, pp. 1–5.

[26] M. Taneja, A. Davy, Resource aware placement of data analytics platform in fog computing, Procedia Comput. Sci. 97 (2016) 153–156.

[27] M. Aazam, E.-N. Huh, Dynamic resource provisioning through fog micro datacenter, in: Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops (PerCom Workshops), St. Louis, MO, USA, 2015, pp. 105–110.M.

[28] N.K. Giang, M. Blackstock, R.

Lea, V.C.M. Leung, Developing IoT applications in the fog: A distributed dataflow approach, in: Proc. 5th Int. Conf. Internet Things (IOT), Oct. 2015, p. 155162.

[29] K. Intharawijitr, K. Iida, H. Koga, Analysis of fog model considering computing and communication latency in 5G cellular networks, in: Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops (PerCom Workshops), Mar. 2016, pp. 1–4.

[30] G. Albeanu, F. Popentiu-Vladicescu, A reliable e-learning architecture based on fog-computing and smart devices, in: Proc. Int. Sci. Conf. eLearn. Softw. Edu. Vol. 4, 2014, p. 9.

[31] E. Baccarelli, P.G.V. Naranjo, M. Scarpiniti, M. Shojafar, J.H. Abawajy, Fog of everything: Energy-efcient networked computing architectures, research challenges, and a case study, IEEE Access 5 (2017) 9882–9910.

[32] T.H. Luan, L. Gao, Z. Li, Y. Xiang, G. Wei, L. Sun, Fog computing: Focusing on mobile users at the edge, 2016, pp. 1–11, [Online]. Available: https://arxiv.org/abs/1502.01815.

[33] S. Sarkar, S. Misra, Theoretical modelling of fog computing: A green computing paradigm to support IoT applications, IET Netw. 5 (2) (2016) 23–29.

[34] C. Mouradian, D. Naboulsi, S. Yangui, R.H. Glitho, M.J. Morrow, P.A. Polakos, A comprehensive survey on fog computing: State-of-the art and research challenges, IEEE Commun. Surv. Tutor. 20 (1) (2018) 416–464.

[35] O. Skarlat, S. Schulte, M. Borkowski, P. Leitner, Resource provisioning for IoT services in the fog, in: Proceedings of the 2016 IEEE 9th International Conference on Service-Oriented Computing and Applications, SOCA 2016, Macau, China, 4–6 November 2016, pp. 32–39.

[36] N. Peter, FOG computing and its real time applications, Int. J. Emerg. Technol. Adv. Eng. 5 (2015) 266–269.

[37] M. Chiang, T. Zhang, Fog and IoT: An overview of research opportunities, IEEE Internet Things J. 3 (2016) 854–864.

[38] M. Suarez-Albela, T.M. Fernandez-Carames, P. Fraga-Lamas, L. Castedo, A practical evaluation of a high-security energy-efficient gateway for IoT fog computing applications, Sensors 17 (2017) 1–39.

[39] F. Bonomi, R. Milito, P. Natarajan, J. Zhu, Fog computing: A platform for internet of things and analytics, in: Big Data and Internet of Things: A Roadmap for Smart Environments; Studies in Computational Intelligence, Vol. 546, Springer, Cham, Switzerland, 2014, pp. 169–186.

[40] Definition of fog computing, 2018, Available online: https://www.open

fogconsortium.org/#definition-of-fogcomputing (accessed on 24 March 2018).

[41] K. Hong, D. Lillethun, U. Ramachandran, B. Ottenwalder, B. Koldehofe, Mobile fog: A programming model for large–scale applications on the internet of things, in: Proceedings of the Second ACM SIGCOMM Workshop on Mobile Cloud Computing, Hong Kong, China, 16 August 2003, p. 15.

[42] A. Yousefpour, G. Ishigaki, J.P. Jue, Fog computing: Towards minimizing delay in the internet of things, in: Proceedings of the 2017 IEEE 1st International Conference on Edge Computing, Honolulu, HI, USA, 25–30 June 2017, pp. 17–24.

[43] R. Mahmud, R. Kotagiri, R. Buyya, Fog computing: A taxonomy, survey and future directions, in: Internet of Everything: Internet of Things (Technology, Communications and Computing), Springer, Singapore, 2016, pp. 103–130.

[44] C. Mouradian, D. Naboulsi, S. Yangui, R. H. Glitho, M. J. Morrow, and A. Polakos, "A Comprehensive Survey on Fog Computing: State-of-the-art and Research Challenges," IEEE Communications Surveys & Tutorials, vol. 20, no. 1, 2017, pp. 1–49.

[45] A. Thakur, M. Reza, "Fog Computing for Detecting Vehicular Congestion, an Internet of Vehicles Based Approach: A Review", IEEE Intelligent Transportation Systems Magazine, Vol.11, no.2, pp.8-16, 2019.

[46] S. F. Abedin, M. G. R. Alam, N. H. Tran, and C. S. Hong, "A Fog based system model for cooperative IoT node pairing using matching theory," Network Operations and Management Symposium (APNOMS), 17th Asia-Pacific, 2015, pp. 309–314.

[47] J. Oueis, E. C. Strinati, S. Sardellitti, and S. Barbarossa, "Small Cell Clustering for Efficient Distributed Fog Computing: A Multi- User Case," Vehicular Technology Conference (VTC Fall), IEEE 82nd, 2015, pp. 1–5.

[48] T. Nishio, R. Shinkuma, T. Takahashi, and N. B. Mandayam, "Service-oriented heterogeneous resource sharing for optimizing service latency in mobile cloud," Proceedings of the first international workshop on Mobile cloud computing & networking, MobileCloud '13, Bangalore, India, 2013, pp. 19-26.

[49] H. Wu, "Multi-objective decision-making for mobile cloud offloading: A survey," IEEE Access, vol. 6, pp. 3962–3976, Jan. 2018, https://doi.org/10.1109/ACCESS.2018.2791504

[50] X. Meng, W. Wang, and Z. Zhang, "Delay-constrained hybrid computation offloading with cloud

and fog computing," IEEE Access, vol. 5, pp. 21355–21367, Sep. 2017, https://doi.org/10.1109/ACCESS.2017.2748140

[51] D. Rahbari and M. Nickray, "Task offloading in mobile fog computing by classification and regression tree," Peer-to-Peer Networking and Applications, vol. 13, pp. 1–19, Feb. 2019. https://doi.org/10.1007/s12083-019-00721-7

[52] O. Skarlat, M. Nardelli, S. Schulte, M. Borkowski, and P. Leitner, "Optimized IoT service placement in the fog," Service Oriented Computing and Applications, vol. 11, no. 4, pp. 427–443, Oct. 2017. https://doi.org/10.1007/s11761-017-0219-8

[53] R. Mahmud, S. N. Srirama, K. Ramamohanarao, and R. Buyya, "Quality of experience (QoE)-aware placement of applications in fog computing environments," Journal of Parallel and Distributed Computing, vol. 132, pp. 190–203, Oct. 2019. https://doi.org/10.1016/j.jpdc.2018.03.004

[54] Y. Xia, X. Etchevers, L. Letondeur, T. Coupaye, and F. Desprez, "Combining hardware nodes and software components ordering-based heuristics for optimizing the placement of distributed IoT applications in the fog," in Proceedings of the 33rd Annual ACM Symposium on Applied Computing, Apr. 2018, pp. 751–760. https://doi.org/10.1145/3167132.3167215

[55] Y. Jiao, P. Wang, D. Niyato, and K. Suankaewmanee, "Auction mechanisms in cloud/fog computing resource allocation for public blockchain networks," IEEE Transactions on Parallel and Distributed Systems, vol. 30, no. 9, pp. 1975–1989, Sep. 2019. https://doi.org/10.1109/TPDS.2019.2900238

[56] B. Jia, H. Hu, Y. Zeng, T. Xu, and Y. Yang, "Double-matching resource allocation strategy in fog computing networks based on cost efficiency," Journal of Communications and Networks, vol. 20, no. 3, pp. 237–246, June 2018. https://doi.org/10.1109/JCN.2018.000036

[57] Vaigandla, K. K., Thatipamula, S. & Karne, R. K. (2022). Investigation on Unmanned Aerial Vehicle (UAV): An Overview. IRO Journal on Sustainable Wireless Systems, 4(3), 130-148. doi:10.36548/jsws.2022.3.001

[58] S. F. Abedin, M. G. R. Alam, S. A. Kazmi, N. H. Tran, D. Niyato, and C. S. Hong, "Resource allocation for ultra-reliable and enhanced mobile broadband IoT applications in fog network," IEEE Transactions on Communications, vol. 67, no. 1, pp. 489–502, Jan. 2018. https://doi.org/10.1109/TCOMM.2018.2870888

[59] D. Tychalas and H. Karatza, “A scheduling algorithm for a fog computing system with bag-of-tasks jobs: Simulation and performance evaluation,” Simulation Modelling Practice and Theory, vol. 98, Art no. 101982, Jan. 2020. https://doi.org/10.1016/j.simpat.2019.101982

[60] T. Aladwani, “Scheduling IoT healthcare tasks in fog computing based on their importance,” Procedia Computer Science, vol. 163, pp. 560–569, 2019. https://doi.org/10.1016/j.procs.2019.12.138

[61] D. Zeng, L. Gu, S. Guo, Z. Cheng, and S. Yu, “Joint optimization of task scheduling and image placement in fog computing supported software defined embedded system,” IEEE Transactions on Computers, vol. 65, no. 12, pp. 3702–3712, Dec. 2016. https://doi.org/10.1109/TC.2016.2536019

[62] M. I. Naas, P. R. Parvedy, J. Boukhobza, and L. Lemarchand, “iFogStor: an IoT data placement strategy for fog infrastructure,” in 2017 IEEE 1st International Conference on Fog and Edge Computing (ICFEC), Madrid, Spain, Aug. 2017, pp. 97–104. https://doi.org/10.1109/ICFEC.2017.15

[63] M. I. Naas, L. Lemarchand, J. Boukhobza, and P. Raipin, “A graph partitioning-based heuristic for runtime IoT data placement strategies in a fog infrastructure,” in Proceedings of the 33rd Annual ACM Symposium on Applied Computing, Apr. 2018, pp. 767–774. https://doi.org/10.1145/3167132.3167217

[64] T. Huang, W. Lin, Y. Li, L. He, and S. Peng, “A latency-aware multiple data replicas placement strategy for fog computing,” Journal of Signal Processing Systems, vol. 91, no. 10, pp. 1191–1204, Feb. 2019. https://doi.org/10.1007/s11265-019-1444-5

[65] N. Wang and J. Wu, “Latency minimization through optimal data placement in fog networks,” Fog Computing: Theory and Practice, pp. 269–291, Apr. 2020. https://doi.org/10.1002/9781119551713.ch10

[66] M. A. Hassan, M. Xiao, Q. Wei, and S. Chen, “Help your mobile applications with fog computing,” in 2015 12th Annual IEEE International Conference on Sensing, Communication, and Networking –Workshop (SECON Workshops). Seattle, WA, USA, June 2015, pp. 1–6. https://doi.org/10.1109/SECONW.2015.7328146

[67] A. Gómez-Cárdenas, X. Masip-Bruin, E. Marín-Tordera, S. Kahvazadeh, and J. Garcia, “A hash-based naming strategy for the fog-to-cloud computing paradigm,” in European

Conference on Parallel Processing Workshops. Lecture Notes in Computer Science. vol 10659, Springer, Cham, pp. 316–324, 2017. https://doi.org/10.1007/978-3-319-75178-8_26

[68] D. Guibert, J. Wu, S. He, M. Wang, and J. Li, "CC-fog: Toward content-centric fog networks for E-health," in 2017 IEEE 19th International Conference on e-Health Networking, Applications and Services (Healthcom), Dalian, China, Oct. 2017, pp. 1–5. https://doi.org/10.1109/HealthCom.2017.8210830

[69] A. J. Kadhim and S. A. H. Seno, "Energy-efficient multicast routing protocol based on SDN and fog computing for vehicular networks," Ad Hoc Networks, vol. 84, pp. 68–81, Mar. 2019. https://doi.org/10.1016/j.adhoc.2018.09.018

[70] A. P. Abidoye and B. Kabaso, "Energy-efficient hierarchical routing in wireless sensor networks based on fog computing," EURASIP Journal on Wireless Communications and Networking, Art no. 8(2021), pp. 1–26, Jan. 2021. https://doi.org/10.1186/s13638-020-01835-w

[71] P. Hu, H. Ning, T. Qiu, H. Song, Y. Wang, and X. Yao, "Security and privacy preservation scheme of face identification and resolution framework using fog computing in internet of things," IEEE Internet of Things Journal, vol. 4, no. 5, pp. 1143–1155, Oct. 2017. https://doi.org/10.1109/JIOT.2017.2659783

[72] M. Wazid, A. K. Das, N. Kumar, and A. V. Vasilakos, "Design of secure key management and user authentication scheme for fog computing services," Future Generation Computer Systems, vol. 91, pp. 475–492, Feb. 2019. https://doi.org/10.1016/j.future.2018.09.017

[73] J. Xu, H. Liu, W. Shao, and K. Deng, "Quantitative 3-D shape features based tumor identification in the fog computing architecture," Journal of Ambient Intelligence and Humanized Computing, vol. 10, no. 8, pp. 2987–2997, Feb. 2019. https://doi.org/10.1007/s12652-018-0695-5

[74] J. Wan, B. Chen, S. Wang, M. Xia, D. Li, and C. Liu, "Fog computing for energy-aware load balancing and scheduling in smart factory," IEEE Transactions on Industrial Informatics, vol. 14, no. 10, pp. 4548–4556, Oct. 2018. https://doi.org/10.1109/TII.2018.2818932

[75] V. Vijayakumar, D. Malathi, V. Subramaniyaswamy, P. Saravanan, and R. Logesh, "Fog computing-based intelligent healthcare system for the detection and prevention of mosquito-borne diseases," Computers in Human Behavior, vol. 100, pp. 275–285, Nov. 2019.

https://doi.org/10.1016/j.chb.2018.12.009

[76] R. Siddharth and G. Aghila, “A light weight background subtraction algorithm for motion detection in fog computing,” IEEE Letters of the Computer Society, vol. 3, no. 1, pp. 17–20, 2020. https://doi.org/10.1109/LOCS.2020.2974703

[77] B. Tang, Z. Chen, G. Hefferman, T. Wei, H. He, Q. Yang, A hierarchical distributed fog computing architecture for big data analysis in smart cities, in: Proceedings of the 2015 ACM ASE BigData &SocialInformatics, ACM, 2015, p. 28.

[78] M. Chiang, S. Ha, I. Chih-Lin, F. Risso, T. Zhang, Clarifying fog computing and networking: 10 questions and answers, IEEE Commun. Mag. 55 (4) (2017) 18–20.

[79] P. Bellavista, A. Zanni, Feasibility of fog computing deployment based on docker containerization over raspberrypi, in: Proceedings of the 2017 ACM 18th International Conference on Distributed Computing and Networking, ACM, 2017, p. 16.

[80] M. Yannuzzi, F. van Lingen, A. Jain, O.L. Parellada, M.M. Flores, D. Carrera, J.L. Perez, D. Montero, P. Chacin, A. Corsaro, et al., A new era for cities with fog computing, IEEE Internet Comput. 21 (2) (2017) 54–67.

[81] I. Stojmenovic, S. Wen, X. Huang, H. Luan, An overview of fog computing and its security issues, Concurr. Comput.: Pract. Exper. (2015).