Application of IoT-Fog based real-time monitoring system for open-cast mines—A survey

Devendra Kumar Yadav\(^1\) | Pragyan Mishra\(^2\) | Singam Jayanthu\(^1\) | Santos Kumar Das\(^3\) | Sanjay Kumar Sharma\(^4\)

\(^1\)Department of Mining Engineering, NIT, Rourkela, Odisha, India
\(^2\)Accenture, Hyderabad, India
\(^3\)Department of Electronics and Communication Engineering, NIT, Rourkela, Odisha, India
\(^4\)Department of Mining Engineering, IIT BHU, Varanasi, India

**Correspondence**
Devendra Kumar Yadav, Department of Mining Engineering, NIT, Rourkela, Odisha, India.
Email: devenya2091@gmail.com

**Abstract**
Fog computing (FC) archetypes is in focus recently for its potential to utilise resources optimally by the Internet of Things (IoT) network bearing ample end-devices. This enhances the quality of service (QoS) to the proximity of end-users as latency-free processing is obtained in cloud-IoT-based environs. Mining industry will encounter a level-up due to these technological advancements like fog-IoT-based systems. Monitoring mining activities such as blasting, slope monitoring, and miner tracking are time-sensitive. Hence, instant response is demand of extreme mine environment so that life and property are not compromised. New opportunities emerge in mining since the FC approach addresses these demands depending on the demand of the situation/user. The parameters of integrating the state-of-art of fog, IoT, and cloud architectures are provided, and a Fog-IoT mines monitoring (FiIoTMM) system for real-time monitoring of open-cast mines is proposed. This work also envisions the applications and future trends of FC concerning IoT and cloud. It emphasises how key features of fog help the system in achieving lower network influx, reduced latency, resource utilisation, and immediate data computation and storage because these services take place closer to point of data generation. This data can be immediately visualised/accessed and used to generate early warning in the mines.

**1 | INTRODUCTION**

During the last decade, there has been a momentous progression in the field of computing. Cloud computing (CC) has been an astounding area of research item until a staggering advancement of smart things/devices called the Internet of Things (IoT) indicated the bottleneck of such a centralised archetype. The IoT has given boundless aspects of research drawing attention towards the decentralised archetypes. Every second around 130 ‘things’ get added to the IoT. In 2019, the number of devices connected to IoT is roughly estimated to be around 27 billion. It is very interesting to note how much storage, computations, and real-time data analytics response will be able to entertain such oodles of things in the internet. Eventually, dispersed paradigms such as fog computing (FC) and edge computing (EC) found their way with a perception of furnishing the potential of cloud at the edge of network through overcoming most of the challenges faced by CC.

In 2014, Cisco coined the term ‘fogging’ or FC. The concept of cloud and FC are interlinked. It follows the nature, as in, fog is in close proximity to the Earth than clouds. Technically also, fog lies nearer to end-nodes, delivering down the cloud functionalities to the end-nodes [1]. Fog extends the cloud and has many edge nodes connecting directly to the end-nodes; thereby, it brings CC functionalities to the edge/end devices. In the edge connectivity systems, FC has the topmost expansion of the principles of EC. In fact, FC focuses on portraying a thorough configuration for both vertical and horizontal allocation of resources following the cloud-things continuity [2]. Approximately both FC and EC are similar with a small distinction based on the positioning of the intelligence and processing power. While fog is mainly domain-specific, edge mainly operates for cellular/mobile devices. Both the intermediate technologies, EC and FC, push the processing power and the decision-making close to the IoT zone from where data originates.
With the exponential growth of IoT devices, it is very difficult to maintain the variety, volume, and velocity of data as big data is generated from the IoT devices. When such huge amount of data is sent to the cloud directly, it results in a massive network influx. If the FC layer is introduced between the IoT and the cloud, data processing becomes faster, storage becomes handy, and the resultant network influx decreases. Not all data is important to be sent to the network backhaul. Much of the data can be handled and computed by fog nodes, which improve the strength and performance of cloud architecture. Fog provides an empowerment to the cloud services but not a replacement. FC enhances the execution of exhaustive computations aid for the remote nodes as a portion of the data processing happens near the area where the results have to be conveyed to the user nodes. In this research work, a survey is made exploring a significant amount of research work in FC and other related technologies and frameworks involving cloud and IoT. Thereafter, the architecture, features, applications, and challenges of FC are discussed. From all these data, a novel technique has been proposed to monitor the environment of open-cast mines in real-time with the integration of fog with IoT, cloud, and sensors. A general overview of this unification is shown in Figure 1 depicting the basic elements. The three ingredients are (a) mines sensor network, (b) fog layer and (c) cloud layer. Factors such as ground vibration due to blasting, slope failure due to its movements, lorry positioning and mines personnel positioning could be monitored. Data produced from these factors are sensed by the sensors, which are deployed in this area and sent to the fog; then, the refined data is sent to the cloud layer data centre for permanent storage.

This proposal is detailed later in this manuscript. To sum-up, it can be speculated that addition of fog upgrades the architecture of real-time monitoring systems, which do not prefer delay in data transmission. The following are the main objectives of this survey article:

- Introducing the recent research directions by exploring several existing research works in FC.
- Enabling a feasible proposal of fog-IoT-based real-time connectivity and monitoring in open-cast mine.
- Providing latency-free services to monitor open-cast mines and ensure miner as well as equipment safety.

The rest of the research article is organised in the following sequence: Section 2 gives the motivation behind this research. Section 3 gives the background of the components used in Fog-IoT Mines Monitoring (FIoTMM) system, describes how FC aids IoT and CC and outlines the characteristics and architecture of FC. Section 4 shows the related research works. Section 5 compares the feature CC, FC, and EC. Sections 6 and 7 list the applications and concerns of FC respectively. Section 8 describes the proposed system. Section 9 lists the service latency analysis and performance evaluation of FC scenario as compared to CC. Section 10 concludes the paper with a future outlook. Thereafter, mathematical modelling is presented in Appendix section.

2 | MOTIVATION

Fogging is an emerging concept having great potential to implement energy-efficient strategies. The primary motivation behind this work is the need of latency reduction, fast processing of data in IoT applications, and reduced burden on cloud. In the existing mine monitoring system/applications, the fog layer is absent, which means that every time the sensed data has to be sent to the cloud, the monitoring system needs to wait for visualising the response until cloud has processed the data. With the integration of fog in the system composed of IoT and cloud, there will be hike in the overall system behaviour. The real-time monitoring applications usually involve cloud and IoT where latency is a major problem due to variety, velocity, and high volume of data. Due to huge amount of data, round-trip time delay is caused, which increases from milliseconds to seconds and then from seconds to minutes [3,4]. When it comes to monitoring of environment such as air, heat, rainfall, slope movement in mines, rock displacement, and precipitation. Data is uncertain and fluctuates every moment. Hence, it becomes very important to record every data and understand the exact flow or behaviour of that scenario. Moreover, a comprehensive survey of previous research stuffs has been surmised, and an attempt has been made to accomplish a real-time mine monitoring system. This will give another research perspective of FC in real-time environment monitoring.

3 | BACKGROUND

CC has seen a golden decade from 2005 to 2015. Despite the merits of cloud, CC is facing a shift in paradigm, and thus the concept of fog/EC came into spotlight. With the emergence of FC, the colossal load lifted by the centralised cloud framework has transformed closer to a point where the data is produced, that is, the IoT devices. When processing comes closer to the source of data, the response time becomes acutely low. Thus, this transition from cloud to fog travels also touches other technologies such as cloudlets, fog, and mobile edge computing (MEC) or EC.

Cloud computing: CC utilises core resources of the network. Cloud bears a centralised software and application server with a huge data centre and an inherent local network. Thus, it can maintain and store big data in a local medium that cannot be done by any IoT devices. This big data is not only useful for information but also important to know the trend it follows concerning a particular area which is commonly known as big data analytics. The IoT devices can neither process such a large volume of data nor store the ample computational data in its limited memory capacity. To overcome these shortcomings of IoT, CC provides service by connecting to the IoT. Iot devices generate data and send it to the cloud for processing
and storage, and then, the processed data is sent back to the IoT layer as per the requirement. Due to growing number of devices, several requests to the cloud layer become more, thereby increasing network traffic and delayed data exchange with the IoT layer. Here the cloud functionality minifies and becomes less suitable for time-sensitive applications [5]. Ensuring the handiness of the CC performance, IoT framework faces a lot of difficulties such as data synchronisation, maintenance, standardisation, balancing, and privacy [6].

Internet of things: The term ‘thing’ can be any physical entity in this world, connected to internet, with a unique tag through which it can be identified. These things can sense and gather data of the parameters from smart home, atmosphere, environment, agriculture, military area, health care, transport, aircraft, appliances, and so on. Some of the sensors are: sensor to measure water volume to prevent water leakage in irrigation, sensor to measure water pressure to monitor its flow used for cultivation, pressure sensor to monitor pressure acting on vehicles and to resolve altitude in aircrafts, chemical sensors, biosensors, IR sensors used to observe health parameters, GPS, gyroscope, and proximity sensors used in mobile devices and vehicles. These sensors are not influenced by dirt, water, or oil. If such ample devices every time request cloud, then the network's bandwidth is decreased. Thus, fogging has to be utilised to filter the data sent to the cloud.

Fog computing: This archetype offers connectivity support for low latency, location-based and real-time services enhancing user experience. Maximum things in the IoT network demand time-sensitive, energy-efficient, and real-time aids. The fog nodes which are available as a middleware between cloud and IoT in a distributed fashion fulfill this criterion as they have effective data processing power and resource management as compared to cloud. In this way, the cloud functionalities are brought in close proximity to the network edge where data is generated through fogging. The fog layer wisely filters the IoT data into two categories. One class of data has to be processed by fog nodes and another class of data should be dispatched to the CC layer. Though FC is not a substitute for CC, it acts as an add on to the cloud features. Both CC and FC offer sophisticated calculations and durable storehouse for data. FC utilises spare resources at the network edge to execute tasks.

Cloudlets: While fog is considered as a mini cloud, cloudlets are considered as micro cloud data centres operating in small scale [7]. This also extends cloud services like fog does. While cloud offers limitless resource usage, cloudlets can offer only finite resource usage. Cloudlets are situated at the internet edge. Mainly it supports resource-demanding mobile applications for mobile devices which can reciprocate by offering high power computing resources at a quicker pace [8]. These mobile devices can be tablets, smart wearables, smartphones within a closer geographical area.

Mobile edge computing: This technology also aims at enabling CC features at the network edge to improve QoS. MEC computes at the edge like the micro cloud node, or cloudlets can be an edge node between an IoT device and cloud [9]. MEC chiefly operates at the mobile network base station, and both computation and storage are regulated external to the mobile device. MEC and FC are interchangeable, where MEC addresses more of the things in the IoT layer, whereas FC addresses more of infrastructure sector [10].

3.1 Fog computing to aid internet of things and cloud computing

When it comes to the scenario of (1) managing data, (2) processing data, (3) instant data analytics, and (4) quick resolution out of the evaluated data, the cloud being a centralised system and having powerful computation ability and capacity to handle and supervise a huge data is a perfect choice which can perform these four tasks efficiently and simultaneously. This goes fine when the ‘things’ in the IoT are less or concentrated within a smaller geography. Nonetheless, if the number of IoT applications or ‘things’ increases covering a wider geography, then to fulfil all the four tasks simultaneously is not possible for CC due to its concentrated nature. So, to have all four tasks done at the same time, rather than referring everything to the
cloud, there can be resources lying nearer to these ‘things’ that produce data. These resources can locally process the data collected from the ‘things’, and hence manage data; then at the same time it can analyse the data and send back a prompt response to the ‘things’. Thus, this is a smarter way to utilise the resources and have instant response for the requests. Besides, another perk of this concept will curtail extra burden on the cloud servers because only cloud-compatible requests will be sent up to the cloud. These requests are the ones which are essential from the system’s viewpoint based on the type of application. Apart from aiding the cloud layer, this concept also aids the IoT as not even a single ‘thing’ is left unattended. This concept is nothing but the FC paradigm.

Normally, the IoT applications, which need an outside computing source and storage, have the following four characteristics: (1) are distributed in a vast geography, (2) have a copious number of sensor nodes (SNs) for data transmission, (3) have huge data to be processed, collected, and made available for the next processes waiting to act upon, and (4) possess live data analytics. Each of the above-mentioned features can be efficiently addressed by cloud but not in a concurrent manner. This is a notable challenge of CC. Another concept is that the connotation of the data investigated mostly depends on location when it comes to real-time data analytics. For instance, data generated by an event might be an insignificant universal degree but can be very meaningful for a local or intermediate process. Monitoring applications such as health monitoring, environment monitoring, or traffic monitoring operating in cloud-IoT blend, tend to fall under this category [11,12]. These applications have a lot of processes, and only the refined data should be stored in the cloud server for its permanent impression and ubiquitous availability [13]. Data analytics, predictions, and statistics can be computed upon this cloud data. The intermediate results are not needed to be stored in cloud because if not immediately, subsequently, local analytics and aggregation job must be done for the localised data [14]. Then, why not do the same in some local layer between IoT and cloud. Hence, to back both core layer and edge layer, fog becomes a bridge.

3.2 Characteristics of FC

Besides gluing the cloud and IoT layer, fog addresses various other features described in this section as follows:

Location cognizance: Fog nodes can be deployed anywhere in the fog layer. This technical component gives knowledge about the physical position of anything to another application [15].

Ultra-short latency: As the fog layer is closer to the edge of internet, processing occurs very quickly and hence turn-around-time decreases significantly giving very short latency [16].

Geographical distribution: As compared to the centralised cloud architecture, fog offers a distributed system for its applications thereby reducing congestion in network. The fog nodes are widely distributed over the IoT layer, so that they can accommodate all the data coming from the IoT devices [17].

Extensibility: Due to the distributed nature of FC, the fog nodes are sizable in nature. Thus, it supports implementation in large-scale edge devices. For environment monitoring systems, such large-scale sensor network is mandatory, which is scalable and extensible in nature [18].

Mobility support: Fog can connect directly to the mobile devices which can isolate location identity from host identity by using some protocols such as Locator ID Separation Protocol. This enables mobility schemes [16].

Real-time communication: While cloud offers batch processing, fog provides instantaneous processing in fog nodes [19]. This reduces delay and supports interaction in a timely manner.

Heterogeneity: Due to implicitly virtual structure, FC is adaptable in nature, and it can be deployed in any platform. Therefore, it goes with any type of application [20].

Interoperable: Fog can interoperate among a large scope of environment as it is compatible and reconcilable. This includes functions such as spontaneous streaming and ability for analysing and decision-making.

Privacy and security: When it comes to internet, everyone is concerned about safety of their information as it is uploaded in the cloud platform. As fog is distributed, central management of data is not possible. When fog processes, it is at an intermediate level, and the user does not have full authority over the fog application. Thus, it becomes very important to keep sensitive data securely. Many research works have proposed security solutions for fog services which ensure the reliability on fog network [21].

Resource/task management and scheduling: Management of resources and task is a very primary function of FC. Mainly when heterogeneous devices work together, it becomes essential that idle or spare resources are utilised based on their availability, so that system efficiency is improved. The tasks should also be allocated in such a way that throughput time is minimum, and resources utilisation is maximum while computing a task. If this is not managed properly, fog faces unwanted complicated problems while processing IoT data. This in turn will become an obstacle in lowering latency. Fog resources are not committed, and hence, available resources are efficiently allocated for processing [22].

Fault tolerance: This is the quality of a system to be ongoing even though a portion of the system stops working; it may be a hardware, software, or network breakdown. As a result, the system seems to perform as it is instead of closing fully. Fault tolerance is addressed in CC and should be addressed in FC as well. Many research works have given to consider this concept [23,14].

A very interesting mechanism known as duplication is commonly followed which replicates data from one fog node to another [15]. This serves as a data back-up and thus ensures data integrity.

Multi-occupancy: Multiple users use a service with a secluded run-time for an individual user known as multi-occupancy. The users or devices are referred to as occupants
within the network. This feature is important as count of resources is limited in fog layer. One occurrence of multi-occupancy will serve multiple nodes to operate [15].

Real-time data analytics: Due to the placement of fog between the cloud core and internet edge, fog ingests and processes data closer to the Internet in real-time. This is also done by the real-time analysis [14].

Cognition: Fog framework is conscious of user requirement and thus intelligently manages the tasks and resources. It carries out fine execution of data and provides control, connectivity, and storage facilities as per the cloud fashion and hence improves service quality of the clients and end users [24].

Agility: To start a completely new framework from the scratch is an expensive task. In this context, fog supports the concept of create–respond–change popularly known as the agile development of an application. This service is less expensive and adaptable to any type of application in the IoT layer. Due to agile support, managing development of an application becomes more efficient [25].

3.3 Architecture of fog computing

Fog bridges the gap between the cloud (core) and the IoT edges. The backbone of the cloud architecture are the database centres where large amount of data is stored and processed. The fog network contributes to faster data processing and decreasing response time and round-trip time to the cloud and back to the end devices. As shown in Figure 2, the fog network architecture is formed of three layers, namely (1) the IoT layer, (2) the fog layer, and (3) the cloud layer [26,27]. The following are the detailed working of each layer.

1. IoT layer: This is the bottom layer (also known as sensing or edge layer) which comprises of universally distributed sensor devices. It operates in the physical and data link layer of the network stack model. It consists of sensors as a part of sensing technologies making the base of the IoT layer [28,29]. Sensor devices could be smart home appliances; environment scenarios such as mines, rocks, water bodies; health care devices; smart devices; smart vehicles; fingerprint sensors; and manufacturing factories. The sensed data goes through some local processing before going to the upper layer. This layer collects data from the solely identified objects of the IoT network through the SNs, then converts this data to digital signals and finally transmits them upwards for further processing. The sensed data is communicated from IoT edge to the fog layer through communication mediums like WSN, Wi-Fi, radio-frequency identification (RFID) technology, long range communication technology (LoRa), Bluetooth, cellular technology like 4G/5G, and near-field communications (NFCs).

2. Middleware: This is the fog layer or the middle layer where the countless fog nodes operate for processing abundant data and produce refined evaluated data [30]. The fog nodes act as local servers having intense processing power. Fog layer consists of gateways, firewalls, access points, routers, cloudlets, and switches. The fog information is temporarily stored and transmitted to and fro the two end layers: edge and core, and this data can be easily accessed and visualised anywhere in between these two layers. Therefore, fog connects to the edge to receive data and also connects to cloud to produce and collect data or requests. Fog is also intelligent enough to filter out what data has to be sent to cloud and what data should be locally processed in fog. This reduces cloud overheads and improves all round system efficiency and cost-cutting. For time-sensitive applications, fog instantly computes the data and gives the result to the end user rather than waiting for cloud. This minimises latency. Specifically, for mining applications, where there is risk of losing human lives and equipment, even a little delay in generating early alarm may prove fatal. The importance and justification of introducing FC in mine monitoring application are elaborated in Section 11.

3. Back-end layer: This layer comprises of cloud data centres and network core layer. The cloud layer has numerous high-performance base stations with high storage capacity for permanent data storage. Main units of cloud layer are cloud servers, application servers and data centres. The data fetched from the IoT devices do not get stored directly in cloud server rather. They are refined in the fog layer, and then, that data goes to cloud to be stored permanently and be accessed through internet. The internet or World Wide Web acts as a connectivity between cloud and fog layer.

4 RELATED WORKS BASED ON INTERNET OF THINGS AND FOG

With the growing demand of instant wireless connectivity in day-to-day life, FC is also actualised. Fog originates from the convergence of WSN, IoT, CC, EC and mobile computing [31]. It acts as a powerful tool to harness the resources in closeness to IoT so that both communication with cloud servers and provision of timely processing of data are obtained simultaneously. FC is very powerful tool when it comes to data to be collected at extreme edge like roadways, extreme environmental condition, vehicles, ships, smart cities and so on. Besides, when there is need to segregate data from a large number of devices across larger geographies. Most importantly, for time-sensitive applications which demands analysis and processing of data instantly having both time and resource constraints. Many research activities are going on in this field. Work done in [32] proposes a weight cost model which checks execution time and energy use of IoT applications. This algorithm ensures that tasks are processed concurrently. FC is also applied for privacy preserving data collection. Lu et al. have implemented lightweight data collection scheme which can exclude false data and endures fault tolerance [33]. Baccarelli et al. have combined both cloud and fog frameworks to ensure minimum computational energy during task and resource allocation using genetic algorithm [34]. Fog can accurately recommend contents while
browsing in IoT based environment applying association rules [35]. Besides recommending content, abstraction of useful information can be done by recommender system based on fog [36]. A scientific research work for developing semantic model through guided approach has been proposed by Petrovic et al. for deployment and flexibility of container-based operations in FC [37]. Proposal in Reference [38] describes a model for IoT applications to reduce usage of energy and processing/execution time; Memetic Algorithm is also proposed which decides which IoT application/s to be chosen to run concurrently in fog environment. Information-oriented service model is proposed for fog architecture for being auto-adaptive and operate in both cloud and edge devices [31]. To cope up with exponential growth of IoT data in cloud which is referred to as big data [39], it has become a pre-requisite to have a fog layer in between the IoT devices and cloud [40] in a smart home [41,42], or smart manufacturing system [43], or smart connectivity through gateway [44] or smart grid [45]. Even this big data is ubiquitously available from cloud server through the fog layer. When container is used as a resource in multiple cloud-fog architecture, servicing time of requests decreases [46].
Latency is the most important factor for time-sensitive IoT applications. Fog infrastructure is scalable and hence the fog nodes adjust accordingly, working in synchronization while processing data to reduce turn-around-time [47]. Another human-driven and device-driven intelligence approach applies machine learning and task-offloading algorithm which enables fog to reduce latency [48]. Service-level agreements (SLA) for IoT services can also be monitored using fog thereby bringing energy efficiency and profit to the users [49]. Fogging is also used for task scheduling of offloaded tasks preventing any miss in deadline by managing and predicting resources in such a way that tasks are distributed to under loaded or leisure fog nodes [50–52]. Task scheduling and resource management are two important features addressed by fogging in a better way than any other framework [53]. Resource allocation could also be done via physical fog kit to fulfill the need [54]. For multiter fog systems, offloading of workload is a challenge. Though it reduces latency, during network traffic, it may consume more energy for processing. Thus, systems having features like predictive offloading [55], computation with energy gathering capacity [56], ubiquitous availability [57] are useful for such scenario while allocating IoT resources in a cloud-fog-IoT environment. Vehicular task offloading could also be checked using 802.11p transmission protocol [58] and through fogging for sensing for self-driven vehicles [59]. To charge the electric vehicles, fog incorporates smart grid [45]; here fogging is used to minimize plug-in time in public charging points.

Data privacy and security are also an important feature addressed by fog [21]. In some studies, machine learning is applied to FC addressing security and resource management [60,61]; while some studies propose trust models combining cloud and IoT to enable privacy [62]. Agent-based encryption is implemented to put a check on access using unknown key-issuing protocol preserving user's identity [63].

While comparing cloud with fog platform and without fog platform, the former one proved that cloud with fogging helps increasing energy efficiency and reducing queue length of time averaged applications [64]. Drones are significantly used in different fields to capture data. Fogging is also applied here to compute the collected data locally to provide instant processing for applications like entity recognition, military applications, path planning, etc. [65]. Space applications are also time sensitive ones requiring intense computation and this is taken care by fogging in the form of satellite nodes [66]. Geographically distributed applications need fog integrated IoT system for effective monitoring [67]. Ensuring QoS is effectively maintained by fogging as task scheduling is optimal and hence response time is swift thereby decreasing cost [68]. In health care applications and services operated in IoT network, QoS is enhanced by fogging [69,70]. Besides QoS, Quality of Experience (QoE) is also improved by fog. In Reference [71], application placement policy is suggested which is aware of placing applications appropriate for fogging.

Other miscellaneous applications like gaming also share resources that are unoccupied through fogging [72]. Mainly inexpensive and time-sensitive have a high requirement of fogging. There should be a balance between the fog nodes which carry out inexpensive and minimum latency services and the cloud nodes which bears costly processes. Therefore, load balancing of cloud nodes and fog nodes is crucial so that resource limitation is not compromised [73]. After coming across the literature work, a more detailed survey of research work and the aim of the research is summarized and depicted in Table 1. This table envisions several existing fog-IoT based research works. The main characteristics outlined are based on QoS, cost of development, deployment and maintenance the system, efficiency, security, monitoring systems built on fog, bandwidth, latency, storage space, resource or task management and their allocation, modelling and simulation, healthcare and load balancing. To summarize, most of the research works show efficiency of FC. Moreover, the research works show how introduction of fog layer minimizes latency and hence can be applied to most of the monitoring systems which are time-sensitive. Security concerns are addressed in research works to ensure security of information.

### 5 A VISION OF COMPARISON AMONG FOG COMPUTING, EDGE COMPUTING AND CLOUD COMPUTING

After understanding the factors based on which the concepts of fog, cloud and EC differ, it will give a clear picture of how FC can be a helping hand to most of the cloud-IoT applications. Besides, it should be known how the two similar concepts of FC and EC differ from each other and are different from CC.

**Architecture type:** Both FC and EC have dispersed architecture whereas CC has centralized architecture [13].

**Architecture component:** The physical building blocks of the EC and FC architecture are multiple edge nodes or fog nodes whereas the cloud is composed of data centres [149].

**Server availability:** In cloud, there are limited number of servers; often confined to less data centres but in fog layer several servers are available and are more scalable; whereas in EC, the servers are less scalable and less in number as compared to FC [13].

**Services offered:** As FC is distributed and locally present above IoT zone, it offers services based on specific domain of IoT device/application. EC is confined only to offer service to cellular network and mobile devices [88].

**Amount of data processed:** In contrast to cloud processing big data, fog and edge process small/less amount of data [114].

**Cognizance of location:** CC, being a centralized system, is never aware about the position or location whereas both FC and EC are aware of their location as they have distributed architecture and have a number of nodes [131].

**Mobility:** CC has limited mobility as it is centralised. Both FC and EC support mobility as they can directly connect to the devices at Internet edge. FC is a better choice for this feature as compared to EC [4].

**Real-time connectivity and response:** Real-time connectivity is supported by all the three computing architectures, but only difference lies in ability to respond instantly. FC has the
| Paper                | QoS | Cost | Efficiency | Security | Monitoring | Bandwidth | Latency | Storage | Resource/Task Management/Allocation | Modelling/Simulation | Healthcare | Load Balancing |
|----------------------|-----|------|------------|----------|------------|-----------|---------|---------|-------------------------------------|----------------------|------------|----------------|
| Nguyen et al. [61]   | N   | N    | Y          | Y        | N          | N         | Y       | N       | N                                   | N                    | N          | N              |
| Shivi et al. [12]    | Y   | Y    | Y          | Y        | Y          | Y         | Y       | N       | Y                                   | N                    | N          | N              |
| Dimitrios et al. [74]| Y   | Y    | Y          | N        | N          | N         | Y       | N       | Y                                   | N                    | N          | N              |
| Jinhui et al. [75]   | N   | Y    | Y          | N        | N          | N         | N       | N       | Y                                   | Y                    | N          | N              |
| Ahmed et al. [76]    | N   | Y    | Y          | N        | Y          | Y         | Y       | Y       | Y                                   | N                    | N          | N              |
| Esuabalew et al. [77]| N   | N    | Y          | Y        | N          | Y         | Y       | Y       | Y                                   | N                    | N          | N              |
| Andras et al. [78]   | N   | N    | N          | N        | Y          | N         | N       | N       | Y                                   | N                    | N          | N              |
| Abdurrahman et al. [79]| Y | Y    | Y          | N        | N          | N         | N       | N       | Y                                   | N                    | N          | N              |
| Huiying et al. [22]  | N   | Y    | Y          | N        | N          | Y         | N       | Y       | Y                                   | N                    | N          | N              |
| Ranesh et al. [80]   | N   | Y    | N          | N        | N          | N         | N       | N       | Y                                   | N                    | N          | N              |
| Simar et al. [81]    | N   | N    | Y          | N        | N          | N         | Y       | N       | Y                                   | N                    | Y          | Y              |
| Mahdi et al. [82]    | N   | Y    | Y          | N        | N          | N         | N       | Y       | Y                                   | N                    | Y          | Y              |
| Muzammil et al. [83] | N   | N    | Y          | N        | N          | N         | Y       | Y       | N                                   | N                    | N          | Y              |
| Morteza et al. [84]  | Y   | Y    | N          | Y        | Y          | Y         | Y       | Y       | Y                                   | Y                    | Y          | Y              |
| Anwesha et al. [85]  | N   | N    | Y          | N        | Y          | N         | N       | Y       | Y                                   | Y                    | Y          | N              |
| Shreshth et al. [86] | N   | N    | N          | Y        | N          | N         | Y       | N       | N                                   | Y                    | N          | N              |
| David et al. [87]    | N   | N    | Y          | N        | Y          | N         | Y       | N       | N                                   | Y                    | N          | N              |
| Kwang [88]           | N   | N    | Y          | N        | N          | N         | N       | Y       | Y                                   | N                    | N          | N              |
| Meikang et al. [89]  | N   | N    | N          | Y        | N          | N         | N       | N       | N                                   | N                    | N          | N              |
| Swati et al. [90]    | N   | N    | N          | Y        | Y          | N         | N       | N       | N                                   | N                    | N          | N              |
| Carlo et al. [91]    | N   | N    | N          | N        | N          | Y         | N       | Y       | Y                                   | N                    | N          | N              |
| Chaogang et al. [92] | N   | N    | Y          | N        | N          | Y         | Y       | N       | N                                   | N                    | N          | N              |
| Ewa et al. [93]      | N   | N    | Y          | N        | N          | N         | N       | N       | Y                                   | N                    | N          | N              |
| Vincenzo et al. [94] | Y   | Y    | N          | N        | Y          | Y         | Y       | Y       | Y                                   | Y                    | N          | N              |
| Judy et al. [95]     | Y   | N    | Y          | N        | N          | N         | N       | N       | N                                   | N                    | N          | N              |
| Paper            | QoS | Cost | Efficiency | Security | Monitoring | Bandwidth | Latency | Storage | Resource/Task Management/Modelling/Simulation | Healthcare | Load Balancing |
|------------------|-----|------|------------|----------|------------|-----------|---------|---------|---------------------------------------------|------------|----------------|
| Enzo et al. [96] | N   | N    | Y          | N        | N          | N         | N       | N       | Y                                            | N          | N              |
| Yuanyuan et al. [97] | N   | Y    | N          | Y        | N          | N         | N       | N       | Y                                            | N          | N              |
| Alexandre et al. [98] | N   | Y    | N          | Y        | Y          | N         | Y       | N       | N                                            | N          | N              |
| Bahar et al. [99]  | N   | Y    | N          | N        | N          | N         | Y       | Y       | N                                            | N          | N              |
| Deze et al. [100] | Y   | N    | Y          | N        | N          | N         | N       | Y       | Y                                            | N          | N              |
| Marc et al. [13]  | N   | N    | N          | N        | N          | Y         | N       | N       | Y                                            | N          | N              |
| Saurabh et al. [70] | N   | N    | Y          | Y        | Y          | N         | Y       | Y       | Y                                            | N          | N              |
| Hina et al. [101] | N   | N    | Y          | N        | N          | N         | Y       | N       | N                                            | N          | N              |
| Xiaodong et al. [102] | N   | N    | Y          | Y        | N          | N         | Y       | Y       | N                                            | N          | N              |
| Qinglin et al. [43] | Y   | Y    | N          | Y        | Y          | Y         | Y       | Y       | Y                                            | N          | N              |
| Zeyu et al. [103] | N   | Y    | Y          | N        | N          | N         | N       | N       | Y                                            | N          | N              |
| Roberto et al. [104] | Y   | Y    | N          | Y        | Y          | Y         | N       | N       | N                                            | N          | N              |
| Bo et al. [105]  | N   | Y    | N          | N        | N          | Y         | N       | Y       | Y                                            | N          | N              |
| Zening et al. [106] | N   | Y    | Y          | N        | Y          | N         | Y       | Y       | Y                                            | N          | N              |
| Shaohan et al. [107] | N   | Y    | N          | N        | N          | N         | N       | N       | Y                                            | N          | N              |
| Chen-Feng et al. [18] | N   | Y    | N          | N        | N          | Y         | N       | Y       | Y                                            | N          | N              |
| Jindou et al. [108] | N   | Y    | Y          | N        | N          | Y         | N       | Y       | Y                                            | N          | N              |
| Ammar et al. [109] | N   | N    | Y          | Y        | N          | N         | Y       | Y       | Y                                            | N          | N              |
| Kwasi et al. [110] | N   | N    | Y          | Y        | N          | N         | Y       | N       | N                                            | N          | N              |
| Quping et al. [111] | N   | Y    | Y          | N        | N          | N         | Y       | Y       | Y                                            | N          | N              |
| Yifan et al. [112] | N   | Y    | Y          | N        | N          | N         | Y       | Y       | Y                                            | N          | N              |
| Ahmad et al. [113] | N   | N    | Y          | N        | N          | Y         | Y       | Y       | N                                            | N          | N              |
| Ke et al. [114]  | N   | N    | Y          | N        | N          | N         | N       | N       | N                                            | N          | N              |
| Shi et al. [115]  | N   | N    | N          | N        | Y          | N         | N       | N       | N                                            | N          | N              |

(Continues)
| Paper                  | QoS       | Cost | Efficiency | Security | Monitoring | Bandwidth | Latency | Storage | Resource/Task Management/Allocation | Modelling/Simulation | Healthcare | Load Balancing |
|-----------------------|-----------|------|------------|----------|------------|-----------|---------|---------|-------------------------------------|----------------------|------------|-----------------|
| Shubho et al.         | N         | Y    | Y          | Y        | N          | N         | N       | N       | Y                                   | Y                    | N          | N               |
| Richard et al.        | N         | N    | N          | N        | N          | N         | N       | N       | Y                                   | Y                    | N          | N               |
| Yang et al.           | N         | N    | Y          | N        | Y          | N         | N       | Y       | Y                                   | N                    | N          | N               |
| Surbhi et al.         | N         | N    | N          | N        | N          | N         | N       | N       | Y                                   | N                    | N          | N               |
| Suvajit et al.        | N         | Y    | Y          | N        | N          | Y         | N       | N       | Y                                   | Y                    | N          | N               |
| Behailu et al.        | N         | N    | N          | Y        | N          | N         | N       | N       | Y                                   | N                    | N          | N               |
| Sandeep et al.        | Y         | N    | Y          | Y        | Y          | N         | Y       | Y       | Y                                   | Y                    | N          | N               |
| Shailendra et al.     | N         | N    | Y          | N        | N          | N         | Y       | N       | N                                   | N                    | N          | N               |
| Zhi et al.            | N         | N    | Y          | Y        | N          | N         | N       | Y       | N                                   | N                    | N          | N               |
| Roberto et al.        | N         | N    | Y          | N        | N          | N         | N       | Y       | N                                   | N                    | N          | N               |
| Jianan et al.         | N         | Y    | N          | N        | Y          | Y         | Y       | Y       | N                                   | N                    | N          | N               |
| Boqi et al.           | Y         | Y    | Y          | N        | N          | Y         | N       | Y       | Y                                   | Y                    | N          | N               |
| Lei et al.            | N         | Y    | N          | Y        | N          | N         | N       | Y       | Y                                   | Y                    | N          | N               |
| Amir et al.           | N         | Y    | Y          | Y        | Y          | Y         | Y       | Y       | Y                                   | N                    | N          | N               |
| Aparna et al.         | Y         | Y    | Y          | Y        | Y          | Y         | Y       | Y       | Y                                   | Y                    | N          | Y               |
| Aparna et al.         | N         | N    | N          | N        | N          | N         | N       | N       | N                                   | Y                    | N          | N               |
| Mohammed et al.       | N         | N    | N          | N        | N          | Y         | Y       | N       | N                                   | N                    | N          | N               |
| Te-Chuan et al.       | N         | N    | N          | N        | N          | Y         | N       | N       | Y                                   | N                    | N          | N               |
| Yang et al.           | N         | N    | Y          | N        | N          | N         | N       | Y       | Y                                   | N                    | N          | N               |
| Jihun et al.          | N         | N    | N          | N        | N          | N         | N       | Y       | Y                                   | N                    | N          | N               |
| Ahmed et al.          | N         | N    | N          | Y        | N          | N         | N       | N       | N                                   | N                    | N          | N               |
| Kai et al.            | N         | Y    | Y          | N        | N          | Y         | N       | Y       | Y                                   | Y                    | N          | N               |
| Romana et al.         | N         | N    | N          | Y        | Y          | N         | Y       | N       | N                                   | N                    | N          | N               |
| Vasileios et al.      | N         | N    | N          | N        | N          | Y         | Y       | N       | N                                   | N                    | N          | N               |
| Souza et al.          | Y         | N    | N          | N        | N          | N         | N       | Y       | N                                   | N                    | N          | N               |
highest ability for real-time response to any IoT request while CC has the lowest ability as it suffers latency. Ability to respond spontaneously for EC lies in the middle of these two. Real-time response depends on latency. Higher the latency, lower is the response time and vice-versa [138].

Data storage and persistence: Cloud promises life-long span as it regulates and deals with big data. Both FC and EC offer short-term storage being an intermediate layer and performing intermediate data processing. This depends on the application or area they are operating for CC [137].

Data analytics and quality of computation: Both FC and EC have short-lived or temporary competence to analyse data, both being a middleware; but FC has high power computations whereas EC computes based on priority of the task. CC has permanent capability for data analytics, and computations are carried out based on category [25].

Count of end-users supported: CC is accessed by general internet users as it is accessed through internet. EC has mainly mobile users as end users. End users of FC are mostly users of industrial IoT devices [91].

Area of deployment: CC infrastructure is setup in an area owned by a cloud service provider. As it has huge servers and data centres, it is mainly kept indoors. FC system can be deployed in roadways, open fields, home, buildings, streets, railway tracks, etc. This implies that wherever Internet can be reachable, fog can exist in those areas. EC infrastructure must be positioned by specific service providers in specific areas [4].

Service providers: Tech giants like IBM, Amazon, Microsoft Azure, and Google are cloud service providers. Cellular network operators are service providers for EC. Intel and Cisco IOx are fog service providers [138].

Cost and energy requirement: As cloud bears several data centres, the cost of maintenance and energy consumption by the data centres are high. When it comes to fog or edge, nodes need less energy to operate and less cost to setup and maintain [111].
Positioning: Both fog and edge are closer to IoT devices generating data and hence they are closer to the user. On the other hand, cloud layer is located far from the IoT layer [150].

By comparing the three platforms—CC, FC and EC—it obviously cannot be deduced that fog/EC can be an alternative for cloud. All three platforms have their own merits and demerits, and all have different contributions to fulfil the demands and aspects. The detailed comparison of CC, FC, and EC is outlined in Table 2.

6 | APPLICATIONS OF FOG COMPUTING

The following are the applications when it comes to FC being applied to different fields. As the remote data is processed closer to where these are generated, it gives real-time experience to users. Some of the pivotal applications which drive the FC paradigm are as follows.

Distributed delivery of content and caching: A large volume of video stuff of different formats and sizes are broadcasted every minute over the internet via live-streaming, gaming, upload, and download. In the internet, video content occupies the maximum traffic while being transmitted over content delivery channels. These channels can facilitate storage of heterogeneous data in cloud database and also distribute web content over various streaming servers situated in diverse networks. As the video content does not lie nearer to the users, they still face a lot of hindrance while viewing it due to jitter, network delay, and buffering [151]. Therefore, it becomes necessary to store the content nearer to network edge so that it can be quickly delivered based on users’ demand. Fogging provides a virtual channel for content delivery and pre-caching the content on demand for the edge.

| TABLE 2 | Comparison FC with CC and EC |
| Parameters | CC | FC | EC |
| Architecture type | Hierarchical or centralised | Hierarchical or decentralised | Distributed or localised |
| Architecture component/hardware | Massive data centres and devices supporting virtualisation | Fog nodes including routers, servers, access points, switches | Edge nodes |
| Server availability | Limited to number of data centres | Several servers and can be scalable | Less servers and scalability as compared to fog |
| Services offered and accessed | Global service via core | Less global via devices connected from edge to core | Local at the Internet edge |
| Amount of data processed | Big data | Less | Less |
| Cognisance of location | Not aware | Aware | Aware |
| Real-time connectivity and response (latency) | Higher | Low | Low |
| Data storage and persistence | Life-time | Short-lived based on requirement | Short-lived based on requirement |
| Data analytics and quality of computation | High | Moderate | Moderate |
| Count of end-users supported | General Internet users | Users of industrial IoT devices | Mobile users |
| Service providers | IBM, Amazon, Microsoft Azure, and Google | Intel and Cisco Iox | Cellular network operators |
| Area of deployment | Deployed indoors owned by cloud service providers | Deployed in roadways, open fields, home, buildings, streets, railway tracks, etc. | Installed by specific service providers in specific areas |
| Internet connection | Needs Internet connection as long as the service is used | Internet connection is optional or needed occasionally | Internet connection is optional or needed occasionally |
| Cost and energy requirement | High | Low | Low |
| Positioning of server | In large-scale devoted buildings, far from the users/edge devices | In devoted locations or can be deployed at the edge | Near the edge nodes |
| Hardware of network | WAN | LAN, WAN, WLAN, cellular, and Wi-Fi networks | LAN, WAN, WLAN, cellular, ZigBee, and Wi-Fi networks |
| Security provision | Throughout the architectural components from cloud to IoT | To the fog nodes | To the edge nodes/devices |

Abbreviations: CC, cloud computing; EC, edge computing; FC, computing.
Enhanced performance of web: When a website or a personal blog is created, obviously, the creators would like to have lot of viewers and visitors. If the website is slow, this would impact on both business as well as viewers and the popularity of the website will go down. On the other hand, popular websites have many unnecessary and unordered contents and often users do not prefer [152]. It is fogging that plays a background role in bringing the performance of web services to the vicinity of the user [153]. Basically the performance of internet or web is viewed in three dimensions: (i) optimisation of web content, (ii) improved browsing, (iii) faster web transfer.

Computational offloading: In case of IoT devices having limited resources, this method is preferred. This technique retains processing power, storage space, lifetime of battery, etc. of the IoT resources. Mainly offloading is carried out for saving energy, minimising latency and maximising battery lifetime, resource utilisation, and QoE. The challenges related to computational offloading are (i) discovering the aspects that influence processes and decisions of offloading particularly for fresh use cases (ii) providing resolution for the heterogeneous and diverse concerns in applications as well as in platforms (iii) designing of lightweight blueprints that can operate in IoT/edge devices without any power constraints [154].

Agricultural applications: Agriculture or farming is another crucial prospect in our life. IoT has also proved its importance in smart farming and smart agricultural processes. This has helped farmers to accomplish more productivity as per the farming standard, water management, crop management, water and pesticide content monitoring, monitoring life stock health, and crops all in real time [155].

Smart cities and smart homes: Cities are the place of industries, business and commercial activities, and manufacturing units which provide empoyment and other services. The ‘smart city’ concept revolves around espousing the applications of IoT to enhance lifestyle of people, infrastructure, cost cutting, innovation, and responsiveness of the cities. Some of the popular smart cities advancements are smart infrastructure monitoring, smart street lighting, smart water management, smart waste management, and smart energy management. Small power stations such as smart grid, traffic signals, logistics, and transport are a part of smart city. When connected to fog, these enjoy instant processing services [45,92,104,112]. Home appliances which have become a part of IoT can connect to fog and have processed data [41,42,84]. The real-time events taking place in cities are handled by big data analytics, which gives the actual picture of what is happening [156]. The role of fogging in big data analytics is already described earlier.

Vehicular connectivity: Mainly two types of vehicular communication are addressed, that is, dynamic/moving vehicles and static/parked vehicles. With regard to computation, first step is to gather data, second step is to analyse this surrounding data such as speed of vehicles, number of vehicles, and presence of any passer-by, and final step is to respond according to this surrounding. With the involvement of fog, data can be pre-processed when there is vehicle to vehicle communication, and this computation can be done faster than when cloud is involved and hence provide real-time or quick response. If every time cloud is involved for vehicular computation, then it would be very difficult to provide real-time feedback because more the number of vehicles, more the amount of computation required. Self-driving cars (SDC) are the future of smart vehicles. So, when it comes to SDC, these vehicles would require quick information of the surrounding so as to avoid any mishap. Many researches direct towards vehicular fogging [8,45,58,59,79,92,108,115,126,147]. Vehicle-to-vehicle communication is achieved using fog-IoT infrastructure within the embedded computers in the automobiles.

Real-time big data analytics: Through platforms such as social media, IoT, artificial intelligence, and machine learning, architecture has moved from data consuming network to data producing network. These act as a source of big data that come in different sizes and formats, which are produced by the connected devices in the IoT network. Around 99% of this huge amount of data is disorganised, and only 1% of it is significant, which is used for analysis and decision-making. Thus, computation and storage of this complete big data in cloud are neither possible nor recommended. This transactional analysis and computation is performed at the fog nodes, and thereafter, cloud handles the business intelligence. As FC offers real-time delay-free services, data can be shifted from manufacturing plants to the logistics and monetary associations which need real-time data analytics [20].

Health care: The main idea of implementing IoT in healthcare is to have the patients perpetually connected to the doctors and the health workers. When the sensitive decisions are made within the device’s range, it helps the information to stay secure within the realm. FC offers its own processing units which can process and transfer data the without bothering cloud services, for example, at the health workers’ office or hospital, which bear the patients’ records. The device will not store any data, but accessing and modification of this data are possible based on set of rules. The IoT devices interfaced in this web will link hospitals, pharmaceutical stores, and other stake holders with the same record every time any modification is needed for that record. The parties with this collaboration will also enjoy the ease of sharing data in a cost-effective, efficient, and secure way as cloud is not involved directly. Different sensors incorporated in the patient's device are used to monitor various bodily parameters and patient’s body location. e-Health applications are time-critical ones demanding instant data processing. Factors such as heart rate, sugar level, and blood pressure are monitored by FC [3,4,11,69,70,85,86,99,109,122,130,136,139].

Simulation models: FC is also applied to form various simulation models where advanced programming and computations are involved for making fog structure more robust [157]. These models can involve machine learning, artificial intelligence, and so on.
7 | CONCERNS OF FOG COMPUTING

Besides the impressive features of FC, it also faces some concerns based on the environment it is implemented.

User security and data privacy: To safeguard user data and IoT device data, proper encryption mechanisms must be followed to reinforce privacy. A vicious user might use a false identification while data is passed through the fog gateways and may hack the data in the fog nodes. As the fog nodes are set up at the internet edge, lot of up keeping cost is needed. Moreover, the cloud is very secure as compared to fog, but the security measures of cloud cannot be extended to fog due to architectural differences between the two. Many cryptography and authorisation methods have been proposed for this challenge [159].

Intricacy: While dealing with a sea of IoT devices, it is obvious that synchronisation is a difficult aspect because different IoT devices will have different system structure with respect to both hardware and software aspects [98]. Though fog is flexible enough for heterogeneous, at times it becomes complicated to operate.

Managing resources: Usually the fog nodes are like network devices with extra computing and storage capabilities. These devices cannot match resource scope of conventional servers. Thus, fog resources have to be wisely managed [26].

Computing delay: While collecting data, over-using the fog resources, computation delay occurs thereby lowering effectiveness of fog servers. Based on priority and mobility, the resources should be utilised, so that delay is avoided [160].

Energy efficiency: As fog layer consists of many fog nodes/devices, its framework is distributed, and often consumes more energy. This factor needs to be resolved, so that energy-efficiency is achieved [14].

Scalability: The count of IoT nodes will undoubtedly come in exceed at least a billion; and every object in the Internet pool needs storage, resource and computing power. Fog being distributed in nature should be able to scale and support with respect to provision of resource, computation and storage for these devices according to their criteria [161].

Being heterogeneous: The structure of every IoT device may vary depending upon its developer, functioning and so on. Some may have sensors while some may have actuators; some may have computation power, whereas some may have storage power. Therefore, IoT pool is always a network with dissimilar devices [129]. Fog nodes should be adaptable to these devices depending on their requirement for supervising and regulating such a network of things.

Being dynamic: Another criterion of the heterogeneous IoT objects is being able to change dynamically in both hardware and software aspects so that the system plan changes accordingly. Performance also changes when the architecture of IoT changes internally. Therefore, fog nodes should also synchronise and arrange the topology and allocate resources accordingly [156].

8 | DESCRIPTION OF THE PROPOSED SYSTEM

Figure 3 describes the flow of the proposed FloTMM system. It shows the position and organisation of different components of the model. The mine's site is outfitted with various sensors to sense the parameters. Due to location awareness offered by fog, the location of these sensors is identified by the fog layer thereby helping to detect any significant fluctuations in the motif. This helps to draw a better conclusion about what is happening in that area and enhance system performance. The area between the fog and cloud is generally connected by WAN. Below are the three main components of the FloTMM model, namely (1) sensing layer, (2) smart fog gateways, and (3) core layer.

Sensing layer: The first layer is composed of the sensors and actuators deployed in the monitoring site or area to monitor the parameters such as slope movement, ground vibration, and lorry collision or positioning. These sensors act as IoT nodes as they are connected to the internet to give live data. The sensed data and signals have ubiquitous recognition and connectivity. Three main areas can be monitored in open-cast mines. The slope movement is monitored using a time-domain reflectometer (TDR) [162,163]. TDR acts as a sensor to sense the movements and shifts occurring in the slope. Slope movements are used to predict slope failure in open-cast mines. Ground vibrations can be monitored through an accelerometer sensor—ADXL345. Ground vibrations occur during blasting in open-cast mines. Collision of lorries is monitored through ultrasonic sensors so that a minimum distance is maintained between two lorries or trucks to avoid collisions. Data captured from these sensors are sent to the fog layer through wired or wireless network such as Bluetooth or cellular network or LoRa, ZigBee, and Wi-Fi.

Smart fog gateways: The next layer is the fog layer that receives the sensed data from the SNs through the edge gateways located at the fog boundary. As mentioned earlier, the fog is smart enough to filter out data, and not every data is sent to the core layer or the cloud for processing. Rather, fog takes care of all the latency-critical applications and real-time data analysis. As FC devices have limited storage capacity, this received data is temporarily stored for analysis and then required feedback is sent back to the source/ground devices. The SNs geo-spatially deployed in groups at each area to be monitored form their corresponding virtual cluster (VC). These VCs, are in turn, mapped to the fog instances (FIIs) on one-to-one basis. Every FI is specific to a geographical location and serves its corresponding VC, having the SNs interlinked among themselves and connected to the internet. Depending on its location, the SN may join or leave any VC and thus get connected or disconnected to the FI. The FI scales itself accordingly based on the demand of SNs.
Core layer: The CC platform lies as the back-end core layer, which deals with big data and performs complex functionalities such as big data analytics and data warehousing, and the data is stored permanently and preserved for future reference. The data from cloud database can be accessed by the end users or clients from anywhere anytime through the internet using a graphical user interface. This big data is used for prediction of a particular scenario, research statistics, and analytics.

9 | APPLYING FOGGING TO MONITOR OPEN-CAST MINES IN REAL-TIME

In the field of mining, slope monitoring of open-cast mines, lorry monitoring, and blasting are the frequent changes that demand delivery of instant response of processed data from the server. In the blink of an eye, a collapse, or a roof fall, or a slope failure may occur in the mining site due to uncertainty of its environment. Therefore, the particular area of the mines' site needs to be monitored every second. This gives rise to bulk data to be stored in the cloud server. The faster the transmission of response, the faster are the prevention measures that can be taken to safeguard the mines’ personnel, equipment, dumper, etc., so that these do not get stuck during mine failure collapse. If there is any delay, the monitoring system will not be able to visualise and analyse the scenario, thereby making the situation worse. It is not feasible on the cloud framework’s part to provide processing of data and instant response to the request made by the IoT devices simultaneously. Hence, there is a need to implement an intermediate fog layer as a communicating gateway in between the cloud server and the sensors deployed in the mines' site. In this regard, the service latency is found out, and performance of the proposed system is evaluated.

9.1 | Service latency

Total delay is the sum of transmission delay and processing delay. Let $\Delta_{IF}$ and $\Delta_{FC}$ be the average delays in transmission of a data packet from a SN/IoT device to the corresponding fog node and from the fog node to the cloud data centre respectively. Thus, the mean transmission latency, $p_{Fog}$ for the data packets of $A_i$ application instances running within fog node $F_i$ is given by

$$T_{Fog-Cloud} = [\Delta_{IF} B_i + \Delta_{FC} b_i]$$ (1)
where $B_i$ and $b_i$ ($B_i > b_i$) are the total number of packets sent cumulatively by $A_i$, application instances to fog node (FN$_i$) and from FN$_i$ to cloud data centres respectively.

The processing latency of a data packet at a fog node is $\psi_F$ and $\psi_C$ at the cloud data centre.

Total processing latency in fog-cloud is given by

$$P_{\text{Fog-Cloud}} = (\psi_F B_i + \psi_C b_i)$$  \hspace{1cm} (2)

where $\psi_F > \psi_C$

Total latency ($T$) = $T_{\text{Fog-Cloud}} + P_{\text{Fog-Cloud}}$  \hspace{1cm} (3)

The latency expression in tradition CC scenario is given by

$$T_{\text{Cloud}} = \Delta_{\text{IC}} (B_i + b_i)$$  \hspace{1cm} (4)

$$P_{\text{Cloud}} = \psi_C (B_i + b_i)$$  \hspace{1cm} (5)

where $\Delta_{\text{IC}}$ and $\psi_C$ is the transmission latency and processing latency, respectively, of a packet from sensor device to cloud data centre.

Total latency ($\tau_{\text{Cloud}}$) = $T_{\text{Cloud}} + P_{\text{Cloud}}$  \hspace{1cm} (6)

The latency expression in fog is given by

$$T_{\text{Fog}} = \Delta_{\text{IC}} (B_i + b_i)$$  \hspace{1cm} (7)

$$P_{\text{Fog}} = \psi_F (B_i + b_i)$$  \hspace{1cm} (8)

Total latency ($\tau_{\text{Fog}}$) = $T_{\text{Fog}} + P_{\text{Fog}}$  \hspace{1cm} (9)

### 9.2 Performance evaluation

Based on the mathematical model described in appendix, performance of FC is evaluated and compared with the traditional CC method. 10 FIs were considered connected to one cloud service provider. It is assumed that VCs consist of SNs that are evenly scattered. MATLAB R2014a is used as an experimental platform based on Intel (R) Core (TM) i7-4770 CPU with 4 GB RAM at 3.40 GHz which runs on Linux (Ubuntu 16.04.3 LTS). The size of the instruction is 64 bits. The user-fog link allows packet transmission of 3464 kilobytes following Poisson arrival pattern. Its instruction size is 8 bytes with mean packet arrival rate being one packet per node per second. The link capacity between the source of data generation and the fog node is taken as 1 Gbps and the inter-fog communication link capacity is taken as 10 Gbps. This experiment is repeated 20 times, and average is taken to avoid the influence of uncertainty factors on the outcome of the experiment.

The percentage of applications which need cloud access is varied and the average latency for all the nodes within a VC is plotted against a variable count of SNs. Figure 4 shows that with decrease in count of fog nodes, there is an increase in the average latency. The average latency is observed to be less as compared to the traditional CC method. In Figure 5, 75% of the data is processed in fog nodes, whereas the remaining in the cloud. Here the number of fog nodes are varied as 2, 4, 8, and 10. With the increase in the number of fog nodes, the average latency reduces.

### 10 Conclusion and Future Scope

FC has just passed its inception stage and is currently being probed at different angles. Instant computation and storage availability are the need of the hour, and FC has entered the world of computing during such a favourable time that it will emphatically influence the working cost of the clients and the end users due to cutting down of the setback suffered during time-critical applications. In this manuscript, a literature on FC and the analogous concepts are surveyed. The survey of a variety of research studies done infers that for raising an extensible and scalable base for IoT with the edge devices, a smart, credible unification of CC and FC is required. This amalgamation could be efficiently applied to monitor scenarios in open-cast mines’ site. Many varieties of monitoring systems could be developed based on FC.

Moving towards the future direction of FC as applied to the field of mining, this blend of FC and CC has to be implemented in open-cast mines with field observation. The main focus will be on field monitoring of slopes in open-cast mines. This observed data has to be modelled and performance evaluation has to be computed to study the scenario more accurately and generate early warnings in the case of a slope failure. From this survey, it can be evidently contemplated that fog serves as a rationally wise platform to regulate and supervise the widely scattered and real-time quality of the looming IoT paradigm. When fog is combined with IoT infrastructure, new business working models could be developed and the existing models or systems could be reconfigured with FC resulting in a new research direction. Fogging seems attractive for some of other applications too, as described in this work. With respect to resource-limited and latency-conscious applications, fogging offers a more affluent platform. So far as the standards are concerned, fogging goes in harmony with the IoT products. FC combined with machine learning and artificial intelligence will expedite its development and make the existing fog systems rich in the future.

### 11 Appendix

Mathematical model of the system:

The mathematical description of FC architecture is given in this section based on the operational functionalities and its composite objects. Two assumptions were taken while formulating the mathematical modelling. First, the total...
The number of SNs installed in each of the monitoring areas is assumed to be fixed with respect to time; and second, it is assumed that there is no packet loss in the network. The SNs in this network can monitor various mine activities such as slope deformation, ground vibration, and lorry collision etc. These SNs are clusters of TDR, accelerometer, and ultrasonic sensors. The total count of SNs traversing over all VC in the primary layer is assumed to be fixed with respect to time [149]. Furthermore, the VCs offer entire coverage for all the SNs up to the bottom most layer.

$N$ is taken to be a set of $x$ numbers of static SNs installed randomly over an area of $a \times b$ in open-cast mine. A single SN is denoted by $\eta \in N$, defined as 10 rows.

$\eta = <N_{id}, N_{ai}, \nu_i, N_c, N_f, N_b, N_l, N_p, N_{cd}, \kappa[p]>$

In the above equation, the unique IDs of SNs are denoted by an integer $N_{id}$. $N_{ai}$ represents the status of a SN. The set $N_{id} = \{0,1\}$ indicates in the Boolean form if the SN is active (symbolised by 1) or inactive (symbolised by 0). The event type sensed by a SN is indicated by $\nu_i$. It can be mathematically represented as an element of the set $\nu = \{\nu_1, \nu_2, \ldots, \nu_n\}$. Here $\nu$ stands for the set of all events that the SNs monitor, and the total number of unique events is denoted by $n$. The hardware like LoRa, ZigBee, Bluetooth, etc. used by the SNs for communicating wirelessly are denoted by the row $N_c$. The range of frequency at which the SNs operate is denoted by $N_f$. The battery level of a SN is given by $N_b$. The SN stops communicating once the $N_b$ value drops below the threshold value. The location of a SN is denoted by $N_l$. The processor specifications of the SN are denoted by $N_p$. These specifications consist of details such as specifications of bus, speed of processing core, and size of cache memory or internal register. Lastly, the cluster ID of a SN is denoted by $N_{cd}$. The last row of SN definition is an 1D array $\kappa[p]$ (having $p$ elements). It holds the instance IDs of the application instance (AI)s which are currently running on the device. An AI exists only in the presence of a parent application in a device.

The application $A$ has two tuples given by:

$A = <A_{id}, A_i>$

Here the application ID is denoted by $A_{id}$ and $A_i$ represents the system requirements needed to run the application that includes operating system version, storage details for both primary storage (RAM), and secondary storage (ROM) and processor details.

The AI has five tuples given by:

$AI = <AI_{id}, N_{id}, A_{id}, AI_r, AI_i>$

Here AI ID is denoted by $AI_{id}$ that is similar to a system generated process ID. $N_{id}$ and $A_{id}$ have the same designation as mentioned earlier. The resource requirement of the AI in terms of computation, network bandwidth, and ability to analyse or process and storage power is denoted by the tuple $AI$. More than one instance of an application may simultaneously run on a SN, and are identified by their unique instance IDs. $AI_r$ is a Boolean variable, such that $AI_r = \{0,1\}$, indicating values 0 (inactive state) and 1 (active state) of the AI. AVC, designated by $V$, is a logical boundary consisting of many localised SNs, and represented mathematically as four tuples:

$V = <V_{id}, V_{d[m]}, V_r, F_{id}>$

Here ID of VC is denoted by $V_{id}$, and $V_{d[m]}$ is a single dimensional array of size ‘u’ that holds the IDs of all the SN constituents. At any given point of time, the array length changes dynamically which indicates that the count of SN in the VC may leave or join the VC altering the array length. $V_r$ denotes the region monitored by a VC including all
intermediate SNs. For SNs to be able to monitor effectively, the geographic terrain is divided into many non-overlapping regions and each region is mapped to a VC. The physical FI to which a VC is mapped is known by \( F_{id} \). Mathematically, \( (I) \) denotes a FC instance or FI which is defined by three tuples:

\[
I = < F_{id}, F_{ap}, F_{[x]} >
\]

Here \( F_{id} \) is the unique ID of FI, and \( F_{ap} \) is the access point which connects the FI to the core CC framework. Third, \( F_{[x]} \) is a single dimensional array of size \( \nu \) which holds the device IDs of all the constituent FC devices of a FI. A FC module is denoted by \( M \), based on its type and characteristic features by three tuples:

\[
M = < M_{id}, M_{type}, M_{t} >
\]

Here \( M_{id} \) and \( M_{type} \) are the ID and the type (such as gateway, router, processing unit, or storage) of the FC device. \( M_{t} \) tuple stores the related specifications and hardware of the fog device.

**ORCID**

Devendra Kumar Yadav 
https://orcid.org/0000-0002-5748-9201

**REFERENCES**

1. Bonomi, F., et al.: Fog computing and its role in the internet of things. In: 2012 Proceedings of 1st edition of the MCC Workshop on ACM Mobile Cloud Computing, pp. 13–15 (2012)
2. De Donno, M., Tango, K., Dragoni, N.: Foundations and evolution of modern computing paradigms: cloud, IoT, edge, and fog. IEEE Access. 7, 150936–150948 (2019)
3. Gia, T.N., et al.: Fog computing in healthcare Internet of Things: a case study on ECG feature extraction. In: 2015 Proceedings of 15th IEEE International Conference on Computer Information Technology (CIT), 14th IEEE International Conference on Ubiquitous Computing Communications (IUCC); 13th IEEE International Conference on Dependable, Autonomic and Secure Computing, pp. 356–363. (2015)
4. Rahmani, A.M., et al.: Exploiting smart e-Health gateways at the edge of healthcare Internet-of-Things: a fog computing approach. Futur. Gener. Comput. Syst. 78, 641–658 (2018)
5. Rimal, B.P., Choi, E., Lumb, I.: A taxonomy and survey of cloud computing systems. In: 2009 IEEE Fifth International Joint Conference on INC, IMS and IDC (NCM), pp. 44–51. IEEE (2009)
6. Al-Fuqaha, A., et al.: Internet of things: a survey on enabling technologies, protocols, and applications. IEEE Commun. Surv. Tutor. 17(4), 2347–2376 (2015)
7. Sun, X., Ansari, N.: Green cloudlet network: a sustainable platform for mobile cloud computing. IEEE Trans. Cloud Comput. 8(1), 180–192 (2020)
8. Yassine, A., et al.: Cloudlet-based intelligent auctioning agents for truthful autonomous electric vehicles energy crowdsourcing. IEEE Trans. Veh. Technol. 59(4), 1–17 (2020)
9. Kuang, Q., et al.: Analysis on computation-intensive status update in mobile edge computing. IEEE Trans. Veh. Technol. 69(4), 4353–4366 (April 2020)
10. Shi, W., et al.: Edge computing: vision and challenges. IEEE Internet Things J. 3(5), 637–646 (2016)

11. Farahani, B., et al.: Towards fog-driven IoT eHealth: promises and challenges of IoT in medicine and healthcare. Futur. Gener. Comput. Syst. 78, 659–676 (2018)
12. Sharma S., Saini H.: Fog assisted task allocation and secure deduplication using 2FBO and MoW in cluster-based industrial IoT (IoT). Comput. Commun. 152, 187–199 (2020)
13. Baucaus, M.J., Spachos, P.: Using cloud and fog computing for large scale IoT-based urban sound classification. Simul. Model. Pract. Theory. 101, 102013 (2019–2020)
14. Munir, A., Kansalak, P., Khan, S.U.: IFChIoT: integrated fog cloud IoT. IEEE Consum. Electron. Mag. 6(3), 74–82 (2017)
15. Mahmud, R., Kotagiri, R., Buyya, R.: Fog computing: a taxonomy, survey and future directions. In: Di Martino, B., et al. (eds) Internet of Things, vol. 1, pp. 103–130. Springer, Singapore (2018)
16. Naha, R.K., et al.: Fog computing: survey of trends, architectures, requirements, and research directions. IEEE Access. 6, 47980–48009 (2018)
17. Bonomi, F., et al.: Fog computing: a platform for internet of things and analytics. In: Big Data and Internet of Things: a Roadmap for Smart Environments, pp. 169–186. Cham, Switzerland: Springer (2014)
18. Liu, C.F., et al.: Poor, dynamic task offloading and resource allocation for ultra-reliable low-latency edge computing. IEEE Trans Commun. 67(6), 4132–4150 (2019)
19. Sarkar, S., Misra, S.: Theoretical modelling of fog computing: a green computing paradigm to support IoT applications. IET Netw. 5(2), 23–29 (2016)
20. Anwar, M.R., et al.: Fog Computing: An Overview of Big IoT Data Analytics. Wireless Communications and Mobile Computing, 2018, 1–22 (2016). http://dx.doi.org/10.1155/2018/7157192
21. Pallas, F., Rasche, P., Bermach, D.: Fog computing as privacy enabler. IEEE Internet Comput. 24(4), 15–21 (2020)
22. Sun, H., Yu, H., Fan, G.: Contract-Based Resource Sharing for Time Effective Task Scheduling in Fog-Cloud Environment. IEEE Transactions on Network and Service Management. 17(2), 1040–1053 (2020)
23. Baccarelli, E., et al.: Fog of everything: energy-efficient networked computing architectures, research challenges, and a case study. IEEE Access. 5, 9882–9910 (2017)
24. Chiang, M., Zhang, T.: Fog and IoT: an overview of research opportunities. IEEE Internet Things J. 3(6), 854–864 (2016)
25. Alli, A.A., Alam, M.M.: The fog cloud of things: a survey on concepts, architecture, standards, tools, and applications, Internet of Things 9, 100177 (2020)
26. Shahzadi, R., et al.: Three-tier fog networks: enabling IoT/5G for latency sensitive applications. China Commun. 16(3), 1–11 (2019)
27. Puriwal, D., et al.: Fog computing security challenges and future directions [energy and security]. IEEE Consum. Electron. Mag. 8(3), 92–96 (2019)
28. Suo, H., et al.: Security in the internet of things: a review. In: Proceedings of 2012 International Conference on computer science and electronics engineering (ICCSEE), vol. 3, pp. 648–651. IEEE (2012)
29. Tewari, A., Gupta, B.B.: Security, privacy and trust of different layers in Internet-of-Things (IoT) framework. Futur. Gener. Comput. Syst. 108, 909–920 (2018)
30. Aazam, M., Huh, E.N.: Fog computing: the cloud multi-tier fog data centers. IEEE Trans. Sustain. Comput. 4(2), 145–155 (2019)
31. Tortonesi, M., et al.: Taming the IoT data deluge: an innovative information-centric service model for fog computing applications. Futur. Gener. Comput. Syst. 93, 888–902 (April 2019)
32. Du, M., et al.: A differential privacy-based query model for sustainable fog data centers. IEEE Trans. Sustain. Comput. 4(2), 145–155 (2019)
33. Lu, R., et al.: A lightweight privacy-preserving data aggregation scheme for fog computing-enhanced IoT. IEEE Access. 5, 3302–3312 (2017)
34. Baccarelli, E., Scarpiniti, M., Momenzadeh, A.: EcoMobiFog-design and dynamic optimization of a 5G mobile-fog-cloud multi-tier ecosystem for the real-time distributed execution of stream applications. IEEE Access. 7, 55565–55608 (2019)
null
135. El-Latif, A.A.A., et al: Secure quantum steganography protocol for fog cloud internet of things. IEEE Access. 6, 10332–10340 (2018)

136. Lin, K., Pankaj, S., Wang, D.: Task offloading and resource allocation for edge-of-things computing on smart healthcare systems. Comput. Electr. Eng. 72, 348–360 (2018)

137. Moysiadis, V., Sarigiannidis, P., Moscholios, I.: Towards distributed data management in fog computing. Wirel. Commun. Mob. Comput. 1, 1–14 (2018)

138. Souza, V.B., et al: Towards a proper service placement in combined Fog-to-Cloud (F2C) architectures. Futur. Gener. Comput. Syst. 87, 1–15 (2018)

139. Mahmoud, M.M.E., et al: Towards energy-aware fog-enabled cloud of things for healthcare. Comput. Electr. Eng. 67, 58–69 (2018)

140. Deng, Y., Chen, Z., Zhang, D., Zhao, M.: Workload scheduling toward worst-case delay and optimal utility for single-hop Fog-IoT architecture. IET Commun. 12(17), 2164–2173 (2018)

141. Ku, Y.J., et al.: 5G radio access network design with the fog paradigm: confluence of communications and computing. IEEE Commun. Mag. 55(4), 46–52 (2017)

142. Wu, J.X., et al.: Blood leakage detection during dialysis therapy based on fog computing with array photocell sensors and heteroassociative memory model. Healthc. Technol. Lett. 5(1), 38–44 (2018)

143. Gu, L., et al.: Cost efficient resource management in fog computing supported medical cyber-physical system. IEEE Trans. Emerg. Top. Comput. 5(1), 108–119 (2017)

144. Brogi, A., Forti, S.: QoS-aware deployment of IoT applications through the fog. IEEE Internal Things J. 4(5), 1185–1192 (2017)

145. Wang, T., et al.: Trajectory privacy preservation based on a fog structure for cloud location services. IEEE Access. 5, 7692–7701 (2017)

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

147. Wang, M.Z., et al.: Vehicles assignment with over-emission intensity considerations: a perspective on integrating the market mechanism with government control. IEEE Access. 4, 5098–5110 (2016)

148. Sarkar, S., et al.: Assessment of the suitability of fog computing in the context of internet of things. IEEE Tans. Cloud Comput. 6(1), 1–14 (2015)

149. Maiti, P., et al: Efficient data collection for IoT services in edge computing environment. In: 2017 Proceedings of the International Conference on Information Technology (ICIT), pp. 101–106. IEEE (2017)

150. Maiti, P. et al.: QoS-aware fog nodes placement. In: Proceedings of the 4th IEEE International Conference on Recent Advances in Information Technology (RAIT), pp. 1–6. IEEE (2018)

151. Zeydan, E., et al: Big data caching for networking: moving from cloud to edge. IEEE Commun Mag. 54(9), 36–42 (Sep. 2016)

152. Reda, A., Cutrell, E., Noble, B.: Towards improved web acceleration: leveraging the personal web. In: Proceedings of the 9th International Conference on Mobile Systems, Applications and Service Co-located Work, NSDR’11, pp. 57–62. Bethesda, Maryland (2011)

153. Taleb, T., et al: On multi-access edge computing: a survey of the emerging 5G network edge cloud architecture and orchestration. IEEE Commun. Surv. Tutorials. 19(3), 1657–1681 (2017)

154. Kumar, K., et al: A survey of computation offloading for mobile systems. Mob. Netw. Appl. 18(1), 129–140 (February 2013)

155. Chiochano, O., Saokaew, A., Boonchieng, E.: IoT for smart farm: a case study of the Lingzhi mushroom farm at Maejo University. In: 2017 Proceedings of 14th International Joint Conference on Computer Science and Software Engineering (ICCSSE), pp. 1–6. IEEE (2017) (2017)

156. Ali, A.A., Alam, M.M.: SecOFF: Machine learning based secure offloading in Fog-Cloud of things for smart city applications, Internet of Things. 7, 100070 (September 2019)

157. Margariti, S.V., Dimakopoulos, V.V., Tsoumanis, G.: Modeling and simulation tools for fog computing—a comprehensive survey from a cost perspective. Futur. Internet 12(5), 89 (May 2020)

158. Hassan, S.F., Fareed, R.: Video streaming processing using fog computing. In: 2018 International Conference on Advanced Science and Engineering (ICOASE), pp. 140–144. IEEE (2018)

159. Yousefpour, A., et al: All one needs to know about fog computing and related edge computing paradigms: a complete survey. J. Syst. Archit. 98, 289–330 (February 2019)

160. Chang, C., Narayana Sirima, S., B乌鲁a, R.: Indie Fog: an efficient fog-computing infrastructure for the internet of things. Computer (Long. Beach. Calif). 50(9), 92–98 (2017)

161. Naha, R.K., Garg, S., Chan, A.: Fog-computing architecture: survey and challenges. In: Big Data-Enabled Internet Things, pp. 199–223. (November 2019)

162. Yadav, D.K., et al: Critical review on slope monitoring systems with a vision of unifying WSN and IoT. IET Wirel. Sens. Syst. 9(4), 167–180 (August 2019)

163. Yadav, D.K., et al: Design of real-time slope monitoring system using time-domain reflectometry with wireless sensor network. IEEE Sensors Lett. 3(2), 1–4 (2019)

How to cite this article: Yadav DK, Mishra P, Jayanthu S, Das SK, Sharma SK. Application of IoT-Fog based real-time monitoring system for open-cast mines—A survey. IET Wirel. Sens. Syst. 2021;11:1–21. https://doi.org/10.1049/wss2.12011