Journey from Cloud of Things to Fog of Things: Survey, New Trends and Research Directions

Ananya Chakraborty[1], Mohit Kumar[1], Nisha Chaurasia[1] and Sukhpal Singh Gill[2]

[1]Dr B R Ambedkar National Institute of Technology, Jalandhar, Punjab, India
[2]School of Electronic Engineering and Computer Science, Queen Mary University of London, London, UK
ananyac.it.19@nitj.ac.in; kumarmohit@nitj.ac.in; chaurasian@nitj.ac.in; s.s.gill@qmul.ac.uk
Corresponding Author: Email: kumarmohit@nitj.ac.in

Abstract: With the advent of the Internet of Things (IoT) paradigm, the cloud model is unable to offer satisfactory services for latency-sensitive and real-time applications due to high latency and scalability issues. Hence, an emerging computing paradigm named as fog/edge computing was evolved, to offer services close to the data source and optimize the quality of services (QoS) parameters such as latency, scalability, reliability, energy, privacy, and security of data. This article presents the evolution in the computing paradigm from the client-server model to edge computing along with their objectives and limitations. A state-of-the-art review of Cloud Computing and Cloud of Things (CoT) is presented that addressed the techniques, constraints, limitations, and research challenges. Further, we have discussed the role and mechanism of fog/edge computing and Fog of Things (FoT), along with necessitating amalgamation with CoT. We reviewed the several architecture, features, applications, and existing research challenges of fog/edge computing. The comprehensive survey of these computing paradigms offers the depth knowledge about the various aspects, trends, motivation, vision, and integrated architectures. In the end, experimental tools and future research directions are discussed with the hope that this study will work as a stepping stone in the field of emerging computing paradigms.

Index Terms—**Cloud computing, Internet of Things, CoT, Fog computing, Latency.**

1. INTRODUCTION

Different computing paradigms have evolved over the decades as per the needs in various situations. The concept of the Cloud computing paradigm has evolved from distributed systems and it has gained attention by offering a plethora of services to end-users anywhere and anytime on pay per use basis. It has also become the business model in the last decade due to well-known characteristics like scalability, multi-tenant environment, virtualization, and resource pooling. It is observed that a large number of industries, government, as well as private organization, have shifted their applications and data to the cloud platform. Further, the Cloud is also playing a significant role in the advancement of Internet of Things (IoT) applications. As per the transformation and movement in ubiquitous computing, we need the integration of the cloud with IoT and terminology referred to as Cloud of Things (CoT). The integration of cloud architecture with IoT provides benefits like higher computation power. When fog is introduced to this existing framework, the workload is balanced. For instance, if data has to be collected and analyzed in real-time with minimum delay, then the fog/edge computing paradigm is preferred, and if data analysis requires high computation-oriented service or storage, the data is transferred to the cloud for the processing due to its vast resources. The evaluation of different computing paradigms has been explained in Fig. 1 along with their objectives and limitations.

Cloud Computing: To bring services and resources closer to the user, regardless of geographical setting, Cloud was conceptualized. It enabled users to request and obtain services faster, cheaper and without the need to own resources. Instead, the resources are provided by Cloud, along with security, mobility and other features. Cloud computing allows the deployment of multiple Virtual Machines (VMs) over a single physical host using the concept of virtualization. VM refers to virtual machines which provide services as per the demand and requirement, without the user knowing the physical specifications. These VMs offers various types of services to end-user in isolation mode while preserving the privacy of user's applications and data. There are several service models available in the cloud that offered the services to end-users as per their demand [1-4].

Internet of Things (IoT): Devices connected preferably over the Internet capable of processing, storing and transferring data are considered as Internet of Things (IoT). IoT involves heterogeneous end devices that are connected through the internet. They have the capability to interact over the network and process the data over internet-connected devices. Examples of IoT devices are sensors, actuators, smart mobiles, smart watches, and smart security systems. This framework helps bring the internet nearer to the user devices, more than the cloud [5]. Cloud computing in itself as a model although powerful, is not utilized efficiently. Including end devices increases the reach of Cloud processing power. End devices on their own have lesser computing power and storage. Hence, the incorporation of IoT in the Cloud framework was conceptualized to reap the benefits of both paradigms.

Cloud of Things (CoT): The end smart devices in a framework have minimal capability to process task requests. Hence, if a task request demands higher computation power, IoT is not able to execute the task. In such a scenario, the cloud framework can be integrated along with IoT to increase the computing power of the existing framework. Then IoT is able to transfer data to Cloud for processing and storage as well. This framework of CoT helps in collecting and processing data over a large geographical area.

Fog Computing: CoT as a paradigm does have a larger coverage till the end devices, but the large distance between Cloud and IoT adds latency to the processing. The end nodes of a CoT framework might receive requests with sensitive data and lesser time latency permissible. The need to process geographically distributed IoT data with minimum latency, and high security, was not achievable by cloud computing. A new computing paradigm named fog computing was invented to be in proximity with IoT devices [6]. Fog computing allows heterogeneous end devices that carry out their own computation by bringing the processing power to the devices rather than transferring the data to the cloud [4]. There are three layers in this paradigm: end devices, fog layer consisting of servers, and cloud [8].

Edge Computing: Edge computing is a distributed computing paradigm that enables the processing of data on the distributed edge devices. Edge devices refer to the devices situated on the edge of the framework. There are no servers in edge as in fog computing, IoT data or applications are processed at the source node where the data is originated i.e., it brings the computation and storage closer to the data source and reduces the latency along with bandwidth [7-8].

Fog of Things: The emerging computing paradigm named fog has been introduced to offer the services for latency sensitive CoT applications. This presence of cloud in the framework increases the computation power, while the presence of fog reduces latency. This provision improves load balancing, reduces latency and also increases the computing power of the entire architecture.

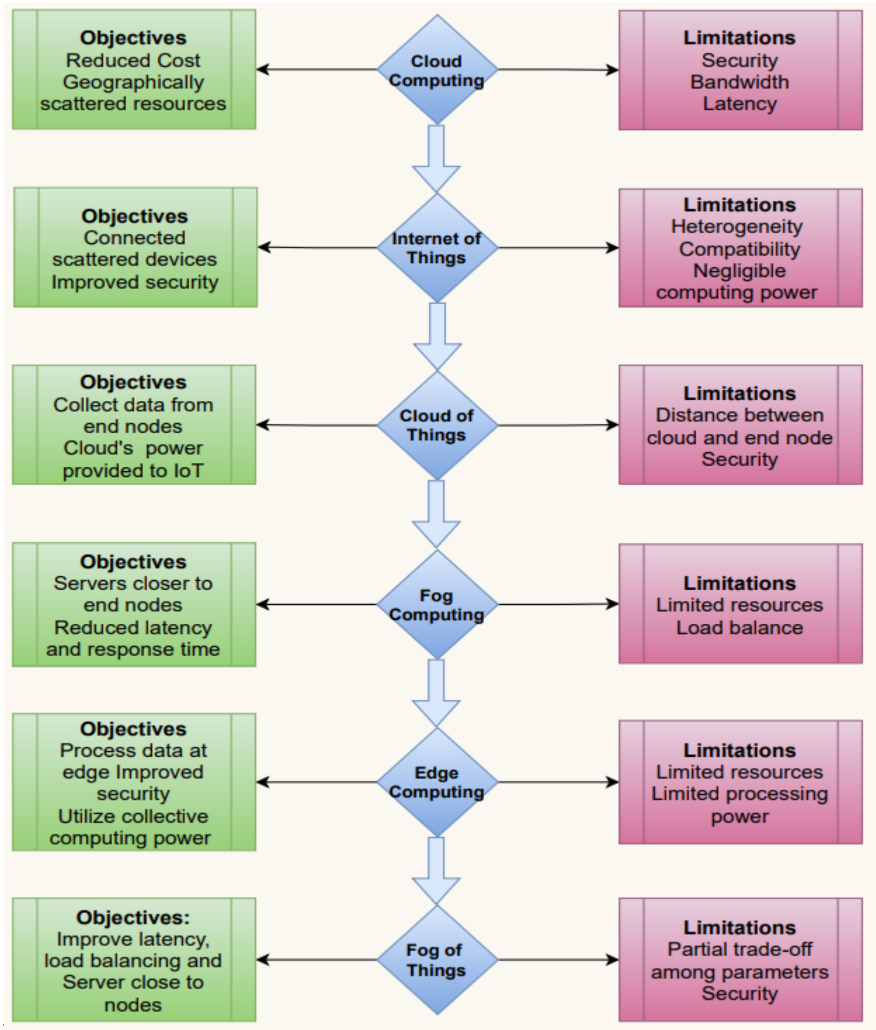

Fig. 1.1 Evolution of Computing Technologies

Cloud Computing, Cloud of Things, Fog computing and Fog of Things have been discussed in detail about research problem and motivation as shown in Fig. 1.2. The evolution of technologies in their order is presented in Fig 1.1. Limitations and objectives follow the technologies. The concept of Cloud framework was designed to achieve higher computing power and portability of the services. All internet services such as portable applications and drive storage, are the applications of the cloud. Although this allowed great flexibility to the consumers, it also is susceptible to security attacks because of the third-party resources used. Hence, different architectures and security protocols are studied for the trade-off between powerful operation and security.

Also, the Internet of Things was conceptualized with time. This means that the real-time data could be collected from the devices and processed without time delay. This benefit is able to collect data from more devices than earlier and gave the rise to the need for edge security and time-sensitive processing. Hence, this gave birth to the idea of the fog and edge computing paradigm, where the servers and resources are brought closer to the edge/end smart devices. Moreover, if some operation requires storing and processing a large amount of data that is not time-sensitive, then the data is transferred to the cloud.

This in turn ensures that data analytics from a large geographic range is carried out and stored for future uses. To entrust this, the following sections about cloud, fog, and fog-cloud integrated paradigms and architecture are discussed in detail along with recent as well as advanced studies in the fields.

1.1 Motivation: Research on cloud computing has been conducted for more than a decade now. It was one of the well-known research areas before the adoption of IoT technology and its applications. In addition, the computing paradigms like fog, edge, IoT, mist, and others are becoming the trending area for research more recently. Research interests include efficiently improving QoS parameters like completion time, cost, energy consumption, throughput, and availability. The evolution of cloud computing has been surveyed by quite a lot of recent studies discussing up-to-date research trends. Various security threats and issues along with their solutions are discussed with the help of the classification of attacks in the aforementioned studies [9]. The existing resource management techniques and taxonomy of cloud computing are discussed in detail [9-13]. These studies in various paradigms need to be brought under an umbrella to compare and generalize the techniques and algorithms used in specific situations. It will help in observing the trade-offs of these algorithms and possible limitations to work upon.

1.2 Related Surveys: A comprehensive survey of resource management techniques is presented [14] and the simulators used for all cloud, fog, and edge computing paradigms are discussed [15]. A survey of communication protocols in cloud-integrated fog computing is provided with classified communication protocols [16]. Classified fog architecture and applications along with research gaps and proposed framework are discussed [17-20]. In the end, security threats in the fog computing paradigm are discussed in detailed [21].

TABLE 1.1: COMPARISON AMONG THE DIFFERENT SURVEYS

Author	Cloud algorithms	Fog algorithms	CoT	FoT	Analysis of Tools	Classification/ Taxonomy	State of art	Year-wise analysis	Graphical
Hu et al., 2017**[22]**	x	✓	x	x	x	x	✓	x	x
Mahmud et al., 2018 **[23]**	x	✓	✓	x	x	x	x	x	x
Mouradian et al., 2018 **[24]**	x	✓	✓	x	x	✓	x	x	x
Mukherjee et al., 2018 **[25]**	x	✓	✓	x	x	✓	✓	✓	x
Zhang et al., 2018 **[26]**	x	✓	x	x	x	✓	x	x	x
Carpio et al., 2019 **[27]**	x	x	✓	✓	x	✓	✓	x	x
Aslanpour et al., 2020 **[28]**	✓	✓	✓	x	✓	✓	x	x	x
Bendechache et al., 2020 **[29]**	x	x	✓	✓	✓	x	x	✓	x
Arunarani et al., 2019 **[30]**	✓	x	x	x	x	✓	✓	✓	✓
Ghomi & Rahmani, 2017 **[31]**	✓	x	✓	x	x	✓	✓	✓	✓
Kumar & Kumar, 2019 **[32]**	✓	x	x	x	✓	✓	✓	✓	✓
Singh et al., 2016 **[33]**	✓	x	x	x	x	✓	✓	✓	✓

Kumar et al., 2019 **[34]**	✓	x	x	x	✓	✓	✓	✓	✓
Barros et al., 2020 **[35]**	✓	✓	x	x	x	x	x	✓	✓
Hong & Varghese, 2019 **[36]**	x	✓	x	x	x	✓	x	x	✓
Prokhorenko & Ali Babar, 2020 **[37]**	✓	✓	x	x	x	✓	✓	✓	✓
Chegini et al., 2021 **[206]**	✓	✓	✓	✓	x	x	✓	x	✓
Samann et al., 2021 **[207]**	✓	✓	✓	x	x	✓	✓	x	x
This survey (Our paper)	✓	✓	✓	✓	✓	✓	✓	✓	✓

The studies conducted are compared on the basis of different paradigms discussed and how are they presented in Table 1.1. The table aims to present how various studies in the same field have collected and presented the selected algorithms and frameworks. From the rigorous studies, it has been observed that none of the authors has covered all the fields. The topics covered by the studies are shown by ✓ otherwise x. Cloud computing, fog computing, Cloud of Things, Fog of things are the paradigms discussed in detail in the current study. The rest of the parameters for the comparison in Table I are:

a) Analysis of Tools: It is checked if the tools used for the respective paradigms are analyzed.
b) Classification/ Taxonomy: This parameter of the table defines if the study presents and also classifies the techniques.
c) Year-wise analysis: It is seen if the study has provided a year-wise analysis of the articles to provide a view on developing algorithms. This analysis would give an idea as to which area is lesser focused on by researchers, for further improvement.
d) State of art: The studies are compared in the table if state of art technologies has been discussed.
e) Graphical: The table checks and compares if the studies conducted in the field have represented algorithms in a graphical manner for easy interpretation.

These surveys discussed the taxonomy or classification, state-of arts techniques, and quality of service for the different computing paradigms. It is also explained as how one QoS parameter is achieved efficiently at the expense of another. The studies also analyze the experimental environments (real or simulated) where the performance of the proposed approach is evaluated. It is paramount to bring the various recent research conducted for as many QoS parameters as possible to picture the relationship between the parameters and the cost to be paid for achieving each. An entire existing survey shows the way to newer studies and research gaps in the above-mentioned fields.

Few of the fields/parameters mentioned in the table are untouched in most of the surveys. Hence, we require a complete survey to develop and incorporate the research in the field of cloud computing, IoT, CoT, and emerging FoT computing. To the best of our knowledge, this article as a fresh contribution will cover the entire field and parameters mentioned in Table 1.1.

1.3 Structure: This study represents the structure of the article in Fig. 1.2. Starting with the introduction to the evolution of technologies, the sources of the papers studied, are discussed in section 1. Cloud computing framework, service, and deployment models along with features, applications, and current research challenges are discussed in section 2. Following the discussion is a year-wise analysis of recent research challenges as mentioned and at the last, still, existing limitations are discussed.

The same structure is followed in section 3 for the integration of cloud with IoT as CoT, and the emerging computing paradigm fog/edge computing in section 4. Architecture, applications, and existing techniques for Fog of Things are discussed in section 5. The experimental environment for the entire computing paradigm is discussed and analyzed. In the end, the conclusion and future research direction are discussed. The parameters discussed in the paper are abbreviated as Execution time (ET), Makespan time (MST), Response time (RT), Transmission rate (TR), Propagation delay (PD), Return time (RNT), Completion time (CT), Service level agreement violation (SLA-V), Energy consumption (EC), CPU utilization (CPU-U), Fault tolerant (FT), Bandwidth (BW), Resource management (RM) as shown in Table 1.2. These parameters are explained in the following section and later used in comparing the algorithms.

Our Contributions: There have been quite a few surveys for the study of the three aforementioned frameworks recently. In this paper, the frameworks are discussed with their applications and motivations. The limitations and challenges are also discussed to study or analyze the improved QoS parameters and the trade-offs between the conflicting parameters. This article studies the following aspects of the paradigms:

a. The transformation of computing paradigms is presented with its advantages and limitations.
b. Complete updated study about the cloud and CoT framework, applications, algorithms, and research issues.
c. The roles and mechanisms of fog/edge computing are discussed along with architectures, features, applications, and existing research challenges.
d. For each of the aforementioned paradigms (Cloud Computing, Fog Computing, Cloud of Things and, Fog of Things), there exists simulating environments and real environments. They are discussed with their respective properties and limitations.
e. Present research work provides deep knowledge to the new researcher about the various aspects, trends, motivation, and future directions.

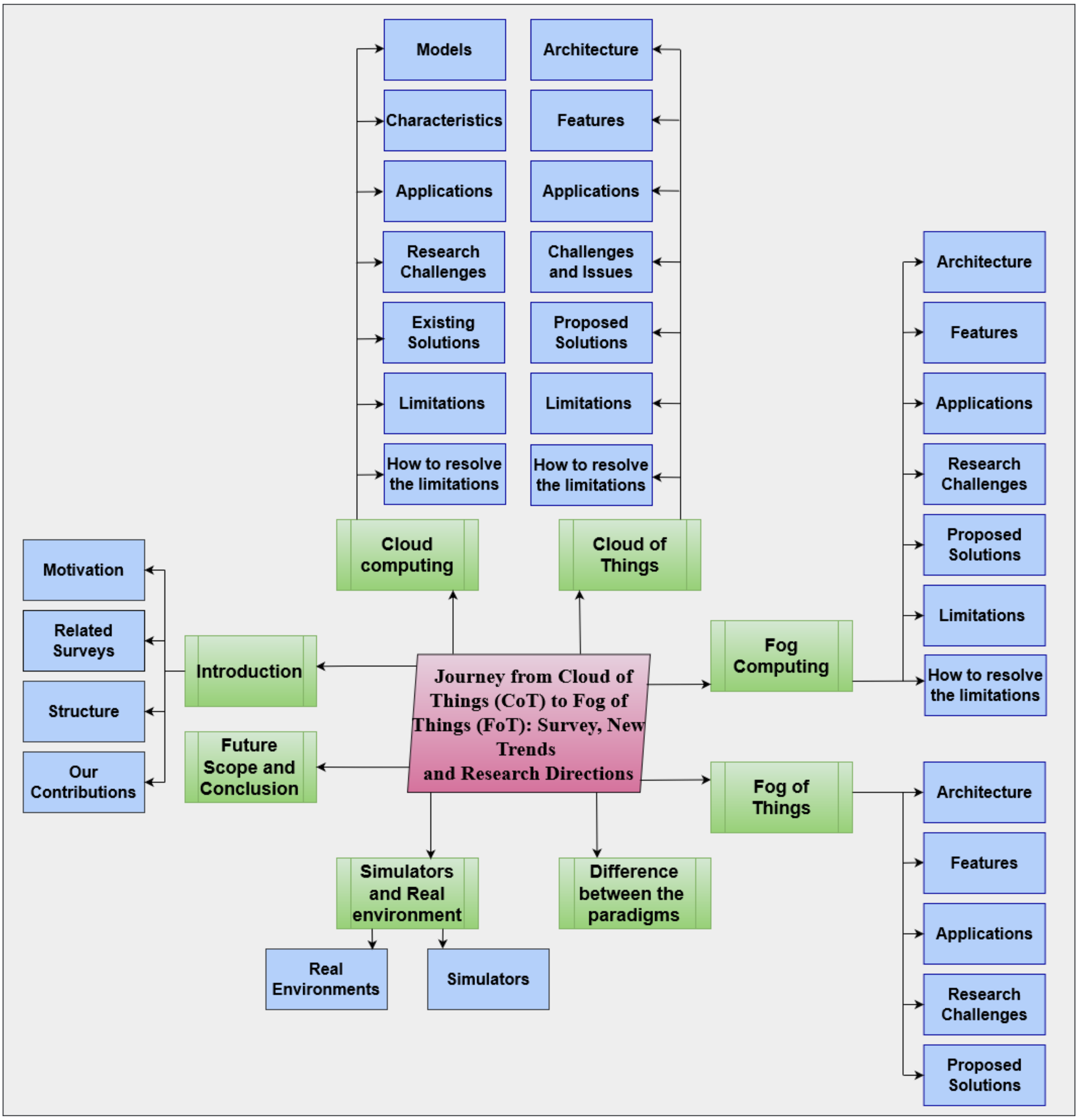

Fig. 1.2: Structure of the paper

1.4 Parameters: This study aims to collect and categorize the recent trends in cloud, fog and fog integrated cloud frameworks in an organized way. Also, the study will help in observing how the recent studies fall short in separate parameters. This study will allow reducing the gap in the trends. The QoS parameters under which the recent studies are organized are discussed below one by one. The aim to categorize the studies under these attributes is to collect and compare recent trends and work. The categories will help in observing techniques adopted for specific environments and corresponding results achieved where at least one objective is fulfilled. Along with this, the study also presents the trade-offs aforementioned algorithms face against the other conflicting parameters. The most common parameters on which the studies are compared, are as follows:

Completion time: Completion time is the time a request takes from the start of processing till its completion.

TABLE 1.2: PARAMETERS AND THEIR ABBREVIATIONS AS USED IN THE PAPER

Parameters	**Abbreviation used**
Execution time	ET
Makespan time	MST
Response time	RT
Transmission rate	TR
Propagation delay	PD
Return time	RNT
Completion time	CT
Service level agreement violation	SLA-V
Energy consumption	EC
CPU utilization	CPU-U
Fault tolerant	FT
Bandwidth	BW
Resource management	RM

Makespan time: It is the total amount of time required to complete a set of requests over all the virtual machines.

Response time: The time taken to respond to the end-user request is known as response time.

Reliability: This parameter assures end-user regarding the availability of services and security of data.

Authentication: This parameter has focused on proving the validity of the resources provided by included parties.

SLA violation: A service level agreement is a contract between the service provider and client about the quality parameters of the service. It is required to ensure that both client and server components are in agreement.

Energy efficiency: Energy is required to run and ensure the maintenance of servers. Most of the time, a lot of energy is wasted due to the system or resources in idle condition. Hence it is an important parameter that decides the efficiency of the framework in saving energy and hence also reducing cost.

Availability: Availability means that the cloud resources will be available to execute the end-user request i.e., downtime of the services will be at least as much as possible, and services should be available all the time for the end users.

Fault Tolerance: There are chances of fault occurrence in a framework, such as power outage. This parameter hence determines whether a system or framework is capable of processing the event after such an occurrence.

Cost function: It determines the total amount or charges paid by the users for accessing services. Cloud and the paradigms provide resources and services on some amount. This cost is defined by technologies such as database servers, processors and VMs used.

Delay: The time latency observed in a system while accepting, processing and transferring requests, is called delay.

Mobility: While running requests, there might arise the need for higher computation power or different resources. Requests will then need to be transferred to desired VMs, without any involvement from the user. Hence, mobility refers to the ability of the components to transfer services as required.

Heterogeneity: Data transmitted over the Internet can be of different types and originate from end devices heterogeneous in nature. This feature of the data is called heterogeneity. Hence, this parameter is used to check whether the server/s can process and provide the services to end devices of heterogeneous nature.

Scalability: Scalability defines the capability of the servers to increase or decrease the resource allocation as per the demand of end-users.

Bandwidth: Requests and data sent over to and from the cloud need sufficient bandwidth to ensure efficient transmission with the least delay possible.

Round Trip time: It is the time elapsed between sending the request and receiving the response.

These parameters are used to compare the algorithms and their effects. After organizing studies under the categories, these parameters are used to determine the performance of the algorithm or system. For this, we have collected the articles from different sources as shown in Fig. 1.3 and rigorously reviewed each article to find the research gap in this field. Most of the articles are based on scheduling and provisioning techniques in cloud computing. Although after the adoption of IoT applications, it has been observed that IoT, CoT, and fog/edge computing have become the most significant research areas as shown in Fig. 1.4. Initially, we will discuss the sources where articles are extracted using the keywords like scheduling in cloud computing, IoT and its applications, CoT and its applications, fog computing with IoT, and many more keywords.

1.5 Sources for the literature: Recent works in cloud and fog environments are studied from the following sources/journals: ACM, IEEE, Elsevier, MDPI, Springer, and others. As it can be observed from Fig. 1.3, most of the articles that we have collected are from IEEE explore, Springer, and Science Direct.

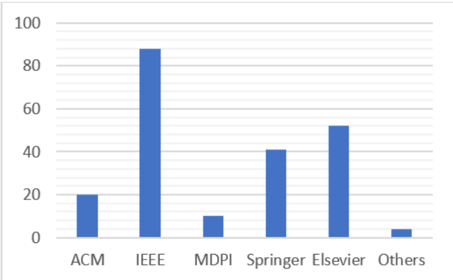

Fig. 1.3 Sources of articles

The papers selected for the study are first searched according to the keywords: Paradigm name + Review. The results were then filtered out to select the latest studies. This produced results for comparison carried out in Table 1.1. Subsequently, more papers were searched and filtered out according to the year published. The keywords used this time were of the following format: Paradigm name + Attribute. For instance, keywords used for the study are: 'Resource Scheduling + Cloud', 'Fault Tolerance + Fog', and 'mobility + Cloud of Things'. Hence the parameters were searched against Cloud, Cloud IoT, Fog and, Fog of Things. In other words, searches resulted in articles from four computing paradigms in a year-wise fashion as depicted in Fig. 1.4. Papers were considered earliest from 2015-2021 for the study. These attributes are specified in section 1.3.

As per the legend of the chart, Cloud and Fog computing appeared in the searches quite more than the rest two paradigms. This shows the advancement and increased involvement of IoT with Cloud and Fog as of 2017. Fog with Cloud IoT on the other hand has benefited from this involvement.

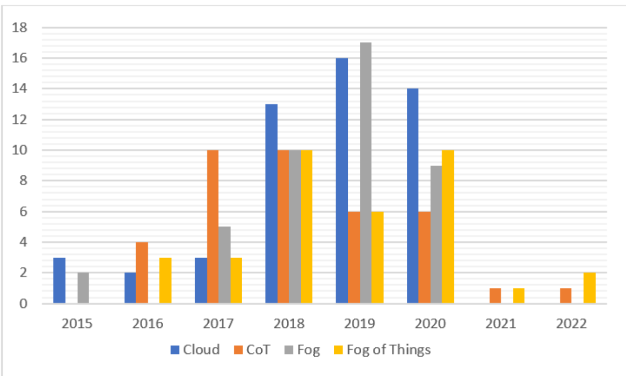

Fig. 1.4: Papers collected on the basis of years

The recent studies on the computing paradigm are based upon the QoS parameters like energy efficiency, resource scheduling, and provisioning, security, SLA, and latency shown in Fig. 1.5. Most of the research is focused on energy efficiency, scheduling, and provisioning approach in cloud computing as well as fog computing. Latency and security are also key issues with cloud computing that are subsequently improved by the fog computing paradigm. Hence, most of the fog computing-based articles focused on these two parameters after the adoption of IoT technology. The areas that would need further study for efficiency are offloading, resilience, mobility, and fault-tolerant. Research in these areas is on the rise with time, as is increased research under fog and IoT frameworks. Hence, the limitations in the studied research articles aid in, for an instance, formulating the various trade-offs that hamper the simultaneous efficient performance of all the parameters.

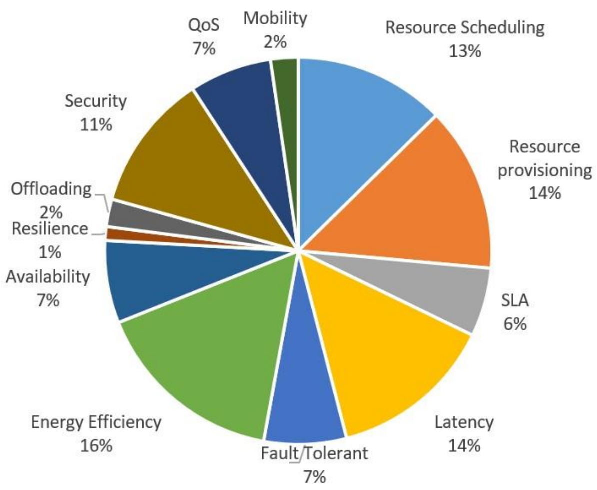

Fig. 1.5: Parameters-wise chart for papers studied

2. CLOUD COMPUTING

Cloud computing as a technology started with the advent of distributed computing and utility computing. It is the collection of pooled resources that allows the provisioning and de-provisioning of resources to the customers in a dynamic fashion [38]. There has been a growing need for on-demand computing power, scalability, huge processing capability, and similar on-computing demands, with time. Many studies have been carried out in the direction of various implementations and integrated applications of the cloud paradigm, all aimed to increase the QoS parameters while maintaining the SLA and other constraints.

2.1 Various services and models of Cloud: There are three basic service of cloud framework as per Fig. 2.1. These are differentiated on the basis of the types of services and resources that can be provided.

Software as a Service (SaaS): SaaS layer is the top-most layer among the leys of abstraction in the cloud environment. It allows the consumer to use the applications, which are running on the cloud infrastructure. The user is able to use the applications without the need of installing the software on the physical resources [39-40]. The user can access the software service from anywhere and anytime from the cloud. Examples are Google apps, and Oracle CRM.

Platform as a Service (PaaS): In PaaS, vendors offer platforms and tools for jobs like deploying a specific software. A scalable environment is provided to help developers for creating and executing their applications. The services can be accessed by the user without knowing the hardware requirements of the intended job. For instance, Google App Engine and Microsoft's Windows Azure offer PaaS.

Infrastructure as a Service (IaaS): In this model, the cloud facilitates on-demand provisioning of the aforementioned resources running on servers. Hence it allows the users the capability to change the virtualized infrastructure. In this level of cloud infrastructure, the users are supposed to manage the software services deployed. For example, Amazon Web Services Elastic (EC2) provides IaaS.

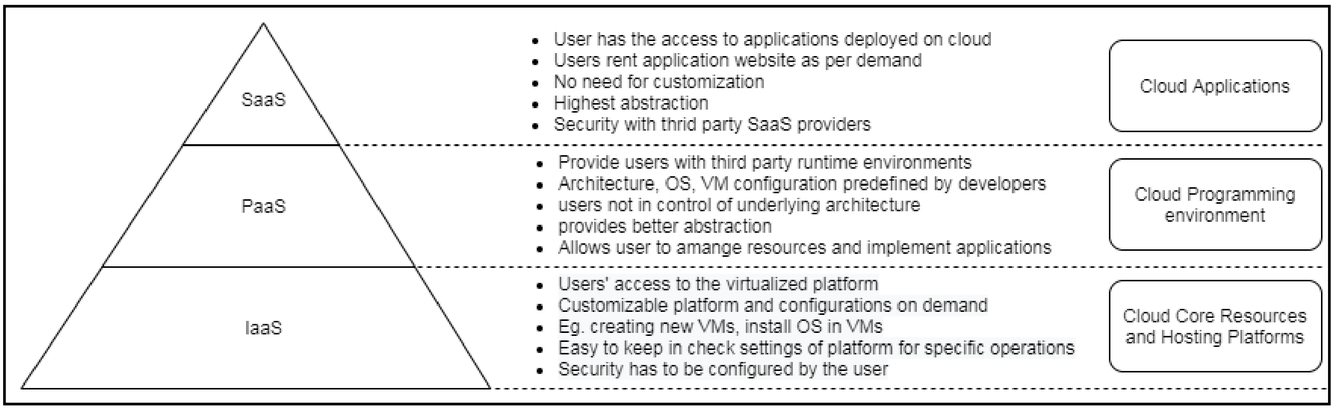

Fig 2.1: Cloud service models

Furthermore, based on the virtual boundaries of the cloud frameworks in different scenarios, there are four types of cloud deployment models explained as follows:

Public Cloud: Public clouds are open to everyone. It is a cloud model available in an on-demand manner for the general public without boundaries. The data centers can be physically distributed throughout the globe for easier access and bigger coverage. Since the public cloud can serve many users without restrictions, multi-tenancy is a characteristic of this deployment model.

Private Cloud: Private clouds are established within an organization, which cannot be accessed by anyone outside the organization. The data within the private cloud does not go outside the framework. Since they are within a group of consumers, the cost is cut less as compared to that of public clouds. And the regulations established in private clouds ensure the security of the data. Also, in public clouds, the control of the infrastructure, regulations and data lies with the service provider.

Hybrid Cloud: Hybrid clouds are designed with the help of private, public, and community clouds. There are a few disadvantages to public and private clouds. For instance, public clouds lack security measures as the data can be shared freely. Similarly, private clouds are expensive to maintain, with the inability to scale. To compensate for the limitations of public and private clouds, the idea of hybrid clouds was conceptualized.

Community Cloud: When an organization deploys all the above-defined types of cloud framework for a flexible framework, it forms a community cloud. For instance, a cloud platform between two or more private organizations would need the security of a private cloud. The standards to be followed by every participant are kept common [40-41]. The idea of community clouds was developed after hybrid clouds. It was developed to cope with the disadvantages of both public and private clouds while catering to the specific needs of a community. Hence the regulation followed will be different among different community clouds.

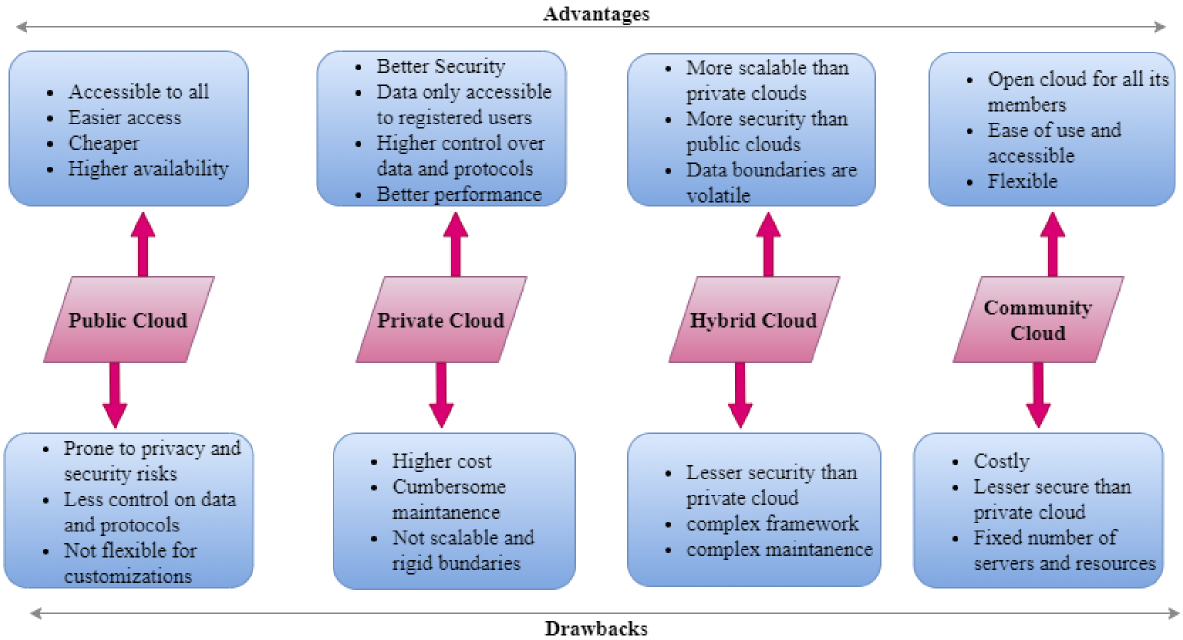

Fig 2.2: Cloud deployment models

2.2 Characteristics of Cloud Computing: Various cloud frameworks provide a wide range of features with subsequent advantages. These frameworks are tailored for different needs, although few common features form the basis of cloud computing's benefits. The basic characteristics of Cloud computing are explained as follows:

On-Demand self-service: Cloud has the ability to allocate the resources to users as per their demand without human interaction. This means that there is no need for the resources to be allocated in a fixed manner. The user is able to customize the resources as well as various computing features without going through a service provider.

Broad network access: Cloud servers are present over a large area in a dispersed manner. This allows users from different regions to access the services.

Resource Pooling: Physical and virtual resources are pooled together for a multitenant model. Multitenancy means that the resources are shared among customers. This pooling of resources from various sources provides cloud scalability, flexibility, and customization power to the user.

Elasticity: The ability of the cloud to provision and release resources on demand, is the elasticity property [38,41. As discussed previously, the cloud lets users access resources as per requirement. This ensures that resource wastage is least, and so is the pricing optimal.

2.3 Applications of Cloud: Although, the cloud offers services to nearly all kinds of user applications in different areas yet some of the extensively applied applications are given below:

Big Data Analytics: Data collected from various IoT devices has been increasing day by day. If the generated data is analyzed with high precision and speed, it would benefit various businesses and services. Hence the need for big data analytics emerged. With help of the Cloud, analyzing big data becomes cheaper and easier [42].

Backup: Data collected and saved on one machine, could not be accessed using a different machine without the ability of backup. Cloud allows backup in such a way that, the data could be accessed from anywhere geographically, given the availability of cloud services in all the involved machines [43].

Disaster Recovery: There are quite a few possibilities that the data saved on a machine might be lost due to some kind of failure. This increases the need for techniques to secure the data in such conditions. Cloud allows the storage of data, in fault-resistant nodes, for the recovery of data [44].

IoT: Internet of Things are referred to the devices which might have processing capabilities and sensors connected over internet. These devices when used along with Cloud, the framework, as a result, is more capable than the Cloud. When these end devices are able to share data over Cloud to a larger framework with higher processing power, another paradigm is formed named Cloud of Things. Cloud architecture provides services like data analysis, storage, processing, and transfer; which is necessary for the collaboration of IoTs spread over a specified framework [45].

Social Network: Existing and emerging social networking sites have gained huge popularity among people because of their ability to connect and communicate with others. Social networks use the cloud for allowing their users to share various contents. A framework for securing data shared in social networks, using the cloud is proposed for checking security issues [46].

Education: Cloud-based knowledge systems are studied, along with their effectiveness for education [47]. Here, the perspective of students was noted for higher education and it was concluded that cloud-based education systems are preferred because of accessibility and ease of sharing. Also, the framework enables storing a large amount of data, and the availability of cloud-based applications.

Healthcare: As an application in healthcare, usually patient's medical history is collected in a decentralized and non-synchronized manner. This raises the issue of analyzing a patient's history by the current medical facility. For this, a cloud-based e-health consultancy framework is proposed, allowing healthcare facilities to record and analyze the medical history of patients in an organized way [48]. When organizations are sporadic in high population areas, this system is quite helpful for as fast medical services to people, as possible.

Agriculture: The use of cloud in agricultural practices is also witnessed where on-demand resource allocation methods are studied [49]. Sensors detect when and how much the resources like electricity, water, and fertilizers would, will be required. This data is then analyzed and stored for future requirements.

2.4 Research Challenges: Studies in different aspects of cloud computing are on the rise. The Cloud computing framework has the ability to share resources using various algorithms (static or dynamic), as per the demand. The main aim of these algorithms is to allocate the resources to each job, fulfilling the requirements of end-users and, minimizing the excessive expenditure [41]. Similarly, algorithms try to optimize the resource provisioning along with other challenges like security and resource portability. Optimizing a set of parameters at times leads to a trade-off for other parameters' performance. For instance, increasing availability might lead to increased energy consumption. The provision of third-party resources introduces concerns regarding security measures. Resource management requires effective handling with the least wastage of resources. Such challenges are discussed as follows*:*

• **Security and privacy:** Cloud allows the users to carry out their work on third-party resources pooled from various vendors and infrastructures. Carrying out critical operations in third-party resources increases security threats. Sensitive information accessed by the cloud can be tampered with. A novel DNA-based data encryption scheme is proposed that increases security by implementing a 1024-bit long password and reduces the system overhead [50].

• **Resource Provisioning:** Resources like processing power, RAM and network bandwidth, needed to be provisioned by the cloud service providers to the clients as per their demands. The challenge of an algorithm is to provide resources optimally as per the near exact requirement of the consumers. The aim is to ensure that resources are neither left idle nor get overloaded. Research has been conducted in this field to be able to predict consumer demands for better and optimal provisioning. One such algorithm is able to predict future needs and customer demands, along with features of networks and processes [56].

• **Resource Scheduling:** Resource scheduling algorithms usually optimize proposed frameworks, while some QoS parameters' performance might be traded off. The objective of these algorithms is to ensure that the cloud service provider intelligently schedules the resources without the violation of SLA and with optimized QoS values. Research needs to be carried in any direction where allocating resources optimally compromises other parameters.

• **Energy Efficiency:** The servers, resource infrastructures, and databases are supposed to be kept running all the time for availability. This feature of the cloud is expensive, in monetary terms. Increasing generated data requires an increase in cloud frameworks. CO_2 emissions are increasing with the cost. For this, an approach for effective allocation of VMs to physical machines is proposed, using energy efficiency and request acceptance ratio to design fitness function [69]. The proposed approach was able to perform better in terms of both energy efficiency and request acceptance ratio.

• **Availability:** The user expects to access the data and resources available on the cloud, from anywhere on the globe at any given time. There is a possibility that the services become unavailable due to some failure. These cases of downtime of services might hamper the ongoing processes. Hence provider is expected to implement efficient algorithms to make services accessible without threats. It is also expected of the provider that the data on different data centers gets fetched without the fear of failures in its connectivity. Availability-Aware container scheduling is proposed by selecting VMs with higher availability values [77]. This proposed framework is able to provide better availability and higher security.

• **Fault-Tolerant:** For servers running continuously, any failure in the nodes might cause the information, data or transactions to be lost. To ensure that the computation continues even during any such situation, studies have been done for designing algorithms to induce fault-tolerant behaviour in the framework. An improved Dynamic Fault Tolerant Management (DFTM) algorithm is proposed using Load monitoring and balance mechanism [71]. This proposed algorithm-maintained resource availability, cloud survivability, and minimized complexity.

• **Latency:** Real-time applications require processing without any delay. Although the cloud paradigm might not be able to provide the service to such types of applications because requesting services can run anywhere in the globe and increase the latency. Hence, a need arises for other technologies to bridge this distance between cloud servers and nodes that require real-time results.

• **Cost management:** The objective of the end-user is to get the best service from the cloud within budget while service providers try to gain the maximum profit from the infrastructure. Hence, the cost is required to be managed as a parameter for the algorithms to be optimal.

2.5 Existing solutions to overcome the mentioned challenges: Table 2.1 enlists and compares recent research trends in an organized manner. The table is organized in a way such that the studies are categorized according to the parameters they are aimed at. After this organization, all the studies are compared on the basis of:

- the algorithms used
- techniques followed for achieving the aim
- results the studies were able to achieve
- limitations or challenges the frameworks face.

This sheds light on the different mechanisms and algorithms used and the efficiency of the frameworks. We can classify the existing algorithms in three ways: heuristic, meta-heuristic, and hybrid [34]. Heuristic algorithms are designed to solve a problem with specified scenarios. Although this solution doesn't cater to the rest problems in the same area. Hence, while heuristic algorithms can solve specific problems with higher efficiency, their performance in the remaining situations would be reduced. Instead, metaheuristic algorithms are not designed for only one specific problem. They can be applied to various situations with improved performance. Therefore, heuristic and metaheuristic algorithms are combined to design a hybrid algorithm. Techniques employed along with the aims achieved are compared along with each article's challenges.

There have been quite a few researches in the area of resource availability for cloud environments. Swarm Intelligence Based Prediction Approach by Integrating Autoregressive Integrated Moving Average (ARIMA) and Multiple Support Vector Regression (MSVR) models and feature selection approach based on Kernel Adatron (KA) algorithm and Particle Swarm Optimization (PSO) was proposed [56]. Future demands of consumers like, memory, and network resources are predicted. Although the challenge this framework faces is that it can only be implemented for old consumers since it would be difficult to predict for new consumers with no past data. In another work, a Dynamic resource provisioning algorithm that employs Shared-File aware Resource provisioning algorithm was proposed [59]. Reduced resource consumption was observed with the trade-off of not considering privacy and security for the framework. A policy management technique based on a Blockchain management host was implemented for speedy and secure policy management, although VM migration consumed more time as compared to conventional algorithms [60]. The aforementioned algorithms achieve their results with the trade-off of energy efficiency.

ThermoSim is proposed by employing a Thermal aware and utilization-based approaches resource management framework to model thermal aware characteristics of cloud data centers [58]. But, to be able to distribute uniformly the temperature, there is no thermal management strategy. An Energy-aware fault-tolerant dynamic task scheduling (EFDTS) algorithm is proposed with a fault-tolerant mechanism [69]. Fault tolerance is reduced as a result, along with reduced response time. But it is assumed that at most one host might fail at a time, and for subsequent failures, there is no remedy proposed. Particle Swarm Optimization Based Scheduling Technique BULLET is proposed to search and map resources to cloud workload, according to customers' requirements [52]. As a result, execution time, cost, and energy are reduced. But as the workload increases, the reliability of the framework starts declining. In another study, a three-dimensional virtual resource scheduling method for energy saving framework is proposed by dividing virtual resource scheduling into three stages [55]. The framework is able to reduce energy consumption and also SLA violations. Both the mentioned algorithms result in efficient resource scheduling at the cost of not considering security and privacy. Further research is conducted to remove the compromises between QoS parameters and load balancing [89-92].

Table 2.2 enlists the QoS parameters that are considered by proposed algorithms and frameworks. It can be observed that none of the studies has considered every parameter in their frameworks. For instance, it can be deduced from the comparison as to which algorithms have considered optimizing energy efficiency while increasing availability and latency. The QoS parameters used to compare the algorithms' workings are explained in Section 1.4. In Table 2.3, the algorithms are compared as to which of them the constraints are taken into account. The constraints are as follows:

i. Priority constraint: The first constraint mentioned in the table is the priority. Jobs and resources might have been requested in order of priority. The algorithms proposed are checked if they considered priority as a parameter in their execution.
ii. Deadline Constraint: Deadline constraint is defined by the time that a task has to be executed within. This property is a must while dealing with real-time jobs.
iii. Execution Environment Constraint: There are two types of execution environments where the job could be executed: real and simulation. The algorithms are checked against the type of environment used for execution. Real computing frameworks work in real-time. While, in simulators, the computing framework is simulated or synthesized to implement and analyze the proposed algorithms' working before moving to the real environment. Implementation in real environments ensures that the algorithms work efficiently. Whereas if executed on a simulator, the real-time situations and obstacles cannot be predicted.
iv. VM specification analysis: A virtual machine's specifications like RAM and processing power are considered constraints. This is to check if the algorithm needs a predefined set of constraints of the aforementioned resources, where the efficient results were obtained.
v. Static/Dynamic: This constraint defines the type of data to be used in an algorithm. Static data are provided to the framework before the execution. Whereas if data is provided to the framework at the time of execution, it is called dynamic data. Hence this constraint defines the type of data to be used for the specific algorithms.

These very constraints are used to compare the algorithms in further sections as well.

2.6 Limitations of Cloud: Despite the features provided by the cloud, there are some limitations of the framework. Tasks can be sent to the cloud for processing in real-time. But as cloud servers and resources are placed sporadically, these tasks might not be processed without delay. Also, these servers are needed to be running continuously to provide services at any time. This results in increased energy consumption. In view of this, the limitations are discussed as follows:

• **Latency:** Real-time jobs require responses from the processor as quickly as possible. This drawback gave rise to the idea of the processing being done closer to the sensors. The Cloud isn't generally placed closer to nodes, but sporadically. This requires higher time latency for processing. IoTs require real-time processing for data provided by them, which requires another paradigm to operate with negligible latency. This hence is a limitation of the cloud to not be able to process within the required time frame [83,86].

• **Security:** Cloud environment uses third-party resources for carrying out jobs. IoTs connected to the cloud for data processing would be sending their sensitive data over to the cloud. If this data is attacked or tampered with, IoT safety could also be attacked. Therefore, better security measures or another framework is required to work with Cloud to compensate the drawback created because of using party resources [84].

• **Energy Efficiency:** Cloud data centers consume a huge amount of energy. It is so because they are needed to be run 24*7. This is in turn required for virtual continuous service of the cloud and the ability to allow users to request services at any time. Also, the need to replicate data for achieving fault-tolerant increases the energy consumption [80]. There have been studies in energy-aware algorithms which attain optimized consumption of energy while compromising other QoS values [81-82, 86,88].

• **Network bandwidth:** The continuous communication between cloud data centers, VM and consumers, can sometimes overwhelm the provided network bandwidth. In such a situation of high-power computing, there is a need for continuous and sufficient bandwidth to transfer the information [85, 87].

• **Internet Accessibility:** Since the cloud framework consists of interconnected servers and databases, the internet is crucial for working of this paradigm. If there is network disturbance or failure, data in the transaction is lost. There are some approaches to cope with the situation by maintaining the copies and fault-resistant nodes. Still, there is a need for more research in this area which will also aid in coping with the cost of maintaining several copies of the same data.

2.7 How to resolve limitations of cloud using existing techniques: IoTs require processing in real-time for the data generated from their devices. Cloud cannot provide real-time processing to some applications because data has to reach cloud data centers first and then they get scheduled as per a specified scheduling technique. This approach takes time and it cannot be afforded for latency-sensitive applications. The heterogeneity of end devices needs a framework compatible with all the end devices in a system. Also, the required paradigm should be secure enough unlike cloud computing where the involvement of third-party resources hampers the security. Along with heterogeneity, latency, and security, there also is the issue of energy consumed by the servers and resources running continuously. This increased energy consumption not only affects cost management and the energy consumed but also CO_2 production. Cloud is hence not an optimal technology for such various situations. Therefore, it is required to employ IoT in another paradigm for secure processing.

TABLE 2.1: SUMMARY OF PROPOSED ALGORITHMS ALONG WITH LIMITATIONS

Parameters	Year	Algorithm	Technique	Results	Limitations/challenge
Resource Scheduling	Dewangan et al., 2019[51]	Self-optimized energy efficient strategy	To improve the cost and energy consumption.	Cost effective, fault tolerant, reduced SLA violation	SLA violation is removed partially.
Fault Tolerant	Gill et al., 2018[52]	BULLET	PSO based scheduling technique	Improved execution time, cost, energy and other QoS parameters	Reliability decreases with increasing the workload
	Malarvizhi et al., 2020[53]	CRSOH	Resources allocated on basis of energy	Optimized accuracy, power consumption	Each node is required to have same configuration
	Priya et al., 2019[54]	F-MRSQN	Fuzzy Square Inference and a queuing network model is implemented	Improved average success ratio and reduced response time	Security and privacy are not considered
	Zhu et al., 2017[55]	Three-dimensional virtual resource scheduling method for energy saving	MVBPP based heuristics virtual resource allocation.	Reduced energy consumption and SLA violations	Security and privacy are not considered
Resource Provisioning	Kholidy, 2020[56]	Swarm Intelligence Based Prediction Approach	Integrated ARIMA and MSVR models, Kernel Adatron algorithm and PSO	Predicts memory, disk storage, network resources	The scope of the SIBPA is hinged on the IaaS model only
	Kim, 2018[57]	DC resource provisioning scheme	New resource allocation algorithm using machine game model and Mood value	DC resource usability, cloud service success ratio improved.	VM migration among data centers is not considered.
	Gill et al., 2020[58]	Thermosim	Thermal aware and utilization-based approaches resource management framework	Simulate and model thermal aware cloud data centers	No thermal management for uniform distribution of temperature
	Tuli et al., 2020[59]	Dynamic resource provisioning	Shared-File aware and dynamic Resource provisioning algorithm	Lower number of deadline violations, resource consumption.	Privacy and security are not considered
	Uchibayashi et al., 2019[60]	Policy management technique	Policy management technique based on blockchain	Speedy and secure policy management comparable to conventional methods	VM Migration process takes more time than conventional algorithms
SLA	Li et al., 2019[61]	Energy-Efficient VM Consolidation	Host overloading/underloading detection algorithm and a new VM placement algorithm	Reduced energy consumption by 25.43%, SLA violations by 99.16%	Other resources like RAM, storage is not considered.
	Liu et al., 2020[62]	CloudSec framework	Cloud behavior model using FSM; cloud security SLA mode SecSLA model based on temporal logic	Allows checking if cloud services meet SLA requirements	State explosion (since FSM used in model)
	Mandal et al., 2020[63]	Energy aware VM selection policy in green cloud computing	Framework based on Local Regression Robust method	Energy efficient VM selection policy, reduced SLA violations	Not yet deployed on practical cloud environment
	Wang et al., 2020[64]	SLA-aware resource scheduling algorithm	OpenStack scheduling module-based SLA aware scheduling algorithm	Decreased SLA violation rate, cost of CSPs reduced	Not dynamic approach
Latency	Guo et al., 2018[65]	VM-Shadow: A system to transparently and dynamically manage VMs	VM-Shadow employs WAN-based live migration and a new network connection migration protocol	Optimized location and performance of VMs	Since nested hypervisor is used, flexibility is reduced
	Li et al., 2019[66]	Numerical method for relevant stationary response time distribution	Predict the stationary response distribution of a time-critical service with a Poisson arrival process	Enabled service operator to provision CPU resources for a periodic service	Assumed that a job arrival after slot boundary
	Rodrigo et al., 2019[67]	Elastic Switching Mechanism for data stream processing	Mechanism based on homomorphic encryption (HomoESM)	Improved latency	Applicable only for data streams with finite and unchanging data
	Naghshnejad & Singhal, 2018 [68]	Fixed Multiple Kalman Filter and, Multi-Layer Kalman Filter	Prediction is achieved by modelling applications runtime series as a state space model	Improved prediction, reduced waiting time	Applications are assumed non-pre-emptive
Energy Efficiency	Armstrong et al., 2017 [73]	Energy-efficient interoperable cloud architecture	Implementation on an architectural component Virtual Machine Image Constructor	Energy optimized, VM images creation automated	CPU utilization of the VMIC tool is mainly limited to 1 core
	Zhang et al., 2019 [74]	Energy aware VM allocation	Novel fitness function based on instruction-energy ratio	Better energy efficiency	VM rejection is not considered
	Kumar & Sharma, 2018 [75]	PSO-COGENT	Formulated multi-objective scheduling problem mathematically; Modified PSO algorithm	Reduced execution time, execution cost, task rejection ratio, energy consumption	Reliability, availability, response time, are not considered
	Mishra et al., 2018 [76]	Energy-aware Task-based Virtual Machine Consolidation	tasks are classified according to their resource requirement then allotted VM on a PM	Energy consumption reduced, minimised make-span and task rejection rate	Make-span of the system worsens with lesser number of input tasks.

Availability	Alahmad et al., 2018 [77]	Availability-Aware container scheduling	Strategy selects VMs and hosts that have higher availability values	Higher service availability levels	DC assumed to have enough resources
	Londhe et al., 2018 [78]	DROPS	File is fragmented into pieces and replicated at strategic locations within cloud	Better availability, and security	Replication factor assumed 0
	Mengistu et al., 2018 [79]	Availability and Reliability prediction model	Based on multi-state semi-Markov process	Predict future availability,reliability, increased accuracy	Energy not considered while replicating
	Hassanzadeh-Nazarabadi et al., 2016 [80]	Decentralised availability aware algorithm named Aware	Using availability vector in churn model	Increased availability of replicas	It is energy consuming to frequently update links

TABLE 2.2: COMPARISON OF ALGORITHMS ON THE BASIS OF QOS PARAMETERS

	Name	**ET**	**MST**	**Reliability**	**Authentication**	**SLA-V**	**EC**	**Throughput**	**RM**	**Security**	**FT**	**Availability**	**Latency**	**RT**
Resource Scheduling	Dewangan et al., 2019[51]	yes	no	no	no	yes	yes	no	yes	no	yes	no	no	no
	Gill et al., 2018[52]	yes	no	yes	no	yes	yes	no	yes	no	yes	yes	yes	no
	Malarvizhi et al., 2020[53]	no	yes	no	no	no	yes	no	yes	no	no	yes	no	no
	Priya et al., 2019[54]	no	no	no	no	no	no	no	yes	no	no	no	yes	yes
	Zhu et al., 2017[55]	no	no	no	no	yes	yes	no	yes	no	no	no	no	no
Resource Provisioning	Kholidy, 2020[56]	no	no	no	no	no	no	yes	yes	no	no	yes	no	yes
	Kim, 2018[57]	yes	no	yes	no	no	no	no	yes	no	no	no	no	no
	Gill et al., 2020[58]	yes	no	no	no	yes	yes	no	yes	no	no	no	yes	no
	Tuli et al., 2020[59]	yes	no	no	no	no	no	no	yes	no	no	no	yes	yes
	Uchibayashi et al., 2019[60]	yes	no	no	yes	no	no	no	yes	yes	no	yes	no	no
SLA	Li et al., 2019[61]	no	no	no	no	yes	yes	no	yes	no	no	no	no	no
	Liu et al., 2020[62]	no	no	yes	yes	yes	no	no	yes	yes	no	yes	no	yes
	Mandal et al., 2020[63]	yes	no	no	no	yes	yes	yes	yes	no	no	yes	no	yes
	Wang et al., 2020[64]	no	no	no	no	yes	yes	yes	yes	no	no	no	no	no
Latency	Guo et al., 2018[65]	yes	no	no	no	no	no	yes	yes	no	no	no	yes	yes
	Li et al., 2019[66]	yes	no	no	no	no	no	no	no	no	no	no	yes	yes
	Rodrigo et al., 2019[67]	no	no	no	no	no	no	yes	no	yes	no	no	yes	no
	Naghshnejad & Singhal, 2018 [68]	yes	no	no	no	no	no	no	no	no	no	no	yes	no
Fault Tolerant	Armstrong et al., 2017 [73]	yes	no	no	no	no	yes	no	yes	no	yes	no	no	yes
	Zhang et al., 2019 [74]	no	no	yes	no	no	yes	yes	no	no	yes	yes	yes	no
	Kumar & Sharma, 2018 [75]	yes	no	yes	no	no	no	yes	yes	yes	yes	yes	yes	yes
	Mishra et al., 2018 [76]	yes	no	yes	no	no	yes	no	yes	no	yes	no	no	no
Energy Efficiency	Dewangan et al., 2019[51]	yes	no	no	no	no	yes	no	yes	no	no	no	no	no
	Gill et al., 2018[52]	yes	no	no	no	no	yes	no	yes	no	no	no	no	no
	Malarvizhi et al., 2020[53]	yes	yes	no	no	no	yes	yes	yes	no	no	no	no	no
	Priya et al., 2019[54]	yes	yes	no	no	no	yes	no	yes	no	no	no	no	no
Availability	Alahmad et al., 2018 [77]	no	no	no	no	yes	yes	no	yes	no	no	yes	yes	no
	Londhe et al., 2018 [78]	no	no	no	no	no	no	no	no	yes	no	yes	no	no
	Mengistu et al., 2018 [79]	no	no	yes	no	no	no	no	no	no	yes	yes	no	no
	Hassanzadeh-Nazarabadi et al., 2016 [80]	no	no	no	no	no	no	no	no	no	no	yes	no	no

TABLE 2.3: ALGORITHMS COMPARED ON THE BASIS OF CONSTRAINTS

	Name	Priority constraint	Deadline constraint	Execution (simulation/real env) constraint	VM specification analysis (RAM, processing power)	Static/ Dynamic
Resource Scheduling	Dewangan et al., 2019[51]	yes	no	simulation	yes	dynamic
	Gill et al., 2018[52]	no	yes	simulation	no	dynamic
	Malarvizhi et al., 2020[53]	no	no	simulation	yes	dynamic
	Priya et al., 2019[54]	no	no	simulation	no	dynamic
	Zhu et al., 2017[55]	no	no	simulation	yes	dynamic
Resource Provisioning	Kholidy, 2020[56]	no	no	simulation	no	dynamic
	Kim, 2018[57]	no	no	simulation	yes	dynamic
	Gill et al., 2020[58]	no	no	simulation	no	static
	Tuli et al., 2020[59]	no	yes	real	yes	dynamic
	Uchibayashi et al., 2019[60]	no	no	simulation	no	static
SLA	Li et al., 2019[61]	no	no	simulation	yes	dynamic
	Liu et al., 2020[62]	no	yes	simulation	yes	dynamic
	Mandal et al., 2020[63]	no	no	simulation	yes	dynamic
	Wang et al., 2020[64]	no	yes	simulation	no	static
Latency	Guo et al., 2018[65]	yes	no	real	yes	dynamic
	Li et al., 2019[66]	yes	yes	simulation	yes	static
	Rodrigo et al., 2019[67]	no	no	real	no	static
	Naghshnejad & Singhal, 2018 [68]	no	no	simulation	no	static
Fault Tolerant	Armstrong et al., 2017 [73]	yes	yes	simulation	yes	dynamic
	Zhang et al., 2019 [74]	no	yes	simulation	no	static
	Kumar & Sharma, 2018 [75]	no	no	simulation	yes	dynamic
	Mishra et al., 2018 [76]	no	yes	simulation	yes	dynamic
Energy Efficiency	Dewangan et al., 2019[51]	no	no	simulation	no	dynamic
	Gill et al., 2018[52]	no	no	real	no	static
	Malarvizhi et al., 2020[53]	yes	yes	simulation	yes	dynamic
	Priya et al., 2019[54]	no	no	simulation	yes	dynamic
Availability	Alahmad et al., 2018 [77]	no	no	simulation	no	static
	Londhe et al., 2018 [78]	no	no	simulation	no	static
	Mengistu et al., 2018 [79]	no	no	real	yes	static
	Hassanzadeh-Nazarabadi et al., 2016 [80]	no	no	simulation	no	dynamic

3. CLOUD OF THINGS (CoT)

The devices that have the capability to connect to the internet could be part of the Internet of Things (IoT). This allows the smart devices to compute, transmit, store, and analyze the data while sharing data on the network. When these devices are connected to the internet, the geographically separated IoT nodes can share data among them. For instance, sensors from traffic lights could be collected in real-time to create a map of the traffic quantity in a specific area. Likewise, data can then be either collected in an organized manner or analyzed together to deduce the information required. The presence of the Internet and processing elements would help in the collective working of smart devices. Cloud allows the pooling of resources and servers as per requirement Cloud framework is required to help operate IoT beyond its own finite computational property. These edge devices are either mobile or stationary machines that are geographically sparse. When the cloud paradigm is integrated with IoT, the large and flexible computing power is brought closer to the edge of the network.

The distributed interconnection of edge devices increases the devices' storage and computing capacities. In turn, it helps the cloud in managing real-time big data collected from smart devices [93]. The rising extensive use of IoT devices in various applications on large scale like health, automation, and industries, needs integration of Cloud architecture for easier management of smart devices and analysis of data on large scale in geographically distributed fields.

3.1 Architecture: The Cloud-IoT architecture has the ability to get dynamically modified and adapted as required by the situation. To achieve small healthcare, research was conducted and a Cloud-IoT-based sensing service (HM-SS) was proposed [94]. This algorithm was able to improve service quality and accessibility among small healthcare facilities. Similarly, in another study, an architectural model was proposed to integrate Cloud and IoT [95]. This framework was able to implement smart factories with geographically dispersed devices. The underlying architecture of Cloud IoT for this model comprises three components: IoT/sensing layer, network layer, and application layer [93]. The architecture is explained as follows:

i. The sensing layer is responsible for the collection of data which is further transmitted using the network layer of the Cloud to connect geographically separated IoT devices.
ii. The network layer is responsible for the transmission of data and information in an efficient way so that the bandwidth isn't overloaded.
iii. The various application services are provided by the application layer.

This basic architecture ensures the incorporation of computing and flexible features of the cloud in IoT. If required for a field, this architecture could be customized as per the need.

3.2 Features of CoT: Cloud of things is comprised of smart devices connected to Cloud services. This implies that the storage, processing capability, and coverage of the framework are increased drastically. Scattered smart devices now work together, sending the data to be stored and with higher computing requirements to the cloud. This allows the cloud to be able to collect data aka Big Data. And as smart devices are mobile in nature, cloud services turn mobile as well.

Storage: IoT devices have the ability to process information in real-time, but they are not capable of storing huge amounts of data produced. On the other hand, the cloud has the ability to store and analyze big data with the help of its powerful servers. Hence, this data is transmitted to the cloud for storage, which can be received by smart devices as per the requirement.

Processing power: IoT devices provide real-time processing, that can be hampered when enormous data is to be processed. Cloud offers flexible and high processing power as required. When the Cloud is integrated with IoT, the processing capabilities of the entire framework are more than that of IoT.

Coverage: IoT is interconnected smart devices which communicate with each other. The smart devices might be mobile. With the help of these mobile sources, the cloud is able to collect data from a wide range of devices. With the inclusion of the cloud in the frame, edge devices of huge geographical areas can be interconnected along with the powerful resources of the cloud.

QoS: Cloud's framework allows features like availability and scalability, for a better user experience. IoT paradigm can't assure these parameters. With the introduction of the Cloud and its resources in the IoT framework, these various QoS parameters can be achieved.

Big Data: Information processed by edge devices cannot be collected by the IoT framework itself due to limited storage and processing power. When the cloud is incorporated with this framework, information produced by these devices can be transmitted to the cloud for storing and analyzing big data. It increases the information as well as coverage in the cloud paradigm [93].

3.3 Applications of CoT: Cloud brought processing capabilities to quite a lot of applications. Coverage of the cloud is extended to more areas, with the help of IoT devices connected. Integration of smart devices with the Cloud opens up various possibilities and applications:

Healthcare facilities: With the help of smart devices tracking the patients' current medical status, the patients themselves and concerned facilities can lower the response times to any caused emergency. Such individual smaller healthcare facilities can be brought together to manage all their data together for better accessibility and services [94]. This data becomes resistant to failures and reduces the requirement of staff.

Automated industries: Various devices, either smart or automated, are present in an industry. Automation of such devices for seamless processing brings predetermined results with the least staff requirement. It also allows surveillance in place of production. Such integration and automation of legacy devices in industries are discussed in Industry 4.0 [95].

Smart Home: Automating various tasks of a home-like audio system, lights, news and surveillance, would aid in improving the everyday experience. The smart devices collect information from daily activities to learn and make the user experience as smooth as possible. A multi-layer Cloud-IoT architecture for effective communication among the devices is implemented considering security as an important parameter for automation [97].

Smart grid: Traditional electrical grids provide electricity as per the predetermined need of the area. The aim is to provide more electricity than required so that there is no shortage, with the least expenditure and no wastage. Although when the need of users is lesser than the produced electricity, wastage ensues. Automating the electrical grid using IoT and Cloud can reduce this wastage of electricity by allowing two-way communication between consumers and service providers with the help of sensors. A similar Cloud-IoT architecture is proposed for ensuring secure data transmission, in real-time [98].

Smart traffic system: Predetermined traffic control system is the traditional method used and followed in most places as of present. This system does not take into account the real-time traffic density, resulting in congested traffic. This need for real-time adaptive traffic control has motivated research in the same direction. Framework for Smart Traffic control is designed with AWS IoT which uses many sensors in a specific area to achieve real-time information about traffic [99].

Smart farming: Agriculture requires manual control and work, on a fixed schedule. This could lead to wastage in quite many situations. For instance, water and electricity are used for the irrigation process, which can lead to wastage if faulty equipment is used. Resources like water, manure, and manpower could be wasted if not overlooked. A framework is designed for smart farming, on the basis of various field parameters like pH value, soil moisture, and other parameters [100]. The proposed architecture can predict and hence automate the usage of water and electricity, without any wastage.

Blockchain: The technology named blockchain was developed to maintain a ledger of transactions which is accessible to desired nodes, maintaining security in a decentralized manner. In CoT, the end devices work along the Cloud servers in a decentralized manner. There are studies carried out to ensure security in similar frameworks. Incorporating blockchain in CoT helps in overcoming the challenges faced by the paradigm. This amalgamation of the two technologies is studied as BCoT [208]. This framework along with similar methods of security like Trusted Execution Environments is discussed by various studies [211].

Artificial Intelligence: With the use of neural networks and learning algorithms, most of the processes can be made automated. Sensors collect the data from end devices, which are then analysed. These algorithms are implemented in such a way that the system adapts with the changing environment [212]. Incorporating artificial intelligence aids the framework in reducing costs because of the reduced demand for resources.

3.4 Challenges and Issues: Integrating the Cloud with IoT has quite a lot of advantages. The various IoT devices generate data in a heterogeneous manner. Bandwidth and security measures, both are required to be standardized and optimized for the various data, their sources, and the channels. Although, this framework is also susceptible to issues as explained:

- **Big Data:** Smart devices are good for collecting and processing data on the edge, in real-time. However, these devices are not capable of managing a huge amount of data. Data collected by a large number of IoT devices can amount to big data, which again requires this data to be sent to the cloud for further processing. In case, the analysed big data has to be transmitted back to IoT, it would surpass the capability of the IoT devices [93,96].

- **Heterogeneity:** The IoT devices are usually heterogeneous in nature. Even in one framework, transmitting data to geographically scattered smart devices of heterogeneous nature needs resources like VMs and networks of sufficient standards [93,96]. Comparatively more than cloud, fog standards are capable of managing a few heterogeneous smart devices mapping such scenarios.

- **Security:** Cloud allows third-party resources and users to access the cloud services. IoT devices might transmit sensitive information to the cloud and vice versa. If IoT devices are used in the paradigm, they would be open to security attacks from insecure networks [96]. The end devices should be authorized and authenticated for a secure environment [209, 210]. A new security framework would be required for secured computing.

- **Bandwidth:** Seamless data transfer between Cloud and IoT devices requires sufficient bandwidth. Transmitting data over the network for the huge number of IoT devices to the Cloud over a congested network might result in slow transmission, which then can affect the real-time processing of IoT [93].

• **Resource Provisioning:** An appropriate resource provisioning algorithm would be required to achieve bandwidth utilization, without overwhelming, not letting it get idle [212]. At the same time, time sensitivity should also be considered to maintain the real-time processes. Managing trade-offs between parameters is crucial.

• **New applications**: IoT networks have heterogeneous devices. An application would not be compatible with different IoT devices. Therefore, for these different types of smart devices, diverse and compatible applications are required [93]. Design of new applications for different types of IoT devices is required.

3.5 Proposed solutions*:* Recent research on the cloud-IoT framework is studied, enlisted, categorized, and analyzed in Table 3.1. The techniques and methods adopted for designing the proposed algorithms are mentioned along with discussed shortcomings. The format the table follows is as in section 2.5. The studies are organized according to the parameters worked upon. Following that, each study can be read according to the algorithms proposed along with the techniques used. This is then followed by the results achieved and the challenges the frameworks still face. This format on whole presents a precise but self-explanatory review of the studies in the field. A few of them have been discussed in the following paragraphs. The trade-offs in respect to this, are discussed in section 3.6. The information about how improving one parameter is affecting another's efficiency is studied along with the new techniques. After the analysis of the articles in Table 3.1, the algorithms are compared against the parameters which are worked upon in each framework in Table 3.2. These parameters are discussed in Section 1.4. As it can be observed, every parameter is not considered for the algorithms. This is where the trade-offs are observed and mentioned in the following paragraphs. In Table 3.3, all the constraints of the algorithms are mentioned. These constraints are explained in section 2.5. From these constraints and parameters considered, it is understandable as to which algorithms and frameworks produce efficient results under specific constraints.

An algorithm named LOTEC is proposed and implemented for optimizing energy consumption with the help of green energy along with LYAPUNOV optimization [101]. The only limitation of this technique is that Green energy is unstable and costlier than conventional energy. Another Energy-Aware allocation algorithm is implemented with the help of a dynamic voltage and frequency scaling algorithm, although the mobility of IoT devices is not considered [102]. Similarly, a Unit slot optimization online algorithm is proposed on the basis of LYAPUNOV optimization [104]. The framework is able to achieve cost-effective and delay-sensitive implementation. The communication overheads for smaller messages are not considered.

For improving SLA as a parameter of the services, an event-driven based SLA violations' predictions approach is implemented by separating the framework into three modules [112]. SLA violations are predicted efficiently, but there is the possibility of false-positive predictions as well. In another article, smart gates are used to propose Smart city modeling (Sii-mobility) project for transportation, although expertise is required to implement and operate the architecture [116]. Hence this framework cannot be implemented by a user without prior knowledge.

3.6 Limitations: The challenges of processing and managing IoT devices in real-time still persist. Cloud servers do increase the capabilities of otherwise smart devices with limited capacities. Also, the cloud collects data from over a wide geographical area, but this framework is still unable to resolve other limitations as follows:

• **Latency:** Smart devices need latency-free operations performed on the cloud. It should be noticed that cloud servers are not placed in proximity to the devices and they are responsible for managing tasks from a dispersed range of areas and devices. It means that transferring data and requests from IoT to Cloud and Cloud to IoT will still need more time. This results in delayed operations in cases where real-time results would be required.

• **Cost:** Active involvement of the cloud in this framework implies that the servers would be required to be running constantly. The cost of running cloud servers and resources for sporadic jobs of the edge devices is more than localized nodes of distributed servers and resources for the tasks.

• **Security:** Tasks and information transmitted by edge devices to the cloud face the possibility of privacy issues and security challenges [117-119]. Tasks and data transferred to the cloud are forwarded to the third-party resources. The IoT devices are hence susceptible to attacks. Also, the heterogeneity of smart devices is more than usually not supported by standard security measures. There needs to be sufficient standards for preventing security issues in the Cloud IoT framework.

• **Bandwidth**: For the data and information transmitted between cloud and IoT devices, the nodes share common paths. Hence in the case of every node transmitting data, it onsets a possibility of reduced efficiency if every device is not allocated specific bandwidth [120]. Studies need to be carried out hence to ensure that the tasks and information are allocated bandwidth as per a certain criterion.

TABLE 3.1: SUMMARY OF ALGORITHMS PROPOSED ALONG WITH LIMITATIONS

Parameters	Name	Algorithm	Technique	Results	Limitations/challenge
Energy Efficiency	Nan et al., 2017 [101]	LOTEC	Cost and time optimized using LYAPUNOV Optimization	Time and cost optimized	Green energy can be unstable
	Mahmoud et al., 2018 [102]	Energy Aware application allocation	Used dynamic voltage and frequency scaling algorithm	Reduced energy consumption, round trip time	Mobility of IoT devices is not considered
	Ning et al., 2019 [103]	GSNVE framework	Heuristic algorithm used to solve optimization problem	Energy efficient	Security and privacy compromised
Latency	Nan et al., 2018 [104]	Unit slot optimization online algorithm	Algorithm based on LYAPUNOV optimization	Cost effective, delay sensitive framework	Arrival and service rates should be known prior
Resilience	Khan et al., 2019 [105]	Emergency and disaster recovery system: RESCUE	Broker-messaging server used for exchange of data	Resilient system, better load balancing	Mobility is not considered
Offloading	Hasan et al., 2018 [106]	Localized IoT based cloud computing model Aura	Score an IoT device based on its performance	Local and scalable computation	Security is not considered
	Jia et al., 2019 [107]	STOFDM	Truncate orthogonal FDM signal in time domain	Improved security, resource utilization	Knowledge of private matrix Eve is necessary
Resource management	Kim et al., 2016 [108]	Efficient resource management scheme	XML used to achieve data sensing storage system	Better availability, scalability and processing amount	Data stored on cloud has to be homogenous
Security	Maati & Saidouni, 2020 [109]	CIoTAS protocol for denial-of-service attacks	Autonomic computing used along with Cloud IoT paradigm	Fault tolerant IoT devices	Only denial service attacks considered for security
	Sharma & Kalra, 2020 [110]	Lightweight remote user authentication scheme	Authentication and password change phases used	Better security	Login phase takes longer as compared to other algorithms
	Wazid et al., 2020 [111]	LAM-CIoT	Used one-way cryptographic hash functions along with bitwise OR operations	better communication, computation overheads	Energy consumption is not considered for the smaller messages transmitted
SLA	Nawaz et al., 2018 [112]	Event driven based approach	preprocessing and translation module, knowledge base module and, reasoning and decision support module	System is efficient and able to predict SLA violations	There is a possibility of false-positive predictions as well
QoS	Khodkari et al., 2017 [113]	Integrated IoT with cloud services with QoS assured	Genetic Algorithm based service used to calculate QoS values	QoS is assured	No experimental environment is discussed
	Nawaz et al., 2017 [114]	Event based approach for monitoring QoS	Event calculus used to monitor QoS values	QoS compliance is achieved	SLA violation is not predicted
	Asghari et al., 2020 [115]	SFLA-GA	Combines two meta heuristic: SCE and PSO	Improved fitness	Not dynamic in nature
Mobility	Badii et al., 2019 [116]	Sii mobility:-Smart city mobility and transportation	IoT devices and smart gates to predict traffic	Dynamic switching of a road	Experts required to operate the framework

TABLE 3.2: COMPARISON OF ALGORITHMS ON THE BASIS OF QOS PARAMETERS

Challenges	Name	ET	MST	RTNT	Reliability	Authentication	SLA-V	EC	Throughput	RM	Security	FT	Availability	Latency	RT	TR	PD	Power	Cost	Scalability
Energy Efficiency	Nan et al., 2017 [101]	no	no	no	no	no	no	yes	no	no	no	no	no	yes	yes	yes	yes	yes	yes	yes
	Mahmoud et al., 2018 [102]	no	no	no	yes	no	no	yes	no	yes	no	no	no	yes	yes	no	no	no	no	no
	Ning et al., 2019 [103]	no	no	no	no	no	no	yes	no	no	no	no	no	yes	no	yes	no	yes	yes	yes
Latency	Nan et al., 2018 [104]	yes	no	no	no	no	no	no	yes	yes	no	no	no	yes	yes	yes	yes	no	yes	no
Resilience	Khan et al., 2019 [105]	no	no	no	yes	no	no	no	no	yes	no	no	no	no	no	no	no	no	no	yes

Offloading	Hasan et al., 2018 [106]	yes	no	no	yes	no	no	no	no	yes	no	no	no	no	no	no	no	no	no	no
	Jia et al., 2019 [107]	no	no	no	no	no	no	yes	no	yes	yes	no	no	no	no	no	no	yes	no	no
Resource management	Kim et al., 2016 [108]	no	no	no	yes	no	no	no	no	yes	no	no	yes	yes	no	no	no	no	no	no
Security	Maati & Saidouni, 2020 [109]	no	no	no	no	no	no	yes	no	no	yes	yes	yes	no	yes	yes	no	no	no	no
	Sharma & Kalra, 2020 [110]	yes	no	no	no	yes	no	no	no	no	yes	no	no	yes	no	yes	no	no	yes	no
	Wazid et al., 2020 [111]	no	no	no	no	yes	no	no	yes	yes	yes	no	no	yes	no	yes	no	no	no	yes
SLA	Nawaz et al., 2018 [112]	yes	no	yes	no	no	yes	no	no	no	no	no	yes	no	yes	no	no	no	no	no
QoS	Khodkari et al., 2017 [113]	no	no	no	yes	no	no	no	yes	no	no	no	yes	yes	yes	no	no	no	no	yes
	Nawaz et al., 2017 [114]	no	no	yes	no	no	no	no	no	no	no	no	no	no	yes	no	no	no	no	no
	Asghari et al., 2020 [115]	no	no	no	yes	no	no	no	yes	no	no	no	yes	yes	yes	no	no	no	yes	no
Mobility	Badii et al., 2019 [116]	no	no	no	yes	yes	no	yes	no	no	yes	no	no	yes	no	yes	no	yes	no	yes

TABLE 3.3: ALGORITHMS COMPARED ON THE BASIS OF CONSTRAINTS

	Name	**Priority Constraint**	**Deadline constraint**	**Simulation/Real environment**	**VM Specification analysis**	**Static/Dynamic**
Energy Efficiency	Nan et al., 2017 [101]	no	no	simulation	no	dynamic
	Mahmoud et al., 2018 [102]	no	no	simulation	no	dynamic
	Ning et al., 2019 [103]	no	no	simulation	no	dynamic
Latency	Nan et al., 2018 [104]	yes	yes	simulation	no	Dynamic
Resilience	Khan et al., 2019 [105]	no	no	simulation	no	dynamic
Offloading	Hasan et al., 2018 [106]	no	yes	simulation	yes	dynamic
	Jia et al., 2019 [107]	no	no	simulation	no	static
Resource Management	Kim et al., 2016 [108]	no	yes	simulation	no	dynamic
Security	Maati & Saidouni, 2020 [109]	no	no	real	no	dynamic
	Sharma & Kalra, 2020 [110]	no	no	simulation	no	static
	Wazid et al., 2020 [111]	no	no	simulation	no	dynamic
SLA	Nawaz et al., 2018 [112]	no	yes	simulation	yes	dynamic
QoS	Khodkari et al., 2017 [113]	no	no	theoretical	no	dynamic
	Nawaz et al., 2017 [114]	no	yes	theoretical	no	dynamic
	Asghari et al., 2020 [115]	no	no	simulation	no	static
Mobility	Badii et al., 2019 [116]	no	no	Simulation	no	dynamic

3.7 How to resolve limitations of CoT using existing techniques: Cloud is integrated with the working of IoT to increase the processing capabilities of the smart devices. This in turn increases the coverage with Cloud services. Hence for various applications where smart devices are used. Cloud collects data over scattered geography and is able to provide higher processing power and resources. Although, it also introduces challenges to the framework. The distance of cloud servers from IoT devices doesn't assist in reducing the latency for processing tasks. Similarly, the various types of devices also pose a challenge to limited bandwidth and existing security standards.

These limitations can be tackled with the help of a computing paradigm with its own processing and storage capabilities placed much closer to the smart devices. Fog computing is the solution in this case. Servers and resources are brought closer to the end nodes, reducing the latency drastically. Only the jobs with higher processing requirements might be offloaded to the cloud and the delay-sensitive tasks can be processed in the fog layer. This framework is discussed in detail in the following section 4.

4 FOG COMPUTING

The transmission of big data via IoT has been increasing exponentially. IoT devices have limited resources for processing data. The use of cloud computing ensures elastic and on-demand resource provisioning. Although, the need for real-time processing with better security for IoT devices gave rise to another paradigm. In CoT, computing resources were present at the Cloud level. Instead, fog computing is a framework where the computing resources are placed in the closed vicinity of the end devices. Since the computing resources are extended from the cloud towards the data sources, the latency in processing is reduced drastically because the load is now not managed only by the cloud. Also, since the data centers are located at the edge of the network, fog computing provides comparatively additional security. As compared to cloud computing's huge number of available resources like datacenters, fog computing includes datacenters with processing power lesser than that of clouds. The fog layer is present at the edge of the network closer to the smart devices [122, 205]. There have been many studies as to design various dynamic resource provisioning and portability of resources if required, without compromising the quality of service [4, 121].

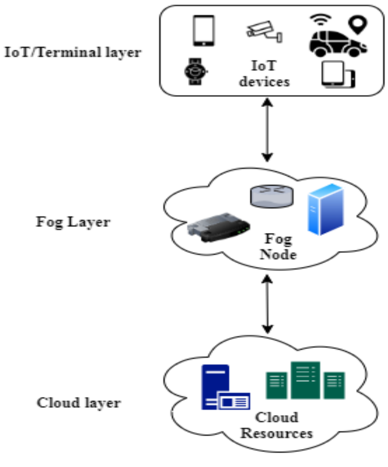

Fig. 4.1: Fog architecture

4.1 Architecture: The fog computing paradigm brings processing nodes from the cloud towards IoT. This creates a hierarchical architecture of communication between the smart devices, fog nodes, and cloud infrastructure. The fog layer extends cloud services to the edge [123]. The fog architecture is depicted in Fig. 4.1.

Terminal layer: This layer consists of geographically separated smart devices which might interact with each other if required. These devices like sensors collect information and then forward it to the next layer.

Fog Layer: The smart devices in the terminal layer transmit their data to the nodes in the fog layer. This layer is at the edge of the network and made available when time-sensitive data is needed to be processed in real-time. Fog nodes have resources to store and process the transmitted data. Though if required, data might be transferred to the next layer, i.e., the cloud layer.

Cloud Layer: Cloud layer comprises of highly efficient resources like servers, storage devices, and processing components. So, if the need arises for powerful computing or permanent storage, the fog layer transfers data to the cloud layer.

4.1.1. Layered Architecture of Fog computing: Fog computing can also be divided into layered architecture on the basis of data flow and functionalities as depicted in Fig. 4.2 [25]. The first layer is the Physical and Virtualization layer which consists of edge devices in the network. The next layer named the Monitoring layer is responsible for the handling of requests and tasks. Data management jobs like filtering of data are carried out at the next level by the Pre-processing layer. The Temporary storage layer stores data until it would be required to transmit again. The next layer, namely Security Layer is responsible for ensuring the security of the network and data. Finally, Transport Layer is responsible for transmitting data to the cloud layer.

4.2 Features of Fog Computing*:* Fog nodes consist of processors and servers, in proximity to the smart devices. Smart devices are then able to transfer their data and request processing within the time limit. This allows even the bulky transmission from IoT devices to fog nodes through smaller dedicated channels. Unlike cloud nodes, fog nodes allow mobility without compensating for the connection. As well as, fog nodes possess resources quite lesser than cloud, reducing the energy consumption by a node. The benefits provided by the fog computing framework are discussed as follows:

Heterogeneity: Various types of fog nodes are formed by devices such as servers, routers and, gateways. Similarly, networks too can be different, like – high-speed links to servers, and wireless connections with smart devices. This variety of both devices and network connections makes the framework and data heterogeneous.

Proximity to IoT: Fog nodes are distributed to support the mobility of terminal devices. This decentralized nature of fog computing enables the storage and processing of data much closer to the source of data. Smart devices are then able to receive the processed information faster.

Low latency: For real-time processing, the Fog layer enables bringing the computing power to the IoT devices, and also stores the data temporarily, if required. Fog nodes help in processing and storing data for time-sensitive tasks leaving the ones with higher computation requirements for the cloud. This results in real-time computing and reduced latency for time-sensitive processes.

Support for mobility: IoT devices like smartphones, and vehicles, are mobile in nature. So, the fog nodes are also mobile if the situation is so. This property of fog nodes ensures that the fog nodes communicate directly with the devices, as intended.

Security: Since fog services are closer to terminal devices, the need for third-party services on the cloud is reduced drastically. This in turn reduces the risk of security attacks on devices. Fog nodes also provide encryption schemes and isolation that increase the security of the heterogeneous terminal devices and sensitive data [124].

Low energy consumption: Fog nodes are comprised of limited process power and storage, for real-time processing and temporary storage of data. Also, these nodes are decentralized. This reduces the energy consumption of fog nodes [123,125].

4.3 Applications of Fog Computing*:* The proximity of fog nodes to edge devices allows real-time processing. This opens up possibilities for multiple applications where the fog paradigm can be implemented. Along with delay-sensitive processing, the distributed nature of the fog framework allows processing even if the smart devices are mobile in nature.

Artificial Intelligence: Artificial intelligence was introduced in the fog computing paradigm because it brought the ability to automate processes [212]. It also enables prediction and efficiency in features like load balancing and resource sharing. Studies have also been carried out for the inclusion of such algorithms in the Fog framework for applications like smart healthcare.

Blockchain: Fog computing faces security and privacy concerns because of the heterogeneous nature of end devices and data. For this reason, studies have been carried out in this field. Such a study integrated blockchain along with Fog computing [216]. Blockchain technology helps in maintaining a ledger of all the transactions in a shared manner. This helps in enforcing security and privacy.

Healthcare: Fog computing offers decentralization, mobility, and real-time processing. These all are required for processing sensor data, where patients might be mobile, and processing might be time-sensitive for health-related issues. Storing patient's past and ongoing treatment/s on the cloud is helpful in managing current treatments from anywhere. Fog helps in managing healthcare data and also alerts the concerned individual or staff, in case of an emergency.

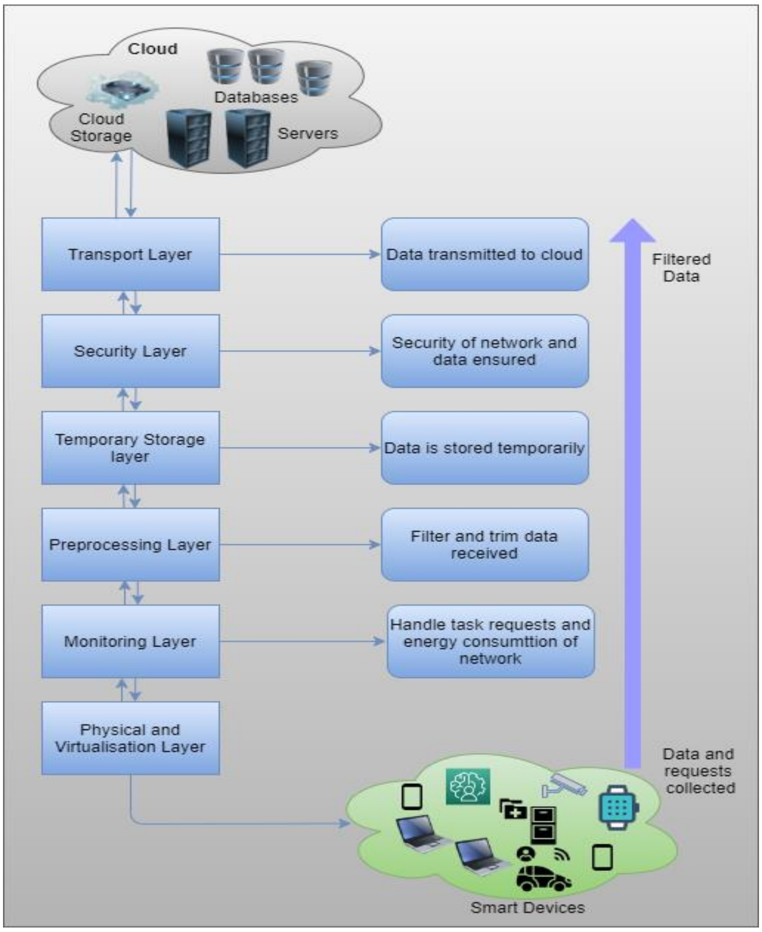

Fig. 4.2: Layered architecture of Fog Computing

Vehicular fog computing: Fog nodes might be deployed statically on roads or other transportation routes, or be mobile in nature to form fog ad-hoc networks. This paradigm is helpful for traffic efficiency, traffic congestion control, and in processes of a similar manner. The information regarding the current status of traffic and any obstacles are collected by smart devices and sensors. This information is then analyzed and used in smart traffic signals, GPS, etc.

Smart environments: Cloud computing for IoT devices would incur issues like mobility, scalability, and latency. It is so as smart devices need to communicate directly to the cloud for processing. The involvement of processing power in the proximity of IoT devices allows the advent of smart environments. Since smart devices are distributed, hence fog nodes are helpful in processing the sensors' data, relatively faster [123,126].

4.4 Research Challenges*:* Introducing the fog paradigm closer to smart devices has its own advantages. Although, the processing and storage capabilities of fog nodes as well are not quite sufficient for heavy tasks. Also, fog nodes include only a limited number of resources as compared to the cloud's scalable third-party resources. These channels of data transfer are susceptible to security attacks as well. It is so because implementing a single security standard for heterogeneous data and channels might need more dedicated research.

- **Security:** The security standards designed for the cloud, do not work for the fog layer because devices and networks are heterogeneous in nature. There is no appropriate security standard to handle the various attacks on the heterogeneous entities of the framework.
- **Limited resources:** The fog layer is equipped with limited resources for processing and storing data. This results in the need for efficient resource management for the performance of nodes. Hence, it is one of the recent research trends to offload tasks and data to the cloud. This enables to achieve fully optimized fog nodes as well as executing real-time tasks.
- **Management:** Fog nodes are distributed geographically with the scattered IoT devices, mobile and stationary. An increasing number of smart devices entails an increased number of fog nodes. As per the resources and computing power, tasks are assigned to the nodes. These decentralized nodes are required to be managed for the efficient use of resources.

TABLE 4.1: SUMMARY OF ALGORITHMS PROPOSED ALONG WITH LIMITATIONS

	Name	Algorithm	Technique	Results	Limitations
Availability	Lera et al., 2019 [127]	Service placement policy	Map applications in fog and allocate services to devices	Improved service availability	Mobility patterns is not considered
	Mseddi et al., 2019 [128]	Novel resource management algorithms for flexible service provisioning	Optimization problem platform is solved using CPLEX, proposed PSO-based metaheuristic	PSO-based algorithm achieves near-optimal results	Mobility of user nodes is not considered
Energy Efficiency	Pooranian et al., 2017 [129]	Proposed fog data centre architecture; energy aware algorithm adopts Fog data center	FDC provides a platform for filtering and analyzing the data generated by sensors	Improved resources, energy consumption, and response time	Fog devices are assumed to be bound in a geographical area
	Oma et al., 2018 [130]	Tree based fog computing (TBFC) model	Processes and data are distributed to servers and fog nodes, with minimum energy consumption.	Reduced execution time ratio	Mobility of nodes is not considered
	Xiao & Krunz, 2018 [131]	Novel cooperative fog computing concept-offload forwarding	Distributed optimization framework based on dual decomposition to achieve optimal trade-off.	Efficient power usage; reduce the service latency for users by around 50%	Assumed pre-existing communication links
	Karimiafshar et al., 2020 [132]	Algorithm for dynamic request dispatching, and frequency and modulation level scaling	Algorithm is based on the current system conditions and the queues' backlog information	Improved service time, the number of deadline misses and energy utilization	Mobility of nodes is not considered
Fault Tolerant	Xu et al., 2018 [133]	Byzantine fault-tolerant networking method and two resource allocation strategies	Fog networking method based on BFS and two BFT resource allocation strategies	Efficient and reliable fog network when faced with Byzantine faults	Framework relies on mutual assistance of fog nodes
	Wang et al., 2020 [134]	RVNS-based sensor Data Processing Framework (REDPF)	Combined advantages of Directed Diffusion and Limited Flooding to enhance the reliability of data transmission	Improved network reliability and faster processing speed	Failure in recollecting lost packets if all the links are broken
Latency	Mahmud et al., 2018 [135]	Latency-aware Application Module management policy for the fog environment	Nodes in fog layer are organized hierarchically	QoS satisfied, resource utilization	Varying processing time of modules, reduces QoS rate
	La et al., 2019 [136]	Device-driven and human-driven intelligence as key enablers to reduce energy	Machine learning technique used to detect user behavior, and perform adaptive low-latency MAC-layer scheduling	Improved context awareness, network adaptability, reduced energy consumption	Security mechanism is not considered
	Martinez et al., 2020 [137]	Optimal design and dimensioning formulation of the fog infrastructure	Used MILP to minimize infrastructure costs and a near optimal column generation formulation	Reduced computation time, scalable design	IoT traffic is not considered to be fluctuating
	Mukherjee et al., 2020 [138]	A latency-driven task data offloading problem	Applied SDR to the optimization problem	Reduced delay	Not consider energy consumption
Mobility	Martin et al., 2020 [139]	An autonomic framework MAMF, to perform migrations	The framework uses MAPE loop concepts and Genetic Algorithm	Average delay, network usage, and cost of execution significantly reduced	The antenna used is assumed to work to its full efficiency
QoS	Skarlat et al., 2017 [140]	Model for an IoT application	FSPP used to formalize optimization model	Execution cost reduced	Cost of resources not considered
	Cao et al., 2019 [141]	Hierarchical renewable-adaptive QoS optimization approach	Techniques of cooperative game theory and mixed-integer linear programming used	Improves the system QoS and application QoS fairness	Fog server is assumed to have unlimited power supply
Resource scheduling	Hong et al., 2018 [142]	QoS-aware network resource management framework	qCon framework is used to bridge driver model for networking and for implementing scheduling framework	Network latency decreased; lowered CPU overhead	Bandwidth control performed only on outbound traffic
	Sun et al., 2018 [143]	Novel resource scheduling scheme	Improved non dominated sorting genetic modified algorithm	Increased stability of task execution	no node failure mechanism
	Li et al., 2019 [144]	fuzzy clustering-based resource scheduling	Fuzzy clustering and particle swarm optimization used	Higher clustering accuracy	Resources assumed static
	Rafique et al., 2019 [145]	Novel bio-inspired hybrid algorithm	Modified PSO and modified cat swarm optimization (MCSO)	Better energy consumption	Execution time increases if no resource found
Resource Provisioning	Yao & Ansari, 2019 [146]	Modified best fit decreasing algorithm	Inspired by the best fit decreasing (BFD) algorithm	Efficient failure recovery ratio	mobility not considered

	Santos et al., 2019 [147]	Network-aware scheduling approach	Fog architecture based on Kubernetes	Efficient provisioning of services, reduced network latency	Bandwidth fluctuations not considered
	Feng et al., 2019 [148]	Proposed dynamic Stackelberg game for dynamic interactive decision making	Dynamic Stackelberg game framework based on optimal control theory and evolutionary game theory	Scalable framework, defending against the APT attacks	Framework is neither simulated nor implemented in real environment
Security	Daoud et al., 2019 [149]	Proposed a clustering algorithm for security	Control scheme based on trust assessment and user's activities	Efficient network usage, security, latency optimized	SLA violation, energy efficiency not considered
	Gill et al., 2020 [150]	Framework to place of multimedia files based on security requirements	Deep neural network used to evaluate parameters and requirements	84% accuracy in selecting fog environment without compromising security	Deadline is not considered as a parameter
	Hussein et al., 2020 [151]	Hybrid security strategy	HS2 contributes encryption algorithm and steganography methodology	Secured fog environment against common attacks	Framework is not dynamic in nature

TABLE 4.2: COMPARISON OF ALGORITHMS ON THE BASIS OF QOS PARAMETERS

	Name	**ET**	**MST**	**RT**	**Reliability**	**Authentication**	**SLA-V**	**EC**	**Throughput**	**CPU-U**	**Security**	**Availability**	**FT**	**Delay**	**Mobility**	**Heterogeneity**	**Scalability**	**BW**	**QoS**	**RTT**
Availability	Lera et al., 2019 [127]	yes	no	yes	no	no	no	no	no	no	no	yes	no	yes	no	no	no	yes	yes	no
	Mseddi et al., 2019 [128]	yes	no	no	no	no	no	no	no	no	no	yes	no	yes	yes	no	no	yes	no	no
Energy Efficiency	Pooranian et al., 2017 [129]	no	no	yes	yes	no	yes	yes	no	no	yes	no	no	no	yes	yes	yes	no	yes	no
	Oma et al., 2018 [130]	yes	no	no	no	no	no	yes	no	no	no	no	no	no	no	no	no	no	no	no
	Xiao & Krunz, 2018 [131]	no	no	yes	no	no	no	yes	no	no	no	no	no	yes	no	no	yes	no	no	no
	Karimiafshar et al., 2020 [132]	no	no	no	no	no	no	yes	no	no	no	yes	no	yes	no	no	no	no	no	no
Fault Tolerant	Xu et al., 2018 [133]	no	no	no	yes	no	no	yes	no	no	yes	no	yes	yes	yes	yes	no	yes	no	no
	Wang et al., 2020 [134]	no	no	no	yes	no	no	no	yes	no	no	no	yes	no	no	no	no	yes	no	no
Latency	Mahmud et al., 2018 [135]	no	no	no	yes	yes	no	yes	no	yes	no	yes	no	yes	no	yes	no	yes	yes	no
	La et al., 2019 [136]	no	no	no	no	no	no	yes	no	no	no	no	no	yes	no	yes	no	no	yes	no
	Martinez et al., 2020 [137]	no	no	no	no	no	no	no	no	no	no	yes	no	yes	no	yes	yes	yes	yes	no
	Mukherjee et al., 2020 [138]	yes	no	yes	no	no	no	no	no	no	no	no	no	yes	no	no	yes	no	no	no
Mobility	Martin et al., 2020 [139]	no	no	yes	no	no	no	no	no	no	no	no	no	yes	yes	yes	no	yes	no	no
QoS	Skarlat et al., 2017 [140]	yes	yes	yes	no	no	no	no	no	no	no	no	no	no	no	no	no	no	yes	no
	Cao et al., 2019 [141]	yes	no	no	no	no	no	yes	no	no	no	no	no	yes	no	yes	no	no	yes	no
Resource Scheduling	Hong et al., 2018 [142]	no	no	no	no	no	no	yes	no	yes	no	no	no	no	no	no	no	yes	yes	no
	Sun et al., 2018 [143]	yes	no	no	no	no	no	no	no	yes	no	no	no	no	yes	yes	no	yes	yes	no
	Li et al., 2019 [144]	no	no	no	no	no	no	no	no	yes	no	no	no	no	no	yes	no	yes	yes	no

	Rafique et al., 2019 [145]	yes	no	yes	no	no	no	yes	no	no	no	no	no	no	no	no	no	no	no	no
Resource Provisioning	Yao & Ansari, 2019 [146]	no	no	no	yes	no	no	no	no	yes	no	no	no	yes	no	yes	no	no	yes	no
	Santos et al., 2019 [147]	yes	no	no	yes	no	no	yes	no	yes	no	no	no	no	no	no	yes	yes	yes	yes
	Feng et al., 2019 [148]	no	no	no	no	no	no	no	no	no	yes	no	no	no	no	no	yes	no	no	no
Security	Daoud et al., 2019 [149]	yes	no	yes	no	no	no	no	no	no	yes	yes	no	yes	no	yes	yes	yes	yes	no
	Gill et al., 2020 [150]	no	no	no	no	no	no	no	no	no	yes	no	no	no	no	no	no	no	no	no
	Hussein et al., 2020 [151]	yes	no	no	yes	yes	no	yes	no	no	yes	no	no	yes	no	no	no	no	no	no

TABLE 4.3: ALGORITHMS COMPARED ON THE BASIS OF CONSTRAINTS

	Name	**Priority constraint**	**Deadline constraint**	**Execution (simulation/real env) constraint**	**VM specification analysis (RAM, processing power)**	**Static/ Dynamic**
Availability	Lera et al., 2019 [127]	no	yes	simulation	no	dynamic
	Mseddi et al., 2019 [128]	no	no	simulation	yes	dynamic
Energy Efficiency	Pooranian et al., 2017 [129]	no	no	simulation	yes	dynamic
	Oma et al., 2018 [130]	no	no	simulation	yes	static
	Xiao & Krunz, 2018 [131]	no	no	real	no	dynamic
	Karimiafshar et al., 2020 [132]	no	yes	simulation	no	dynamic
Fault Tolerant	Xu et al., 2018 [133]	no	no	simulation	yes	dynamic
	Wang et al., 2020 [134]	no	no	simulation	yes	dynamic
Latency	Mahmud et al., 2018 [135]	yes	yes	simulation	no	dynamic
	La et al., 2019 [136]	no	no	simulation	no	dynamic
	Martinez et al., 2020 [137]	no	no	simulation	yes	static
	Mukherjee et al., 2020 [138]	no	no	simulation	yes	dynamic
Mobility	Martin et al., 2020 [139]	no	yes	simulation	no	static
QoS	Skarlat et al., 2017 [140]	yes	yes	simulation	yes	static
	Cao et al., 2019 [141]	no	yes	simulation	yes	dynamic
Resource Scheduling	Hong et al., 2018 [142]	yes	no	simulation	yes	dynamic
	Sun et al., 2018 [143]	yes	no	simulation	yes	static
	Li et al., 2019 [144]	no	no	simulation	yes	static
	Rafique et al., 2019 [145]	no	no	simulation	no	static
Resource Provisioning	Yao & Ansari, 2019 [146]	no	yes	simulation	no	static
	Santos et al., 2019 [147]	yes	no	simulation	yes	static
	Feng et al., 2019 [148]	no	no	-	yes	dynamic
Security	Daoud et al., 2019 [149]	yes	no	simulation	yes	dynamic
	Gill et al., 2020 [150]	no	no	simulation	no	static
	Hussein et al., 2020 [151]	no	no	simulation	no	dynamic

• **Fault Tolerance:** In case of node or device failure, the users should be able to access normal services with the help of some other working node. Although for such a decentralized system it becomes tedious to identify the faulty node. This motivates the study of designing fault-tolerant algorithms in fog computing to ensure the continuous execution of tasks in any similar situation [123, 125].

4.5 Proposed solution*:* The recent researches in the area of fog are studied and categorized as in Tables 4.1, 4.2, and 4.3. The parameters and the constraints considered to compare with these works are discussed in sections 1.4 and 2.5 respectively. As discussed in previous paradigms, these parameters are used to depict a specific set considered for an algorithm or framework. The constraints present in Table 4.3 depict the set of constraints for respective executions and their respective efficient results. For each framework, there are compromises made in one or more parameters for achieving efficiency in desired parameters. These proposed frameworks, along with their consecutive trade-offs are studied in the following paragraphs.

Various studies searched in the area of fog are studied and analyzed as follows in Table 4.1, organized as described in section 2.5. All the studies are organized first according to the parameters worked on. Techniques employed along with the aims achieved are compared along with each article's challenges. The algorithms are discussed with their techniques and shortcomings in this section. A few of these studies are discussed in the following paragraphs. After the analysis of the articles in Table 4.1, the algorithms are compared as to which parameters are considered in each study in Table 4.2. Though all the studies considered are not successful in considering every parameter. These trade-offs help in observing as to which technique improves parameters. The constants considered in each study are then mentioned in the articles in Table 4.3. Hence these tables depict as to which parameters improved efficiently with the help of a specific algorithm, and under what circumstances.

A service placement policy is proposed by firstly mapping the applications to fog communities and then services to fog devices [127]. Service availability and QoS are improved, although, as it is assumed that the cloud device has infinite resources, it incurs increased cost. A particle swarm optimization-based metaheuristic and greedy heuristic algorithm is proposed using CPLEX to decrease execution time and increase availability [128].

A latency-driven task data offloading framework is proposed by applying semidefinite relaxation (SDR) to the optimization problem, resulting in reduced delay [138]. In another article, a hierarchical renewable-adaptive QoS optimization algorithm was proposed using cooperative game theory [141]. QoS of the entire system is improved as a result. Although, both aforementioned frameworks ignore energy consumption.

The bridge driver model is used for proposing and implementing a QoS aware network resource management framework, named qCon [142]. The network latency and CPU overhead are decreased as a result, but bandwidth only for outgoing traffic is controlled and not the incoming traffic. In another study, an Optimized fuzzy clustering-based resource scheduling algorithm is proposed using Fuzzy clustering and particle swarm optimization [144]. Efficient resource scheduling and higher clustering accuracy are achieved. As a compromise, the dynamic nature of resources in fog environment is ignored.

4.6 Limitations of Fog:

• **Processing power:** The computation required by the IoT in fog environment, is carried out by the fog servers. Although this framework is latency-sensitive as the processing of data is brought close to the end devices. But the processing power of fog nodes is limited. So, with the increasing workload, delay-sensitive processes might not be processed within a specific time frame [155]. Tasks would be required to offload to Cloud as per their requirement of processing power.

• **Storage:** End nodes in fog environments need the servers to store data for future and/or immediate computing. Fog nodes might be required to store data at instances. Although because of limited resources, storing huge amounts of data in these nodes is not efficient. Cloud is hence required to be able to store data when needed.

• **Load balance:** Fog servers are required to balance the workload for processing every delay-sensitive operation within a permissible time limit. For a specific scenario, the end devices might be transferring huge data or large number of service requests to fog servers for real-time processing. With the limited resources of fog, it is difficult to process in real-time [153].

• **Network bandwidth:** For high computing processes, and with increasing end devices in the environment, fog servers need to accommodate the delay-sensitive processes within the bandwidth. But it is not possible until the servers are more powerful or some changes are introduced in the fog environment for higher performance [154]. Either an increase in fog servers or offloading selected tasks to the cloud might amend network bandwidth challenges.

• **Security:** The presence of heterogeneous and sporadically present devices results in non-standard protocols that need to be implemented as per the environment. Hence, further research is required so that the framework is able to maintain a standard set of protocols, provided QoS parameters are not compromised [152].

4.7 How to resolve limitations of fog using existing techniques: Processing real-time data of end devices requires a framework closer to IoT than the Cloud. Hence, it was required to implement a fog framework. The real-time processing is carried out efficiently by the fog paradigm, introducing processing power in between cloud and end devices. Also, security for heterogeneous IoT devices is increased as compared to in cloud frameworks. On the downside, fog servers have limited processing as well as storage capabilities. Hence, there is a need for the incorporation of Fog and Cloud along with IoT. The involvement of the cloud would be able to ensure higher processing and storage power. The framework and standards help fog nodes as explained in section 5.

5 FOG OF THINGS (FOT)

The IoT layer generates task requests in a heterogeneous and mobile manner. Fog paradigm is integrated with IoT layer to process real-time tasks, while also reducing the workload of the cloud. Although, there are a few limitations of the fog paradigm as discussed in the preceding section. One of the limitations of this framework is limited resource power, which makes it difficult for powerful processing. The Cloud paradigm has servers which provide higher computation power. Alternatively, fog servers allow real-time tasks to be executed within a specific time frame while the Cloud paradigm faces the limitation of huge latency. Hence to compensate for the challenges in both architectures, Cloud and Fog paradigms are integrated. This integration of Fog with Cloud of Things is called Fog of Things. This integrated paradigm is discussed in this section.

5.1 Architecture: It is important to learn and design more efficient architecture of the paradigm, as needed. In the most used framework of the paradigm, the terminal devices and fog/cloud layer are connected via a controller layer. This layer is responsible for the virtualization of fog and cloud nodes, inducing flexibility as compared to the limited resources of the Fog layer. The resource allocation algorithm on Fog/cloud tracks the virtualized resources for further requests [156]. The aim of algorithms used for various use cases is to minimize the limitations of fog and cloud paradigms for an integrated efficient architecture. Algorithms can be designed based on QoS parameters for processing data on both Fog layer and Cloud.

5.2 Features:

Load Balance: IoT transfers data to fog framework for processing of delay-sensitive tasks. Though the amount of data to be processed from IoT might overwhelm fog servers' capacity. As the Cloud is included in the framework with Fog, the offloading possibilities increase drastically, increasing the processing capability of the entire framework.

Delay sensitive: In an FoT environment, the processes can be provisioned to either Cloud or fog servers on the basis of time sensitivity and processing requirements of the jobs. Cloud processing the jobs might not produce results in real-time. Hence delay-sensitive tasks are to be offloaded to Fog for processing instead of Cloud. This leads to reduced delay of tasks as they get executed within the required time, without overloading both Cloud and fog servers.

Resources: Fog servers have limited resources. Some requests submitted by the smart devices might require higher computation power, which is implemented on the cloud. It hence increases the delay in response if the task is assigned to cloud servers when required. Also, studies on fog servers usually neglect the storage capacity of the framework. Storing and processing of big data from IoT along with delay-sensitive processing is provided by the fog-cloud architecture.

Security: IoT processing in a cloud environment possess security concerns because of the presence of third-party resources in the environment. Security measures of the fog paradigm in the integrated fog-cloud environment ensure the security of the heterogeneous end devices while providing processing and storing power of the cloud as well.

Heterogeneity: A fog environment manages the heterogeneity of IoT devices, while the cloud is incorporated to operate tasks from heterogeneous IoT from diverse geographical areas.

5.3 Applications: With the inclusion of Cloud along with fog and IoT, the framework becomes capable of executing tasks requiring higher computing power, with reduced delay. Incorporating cloud in the framework allows algorithms to address various offloading techniques. As a result, this opens up quite a few opportunities as discussed:

Smart Grid: Smart grid is designed for energy utilization along with the reduced cost of operating. The different components of a smart grid require to be managed efficiently for achieving the results. Since the end devices are heterogeneous, they are managed by the fog environment. Whereas, the geographically scattered nature of end devices requires a cloud environment to operate and manage the data collected over a large area [159].

Software-Defined Networks: For transferring data from end devices to fog and if required, to the cloud, one requires a defined framework to achieve the quality of the network while maintaining the integrity of the information transferred. Many studies have been carried out to propose an FoT architecture for SDN [160].

TABLE 5.1: SUMMARY OF ALGORITHMS PROPOSED ALONG WITH LIMITATIONS

	Name	Algorithm	Technique	Results	Limitations
Energy Efficiency	Deng et al., 2016 [162]	Optimal workload allocations between fog and cloud	Approximate approach is used to divide the problem in three subparts	Reduced communication latency	Worsens power consumption if workload allotted to fog nodes
	Adhikari & Gianey, 2019 [163]	Meta-heuristic based offloading strategy	Firefly algorithm used to find optimal computing device based on energy consumption and computational time	Improved computational time, CO2 and temperature	SLA is not considered
	Sun et al., 2020 [156]	IoT-Fog-Cloud architecture for time and energy efficient computation offloading	ETCORA algorithm used to achieve aim of architecture	Reduced energy consumption and completion time of requests	Security and reliability is not considered
Latency	Du et al., 2019 [164]	Low-complexity general algorithm framework	Offloading decisions made by binary tailored fireworks algorithm	Decreased delay	Algorithm is not dynamic
	Abbasi et al., 2020 [165]	Model for problem of trade-off between energy and delay	NSGAII algorithm is used	Both energy and delay improved	Algorithm is not dynamic
	Yang, 2020 [166]	BAT algorithm to solve optimization problem	Powell local search to speed up the convergence of algorithm	Processing delay reduced	load balancing not distributed
QoS	Sood, 2018 [167]	Free space fog for fog layer in mobile device	Social Network Analysis used to detect deadlock and remove deadlock	Deadlock detection, resource utilization, QoS, reliability provided	If free fog is occupied and request is bottom priority, then public cloud
	Emami & Saeed, 2020 [168]	Cloud-based platform for management of IoT service selection and composition	Evolutionary game theory, enhanced by evaporation-based water cycle algorithm (EG-ERWCA)	Efficient monitoring of IoT devices, improved reliability and availability	Performance of the algorithm worsens if number of jobs is less
Resource Provisioning	Taneja & Davy, 2017 [169]	Module Mapping Algorithm	deployed Application Modules in Fog-Cloud Infrastructure for IoT based applications	Decreased network usage, balanced energy consumption	Dynamic fog and cloud components not considered
	Du et al., 2018 [170]	Low-complexity suboptimal algorithm	Offloading decisions used CORA algorithm, and the resource allocation is obtained using BCRA algorithm	Longer the delay constraint, the more energy saved	Heterogeneous networks of fog nodes are not considered
	Silva & Fonseca, 2018 [171]	GPRFCA	Employs a Gaussian Process Regression to predict future demands	Reduced energy consumption	Static user devices
Resource Scheduling	Stavrinides & Karatza, 2019 [172]	Hybrid fog and cloud-aware heuristic, Hybrid-EDF, for the dynamic scheduling	Schedules tasks with low communication requirements in cloud and tasks with low computational demands in fog	76.69% lower deadline miss ratio	Usage of cloud resources at significant monetary cost
Security	Fan et al., 2017 [173]	Multi-authority access control scheme	Design an efficient user and attribute revocation method	Security; better computation efficiency	CA (global certificate authority) fully trusted
	El-latif et al., 2018 [174]	Framework for secure quantum steganography	Protocol based on quantum entangled states	Proposed protocol secured	Protocol is not simulated
	Alli & Mahbub, 2021 [175]	Secure computation offloading scheme in Fog-Cloud-IoT environment	Neuro-Fuzzy Model to secure data at the smart gateway; and optimum fog node chosen by PSO	Minimised latency, delay	Security attained is not measured
	Comput et al., 2020 [176]	Authentication protocol	Protocol proposed with proper key establishment between the cloud, fog, and user	Secured protocol, better communication overheads	Communication cost is at times more than already existing schemes

TABLE 5.2: COMPARISON OF ALGORITHMS ON THE BASIS OF QOS PARAMETERS

	Name	CT	MST	RT	Reliability	Authentication	SLA-V	EC	Through put	CPU-U	Security	Availability	FT	Cost	Power	Delay
Energy Efficiency	Deng et al., 2016 [162]	no	no	no	no	no	no	no	no	no	no	no	no	no	yes	yes
	Adhikari & Gianey, 2019 [163]	yes	no	no	no	no	no	yes	no	no	no	no	no	no	yes	yes
	Sun et al., 2020 [156]	yes	no	no	no	no	no	yes	no	no	no	no	no	no	yes	yes
Latency	Du et al., 2019 [164]	yes	no	no	no	no	no	no	no	no	no	no	no	no	yes	yes

	Abbasi et al., 2020 [165]	no	no	yes	no	no	no	yes	no	no	no	no	no	no	no	yes
	Yang, 2020 [166]	no	no	no	no	no	no	no	no	no	no	no	no	no	no	yes
QoS	Sood, 2018 [167]	no	no	yes	yes	no	no	no	no	no	yes	no	no	yes	no	yes
	Emami & Saeed, 2020 [168]	no	no	yes	yes	no	yes	no	no	yes	yes	no	no	yes	yes	no
Resource Provisioning	Taneja & Davy, 2017 [169]	no	no	yes	no	no	no	yes	no	no	no	no	no	yes	yes	yes
	Du et al., 2018 [170]	no	no	no	no	no	no	yes	no	no	no	no	no	yes	no	yes
	Silva & Fonseca, 2018 [171]	no	no	no	no	no	no	yes	no	yes	no	no	no	no	yes	yes
Resource Scheduling	Stavrinides & Karatza, 2019 [172]	no	no	no	no	no	no	no	no	no	no	no	no	yes	no	no
Security	Fan et al., 2017 [173]	no	no	no	no	no	no	no	no	no	yes	no	no	yes	no	no
	El-latif et al., 2018 [174]	no	no	no	no	yes	no	no	no	no	yes	no	no	no	no	no
	Alli & Mahbub, 2021 [175]	no	no	yes	no	no	no	yes	yes	yes	yes	no	no	no	no	yes
	Comput et al., 2020 [176]	no	no	no	no	yes	no	no	no	no	yes	yes	no	no	no	yes

TABLE 5.3: ALGORITHMS COMPARED ON THE BASIS OF CONSTRAINTS

	Name	Priority constraint	Deadline constraint	Execution (simulation/real env) constraint	VM specification analysis (RAM, processing power)	Static/ Dynamic
Energy	Deng et al., 2016 [162]	no	no	simulation	yes	static
	Adhikari & Gianey, 2019 [163]	no	no	simulation	yes	dynamic
	Sun et al., 2020 [156]	no	yes	simulation	yes	dynamic
Latency	Du et al., 2019 [164]	yes	no	simulation	no	static
	Abbasi et al., 2020 [165]	no	no	simulation	yes	static
	Yang, 2020 [166]	no	no	simulation	yes	static
QoS	Sood, 2018 [167]	yes	no	simulation	no	dynamic
	Emami & Saeed, 2020 [168]	no	no	simulation	no	dynamic
Resource Provisioning	Taneja & Davy, 2017 [169]	no	no	simulation	no	static
	Du et al., 2018 [170]	no	yes	simulation	no	static
	Silva & Fonseca, 2018 [171]	no	no	simulation	yes	static
Resource Scheduling	Stavrinides & Karatza, 2019 [172]	yes	yes	simulation	yes	dynamic
Security	Fan et al., 2017 [173]	no	no	simulation	yes	static
	El-latif et al., 2018 [174]	no	no	-	no	static
	Alli & Mahbub, 2021 [175]	yes	no	simulation	yes	dynamic
	Comput et al., 2020 [176]	no	no	simulation	no	static

Mobile applications: Processing closer to end devices is preferable for latency-sensitive applications requested by the mobile application. Although the restrictions on processing power near the end devices raise the need to offload some tasks on the cloud. This helps in ensuring that the delay-sensitive tasks are processed closer to the IoT. While the tasks that need higher computing power to be performed, are assigned to the cloud. Also, for scattered tasks, a greater geographical area needs to be covered which is achieved with the help of the Cloud [157].

Industrial IoT: Industrial IoT are heterogeneous elements of IoT designed to operate for industrial applications. Hence, they are managed by the fog environment. Whereas, the end devices are scattered over a large geographic area, which would require Cloud implementation. Cloud servers enable the processing and storing of big data, whereas fog servers are not capable. Hence various FoT architectures are used for implementing industrial IoT (IIoT) [161].

Blockchain: Blockchain is the concept of maintaining a record of all the transactions of the entities in a distributed manner. This results in a shared, and secure ledger of the information transferred and services requested and provided. Since blockchain is public in nature, the information is visible to every entity in the network. It hence increases security. FoT as a framework is used for efficient working of this application. There have been similar studies for implementing this concept in the FoT environment [23].

5.4 Research Challenges: With the increasing number of smart devices at the edge of the framework and similar fog nodes, the standards and security measures need to be updated regularly. These standards are required to provide the efficient QoS parameters' values, while also securing the transactions carried out.

- **Authentication:** Fog nodes are connected to the cloud in this framework. Although the protocols on the cloud might not be standardized for heterogeneous fog nodes and communication channels. This increases the risk of attacks on user credentials. Hence studies have been carried out to achieve the authentication aimed [215]. One such study enforced authentication by implementing blockchain-based certificate on the IoT-Fog-Cloud architecture [214]
- **Suspicious fog nodes:** In this framework, data is collected from smart devices, and transmitted to the fog layer. If a requirement arises, the data would be distributed among fog nodes. If any of the fog nodes is compromised, the data will be open to the attacker. The attack could then not only compromise the fog nodes, but also the connected IoT devices.
- **Security:** In this framework, the data and the framework are both heterogeneous in nature. This is because of the presence of various end devices of different nature. These devices of various nature generate data which might itself be heterogeneous in nature. This nature of the paradigm causes difficulty in ensuring security. Hence studies have been conducted in this field to enforce security in the framework. To ensure security in the heterogeneous framework of FoT, a two-layer IDS (Intrusion Detection System) architecture is designed and implemented. This framework secures the heterogeneous architecture while reducing bandwidth, latency and energy overhead [213].

5.5 Proposed solutions*:* Various studies conducted in the area of fog of things are studied and analyzed in Table 5.1. Techniques employed along with the aims achieved are compared along with each article's challenges. The algorithms are discussed with their respective techniques, results and shortcomings in this section. The structure of analysis in Table 5.1 is described in section 2.5. The papers are first categorized according to the parameters worked upon. All the studies are then explained in brief about the aims, techniques used, results achieved and the limitations of the proposed frameworks. Few of these works have been explained in the following paragraphs. After the analysis of the articles in Table 5.1, the algorithms are compared as to which parameters are considered in each study in Table 5.2. The parameters used to compare the algorithms are discussed in detail in Section 1.4. Though all the studies considered are not successful in considering every parameter. The table clearly depicts the parameters worked upon in an algorithm for efficient results, while ignoring the others. The constraints considered in each study are then mentioned in the articles in Table 5.3. These constraints are explained in Section 2.5. The conditions under which respective algorithms achieved desired results are mentioned in the table.

A resource allocation framework is proposed in an FoT environment by implementing Artificial Intelligence

[179]. A Meta-heuristic based offloading strategy is proposed and implemented. It is achieved using Firefly algorithm which finds optimal computing devices based on energy consumption and computational time [163]. The proposed algorithm resulted in better computational time, reduced energy consumption, and CO_2 emission.

An IoT-Fog-Cloud architecture for time and energy-efficient computation offloading was proposed using the ETCORA algorithm, to achieve reduced energy consumption and completion time of requests [156]. Although in the aforementioned algorithms, SLA and QoS like security and reliability were not considered [156, 163]. Free space fog layer in mobile device was proposed [167]. It was achieved using Social Network Analysis to detect deadlock. Furthurmore, the proposed framework was able to remove the deadlocks by collecting available free resources. As a result, the deadlock was detected, and resource utilization was achieved along with QoS and SLA. Although there is a discussed situation where if free fog is occupied and the request is the bottom priority, then the job is allocated to the public cloud and increases the security risk. Other studies proposed hybrid offloading in the environment minimizing the delay [177].

A hybrid fog and cloud-aware heuristic, Hybrid-EDF, is proposed for the dynamic scheduling of multiple real-time IoT workflows. It is achieved by scheduling tasks with low communication requirements in the cloud and communication-intensive tasks in the fog [172]. The deadline miss ratio was achieved 76.69% lower as compared to other conventional algorithms. Communication cost is at times more than already existing schemes [176].

6 DIFFERENCES BETWEEN CLOUD COMPUTING, FOG COMPUTING, CLOUD OF THINGS, AND FOG OF THINGS PARADIGMS

As discussed in the preceding paragraphs, the differences among the four paradigms are depicted in Table 6.1. The table presents the differences on basis of the following:

a) Computing model: There are two types of computing model, namely distributed and centralized. A centralized computing model consists of one server or a cluster of servers providing services. Whereas distributed computing model refers to a computing mode like that of Fog computing. Here the servers and resources are present in various separate nodes geographically distributed.

b) Third-party resources: Paradigms like Cloud are dependent on third-party resources to provide services to users. These resources are provided by enterprises outside the cloud datacenters. Fog framework instead provides resources from fog nodes.

c) Cost: The rest of the parameters discussed here like size, algorithms and resources are considered to calculate and decide the cost incurred for the respective paradigms.

d) Size: The size of a paradigm can be defined by the number and capacities of the resources used.

e) Mobility: Mobility of a framework is defined by its ability to transfer the services as demanded.

f) Time Latency: There is a time difference between requests and services. This lag is called time latency.

g) Geography: The geographical coverage of the frameworks is presented in the table.

h) Security and privacy: Security and privacy provided in each paradigm is compared.

TABLE 6.1: DIFFERENCES AMONG CLOUD COMPUTING, FOG COMPUTING, CLOUD OF THINGS AND, FOG OF THINGS

Parameters	Cloud Computing	Fog Computing	Cloud of Things	Fog of Things
Computing Model	Centralized	Distributed	Centralized	Distributed
Third Party Resources	Yes	No	Yes	No
Cost	High	Low	High	Lower than cloud computing
Size	Data centers and resources are huge	Small number of edge devices	Huge	Huge
Mobility	Multiple resources and servers, hence, high mobile applications	Less mobile	Mobile	Mobility is more than fog computing
Time Latency	High	Low as servers are closer	Lower than Cloud computing	Lower than cloud computing
Geography	Sparsely located resources and servers are present	Geographically closely located nodes	Sparsely located resources and servers and geographically connected edge nodes	Sparsely located resources and servers and geographically connected edge nodes
Security and Privacy	Lower	Higher	Lower	Higher than cloud computing

Cloud is the centralized computing framework, that lets the users work on the resources collected from various sources. The resources and servers are present over the globe, allowing availability. This allows for scalable operations and mobile applications. However, including IoT devices requires other computing frameworks for efficient operation. Hence, another computing paradigm was introduced in proximity to the edge devices. Fog computing has dedicated nodes over a specific area, distributing the tasks and requests from IoT devices among themselves. As the nodes are closer to the source of data, real-time operations can be easily carried out in this layer of computing. On the downside, if the task requested by smart devices requires high computation, fog might be unable to process the tasks within time.

To supply this need for high computation power jobs, the cloud is introduced to the edge devices, named Cloud of Things. The high transfer rate is made possible because of the higher capacity of the network and that of the cloud servers. Although, the distance of data source to cloud servers means that the delay incurred can be huge, making it unable to use the cloud for real-time operations. Also, with the increasing number of IoT devices over the globe, the data transferred to the cloud might overwhelm the fewer network channels that connect IoT to the cloud, as opposed to the dedicated channels to the closer fog nodes. Finally, the fog of things computing framework is discussed. Incorporating both, cloud and fog, the IoT devices are able to get their real-time tasks executed within time in the fog layer. And the tasks requiring higher computation are offloaded to the cloud. This framework brings together the benefits of both, Cloud of things and fog computing. Hence, the system is huge, mobile, and yet incorporates lesser cost as compared to the cloud paradigm.

7. Simulators and Real Environments

Cloud algorithms and applications need to be tested before any practical application. Otherwise, resource could be wasted in case the cloud algorithm is implemented without any prior testing. Hence, various real cloud environments and simulators are designed to test if the algorithm or framework is performing as per the required parameters. The major real clouds and simulators are discussed in detail meanwhile also depicted in Table 7.1.

Table 7.1: Comparison of various real environments

Paradigm	Environment	Name	Properties	Limitations
Cloud	**Public Cloud**	Microsoft Azure	Hybrid cloud; flexible; cost on basis of on-demand service; scalable and reliable; better security; higher availability	Platform expertise required
		Amazon Web Services	Ease of use; high-performance databases available; secure	Security limitations, higher cost
		Google Cloud Platform	Durable; lower cost; availability; live migration of VM	Higher support fee; a bit cumbersome to use
	Private Cloud	Opennebula	Flexible; Scalability; Control; Robust; ease of use; multi-hypervisor; manages heterogeneous data centers	Limited customization
		VMware Cloud	Lower cost; Easy rollback feature and adding new VM; multiple OS allowed	Platform expertise required
		OpenStack	Lower cost, ease of use, higher security and reliability, uniform standards	No organized support
Cloud IoT		Microsoft Azure IoT Suite	Scalable; Ease of use; Cheap	Uses SQL database
		Google Cloud's IoT Platform	Security; control; availability; scalability	Higher cost
		AWS IoT platform	Better GUI; ease of use; better customization; higher security; scalability	Lower performance; lesser compatibility; higher cost
Fog Computing		Docker	Consistent solution; automation; Stable	Unstable because of frequent updates
		Kubernetes	Better performance; powerful	Complex to use; needs platform expertise
		FogGuru	Better compatibility; ease of use; no delay	Not fault-tolerant

The table presents a few of the real environments discussed in the following section. Paradigms are divided into three sections and Cloud is further divided into public and private clouds.

7.1 Real Environment:

Various platforms for the cloud, fog, and IoT paradigms are available with their own solutions. These solutions provide various services for the functionalities of the framework. Depending on the need of the situation, the different platforms focus on optimizing the different parameters. Some of those platforms in the three computing paradigms are as follows:

7.1.1 Cloud: There are two types of Cloud platforms (public and private), where users can test the performance of developed or proposed algorithms. They are discussed as follows:

7.1.1.1 **Public Cloud Platforms:** These collections of solutions provide open platforms for applications to be deployed on. Since these are public cloud platforms, they are more scalable and easier to use. The following are a few of the public cloud platforms in use:

Microsoft Azure: Resources like computing power, storage and other cloud services are provided by Microsoft Azure. Microsoft Azure cloud provides its users with an open compatible platform to carry out applications using flexible resources.

Amazon Web Services: Amazon's AWS services provide multiple flexible and scalable on-demand resources with higher security. All the deployment models Cloud Computing paradigm are provided by AWS.

Google Cloud Platform: Google's Google cloud platform is a collection of services that can be used on Google's other platforms without any expertise. The platform has easy-to-use functionalities which help the users.

IBM Bluemix: This platform is the collection of cloud services provided by IBM. It allows workload management and management of SAP as well.

Eucalyptus: It is an acronym for 'Elastic Utility Computing Architecture for Linking Your Programs to Useful Systems'. Eucalyptus allows building more than one cloud.

SAP HANA Cloud: SAP HANA provides an on-premise platform as well as the ability to run applications on the cloud. It provides flexibility as well as security.

Alibaba Cloud: Alibaba Group provides public cloud services. This cloud provides cloud services with multiple database options. It allows scaling and flexibility in resources within a secure framework.

OpenStack: OpenStack public cloud services provide low-cost public cloud, scalable in nature. It is majorly used for the services in the IaaS layer.

7.1.1.2 Private Cloud Platforms: Private cloud platforms are designed for an enterprise with stricter privacy policies. These are quite a bit costlier as compared to public clouds platforms. A few private cloud platforms. A few private cloud platforms are listed as follows:

Opennebula: Opennebula provides cloud computing platforms for various services. The private cloud services of Opennebula are popular. This is because it is majorly used for an enterprise's needs and it allows managing multiple data centers in a powerful and flexible manner.

VMware Cloud: VMware provides types of private cloud computing frameworks by pooling all resources for multiple VMs. It allows management and automating VMs while operating the clouds at cheaper rates.

Dell Cloud: Dell cloud services provide consistent performance. It also allows resource and VM customization of the cloud.

Cisco Cloud Center: Cisco cloud center offers hybrid, public and private clouds. It allows deploying and managing multiple cloud frameworks on the platform. Cisco cloud is able to ensure better security and also enables customization of networks.

7.1.2 Fog: Some of the fog computing platforms are:

Docker: Docker is a framework that lets the customers use applications in containers. It provides efficient solutions for deploying the Fog environment. Applications are first download images in the local server and then subsequently deployed.

Kubernetes: Kubernetes provides a platform for deploying more than one host environment. Moreover, it allows network orchestration as well as the creation of fog components.

FogGuru: It is based on open-source Apache Flink. Fog applications are deployed on the basis of the stream processing approach. Latency is quite less and IoT too can be simulated using FogGuru.

7.1.3 CloudIoT: Some of the platforms for implementing frameworks for IoT with Cloud or Fog or both are discussed as follows*:*

Microsoft Azure IoT Suite: Microsoft Azure IoT Suite allows fast and remote connection and processing of data from wearables. The advantage it provides is the ease of use, even for non-experienced people.

Google Cloud's IoT Platform: Google cloud's IoT enables easy scalability and AI resources for IoT services. Various IoT devices provide information to Google cloud, which is then processed. For example, GPS for smartphones.

AWS IoT platform: AWS IoT platform enables the filtering of noisy information from IoT devices and applies analytics on them. This framework is renowned for the stronger security provided along with lower latency.

Cisco IoT Cloud Connect: Better connection and security are the main features of this platform for IoT simulation, used frequently in various areas. This framework offers better reliability and smart billing for optimal pricing.

7.2 Simulation Tools:

Frameworks and algorithms for Cloud and Fog environments need to be tested for their respective performances and limitations if any. It is not practically efficient to test these frameworks in the real environment, possibly hampering the real-time operations of the environment. Simulation is a better option to test the frameworks without disrupting the real-time operations of cloud and/or fog paradigms. The simulation also allows the tests to be conducted while controlling variables as required. Performance metrics affect each other. Simulation helps in studying the correlation and tradeoffs without any real implementation. The advantages of simulators are as follows:

i. No cost of installation
ii. Ease of operation.
iii. Control on variables to study the change in various parameters accordingly.

Some of the popularly used simulators are discussed as follows:

7.2.1 Simulation tools for Cloud Computing:

CloudSim: CloudSim was conceptualized to be able to model and simulate cloud computing frameworks, entities like data centers, and VM. Various algorithms, scheduling policies, and virtualization can also be modelled on CloudSim.

CloudAnalyst: CloudAnalyst is a CloudSim-based modeller and simulator that helps developers in understanding how applications and services should be distributed among the different entities in a large cloud environment.

GreenCloud: It is a simulator for deploying energy-aware cloud components like data centers. GreenCloud focuses on cloud communication and network-aware load balancing and resource allocation.

EMUSim: It is an integrated emulator and simulator, which is used to extract information from an application about its behaviour, automatically. And then, this generated information is used to attain correct simulation models.

WorkflowSim: WorkflowSim extends CloudSim, to provide another layer for workflow management. It considers the heterogeneous overheads and failures in simulations, unlike its predecessors. Meanwhile, WorkflowSim simulates with better accuracy.

7.2.2 Simulation tools for Fog Computing:

iFogSim: iFogSim is used to model and simulate IoT and Fog entities in Fog computing environments. Distributed Data Flow model is used to model applications, whereas a set of intercommunicating modules are used to build the respective applications.

YAFS: YAFS (Yet Another Fog Simulator) is a simulator in Python language, making it easier to use. It was conceptualized for the evaluation of policies, performance metrics, and routing, in Cloud/Fog networks.

FogNetSim++: Diverse network characteristics are allowed in this simulator along with resource management and mobility. FogNetSim++ facilitates easier and faster algorithms' deployment.

MyiFogSim: This simulator is an extension above iFogSim by enabling VM migration policies. Mobility, resource management, and virtual machine migration are the focus of this simulator for the Fog framework.

7.2.3 Simulation tools for Cloud-IoT:

IoTSim: Mostly CloudSim's functionalities are extended in IoTSim. Big data processing, like, MapReduce is also supported in this simulator along with the ability to simulate IoT applications.

IoTify: Various IoT applications can be simulated on IoTify simulator along with large traffic and a lot of VMs.

NetSim: It provides a self-contained environment for efficient simulation of IoT applications and services along with the cloud.

IBM Bluemix Watson Integration: It can collaborate with multiple cloud platforms. Also, big data processing is possible. Automation is also provided by IBM Bluemix Watson.

The proportion of simulators and real environments opted for the papers studied are as shown in the following mentioned figures. Fig. 7.1 depicts the proportion of real to simulated environments preferred for the articles studied. As it can be observed from Fig. 7.1, simulators are preferred more than the real environment to test the performance of algorithms. The major reason for this occurrence is that real environments are lesser customizable and are required to be paid for. Higher options for customization of parameters and environment variables, make simulators the preferable method to test the algorithms.

Fig. 7.2 represents the percentage of the various instances of simulators used. All the simulators are depicted in the chart. As per Fig. 7.2, it can be deduced that the testbed is the most preferred method to simulate frameworks, with the customized architecture of the systems. The testbed is the collection of specific appliances, mechanisms, architecture, and software, which can be connected and customized to implement the required algorithm. Following the testbed, CloudSim is the next preferable simulator. CloudSim is a java-based simulator with packages to simulate all layers and workings of the Cloud. Implemented on CloudSim, iFogSim is the next popular simulator. As CloudSim is used to implement a cloud environment, similarly iFogSim is used to implement a fog environment.

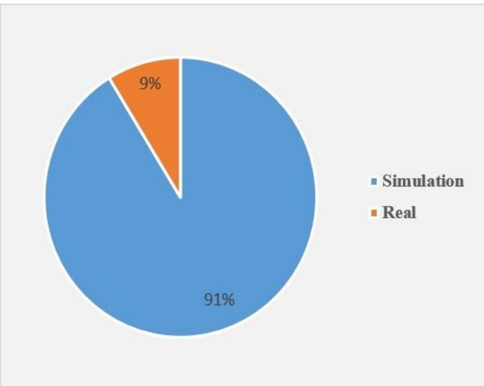

Fig. 7.1: Type of environment preferred in the studies

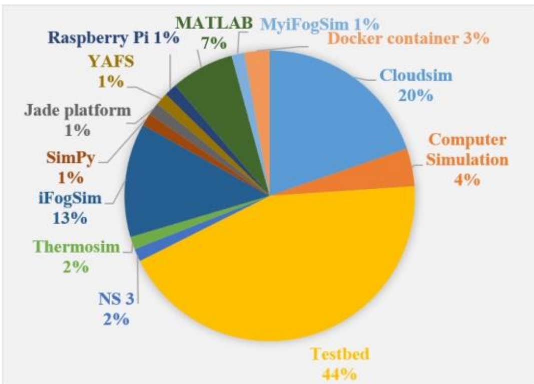

Fig. 7.2: Proportion of the instances of simulated environments

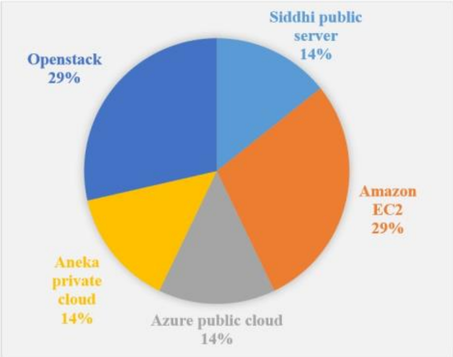

Fig. 7.3: Proportion of the instances of real environments used

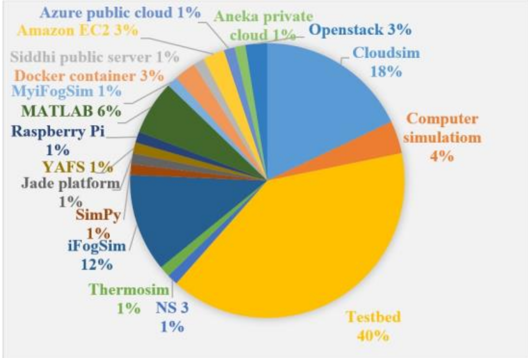

Fig. 7.4: Proportion of all types of environments and simulators

Fig. 7.3 represents the proportion of the various real environments used. As it can be observed, Amazon EC2 and OpenStack clouds are the most preferable options to implement cloud in a real environment. These cloud environments are preferred for their ease of use and flexibility. Meanwhile, Fig. 7.4 represents the proportion of all the environments and simulators used altogether. This comparison depicts that the testbed is the most preferable method of implementing frameworks and algorithms, among both simulators and real environments as observed from the studied articles.

8 FUTURE RESEARCH DIRECTIONS

The future directions in the area are discussed here as depicted in Fig. 8.1.

• **Blockchain:** The basic idea of blockchain is to maintain a public, decentralized ledger, which can be shared among the networks to verify the information transfer among entities. Since it is public and also verified, it increases security for a network. It is decentralized, and hence loss of a node does not lead to the already copied and shared ledger. The data stored in this ledger cannot be changed and it is all transparent for all the components of the network to view it as required. The motivation for integrating blockchain with cloud, fog, and IoT devices is increased availability and security. The trade-off for sharing the ledger with the participants with separate copies is reduced energy efficiency. Also, with each transaction, the size of the blockchain increases multifold, increasing the overhead. A study on these trade-offs would be able to make the integration of the cloud and the latter paradigms with the blockchain more efficient [180-181].

• **Machine Learning, Deep Learning, and Artificial Intelligence:** The data accumulated by the cloud for various tasks, can be processed by learning algorithms to learn about the users or entities. The information learned about a transaction, say, a social site, enables the analysts to learn the preferences of users on basis of their locality, age, gender, and quality of life. With the inclusion of IoT devices, this information becomes specific to different people. So, learning this information would help in increasing functionalities and services to users. Learning the correlation among the various subjects and the factors affecting them helps in future needs [182].

• **Natural Hazard:** Areas prone to natural hazards can maintain historical information regarding the disasters, impact on the local environment, planning, and helping communities in the immediate area. This information can also be collected as data regarding how many people are affected and how much help is needed. If this information is stored in the cloud, the communities for disaster management are able to act quicker with more accurate information, all the while connected to other similar communities [183].

• **Mist Computing:** Mist computing deals with the idea of placing computing nodes at the edge of the network. It reduces latency, although processing and collecting of data are carried out by the cloud. Because of the presence of intelligence at the very edge of the network, mist computing is applied to various fields like geospatial health information of patients [184,190].

• **Industrial applications:** Information transferred over, from, and to an industry can be used to collect and learn from for future needs. This information is learned and stored in the cloud. The presence of edge computing in industries can be used for production or assembly lines, as well as for services like healthcare services. The smart devices are interconnected in IIoT along with nodes to analyse the data and requests. Because of these interconnected edge devices, the automation of processes is achieved [185, 192-193].

• **Serverless Computing:** Serverless computing is the idea of employing the functions in an abstract manner. The functionalities are defined on the server by the developer without worrying about the physical resources. It is based on the idea of 'Function as a service'. Features like auto-scale, flexibility, and VM management are carried out by the service provider. Since the ecosystem is provided by the service provider, the developer needs to ensure that the environment is apt for the users' needs. Or else, users might have to face limitations of the application [181-182,186-188].

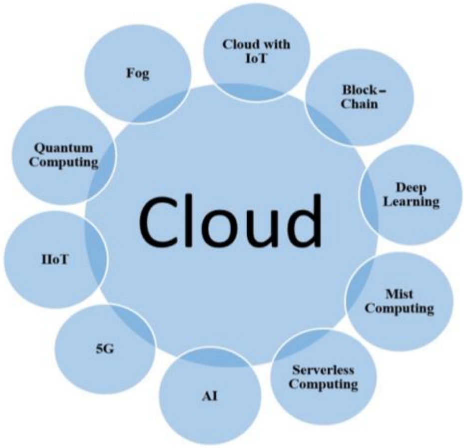

Fig. 8.1: Future Directions of Cloud

• **Quantum Computing:** Using methods of quantum mechanics, various algorithms for the cloud paradigm are enhanced [212]. With the help of neural networks, various predictions are made for resource management, VM allocation, etc. These methods are embedded with datacentres to process with as little latency and higher throughput as possible [182,191, 201-204].

• **5G:** After 4G (LTE) of mobile communication, 5G is the next generation of networks that aims for higher throughput along with reduced latency. A higher amount of data, heterogeneous in nature, can be transmitted easily using 5G. Implemented at the edge level, local data is processed faster with higher availability in presence of 5G. Although, the integration of edge and 5G introduces issues of heterogeneous communication, and gaps in privacy [182, 189, 194-200].

The future directions are discussed on the basis of emerging applications which are concluded based on the aforementioned paradigms. The results and drawbacks discussed for each paradigm would help in generalising the type of algorithms that could be worked upon and used for future directions. We hope the categorization of algorithms, paradigms and applications would be able to help in further improvement in the field.

9 CONCLUSIONS

This study reviews computing paradigms from cloud to fog to fog-cloud integrated environments and architectures. Its objective is to enlist the research gaps in these computing environments. The history of the frameworks is explained along with architecture. Architectures and models are explained in anticipation that the frameworks might be synchronized together optimally. Then, the studies are collected and compared with each other as per research challenges. We enlisted the research challenges yet faced in the framework and in which direction rigorous study will be required. The studies are compared against the performances and parameters of QoS. The trade-offs are then discussed and wherever possible, generalized. From cloud to fog and their integration, the study depicts the benefits of using these frameworks together as their limitations are reduced. Research gaps are presented against the studies' performances to work on. We expect this categorization of studies and algorithms to be able to throw light on trade-offs of parameters addressing time-optimized parameters. Simulators and real environments used in the articles are discussed in section 7. It is done and categorized to be able to detect which platform would be designed for preferred requirements. We expect this study provides an organized study of algorithms and simulators to assist in the area of cloud to the fog-cloud integrated environment.

Conflict of interest: The authors whose names are given in this article certify that they have no affiliations with or involvement in any organization or entity with any financial interest (such as honoraria; educational grants; participation in speakers' bureaus; membership, employment, consultancies, stock ownership, or other equity interest; and expert testimony or patent-licensing arrangements), or non-financial interest (such as personal or professional relationships, affiliations, knowledge or beliefs) in the subject matter or materials discussed in this manuscript.

DATA AVAILABILITY STATEMENT

Data sharing not applicable to this article as no datasets were generated or analyzed during the current study.

REFERENCES

[1] B. Alouffi, M. Hasnain, A. Alharbi, W. Alosaimi, H. Alyami, and M. Ayaz, "A Systematic Literature Review on Cloud Computing Security: Threats and Mitigation Strategies," IEEE Access, vol. 9, pp. 57792–57807, 2021.

[2] A. Sunyaev, "Cloud Computing," in Internet Computing: Principles of Distributed Systems and Emerging Internet-Based Technologies, Cham: Springer International Publishing, 2020, pp. 195–236.

[3] A. Shakarami, M. Ghobaei-Arani, A. Shahidinejad, M. Masdari, and H. Shakarami, "Data replication schemes in cloud computing: a survey," Cluster Comput., vol. 24, no. 3, pp. 2545–2579, 2021.

[4] A. Yousefpour *et al.*, "All one needs to know about fog computing and related edge computing paradigms: A complete survey," *J. Syst. Archit.*, vol. 98, no. December 2018, pp. 289–330, 2019,

[5] R. M. Gomathi, G. H. S. Krishna, E. Brumancia and Y. M. Dhas "A Survey on IoT Technologies, Evolution and Architecture," in *2018 ICCCSP*, 2018, pp. 1-5.

[6] M. S. U. Islam, A. Kumar, and Y.-C. Hu, "Context-aware scheduling in Fog computing: A survey, taxonomy, challenges and future directions," *J. Netw. Comput. Appl.*, vol. 180, p. 103008, 2021.

[7] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge Computing: Vision and Challenges," *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637–646, 2016.

[8] S. P. Singh, A. Nayyar, R. Kumar, and A. Sharma, "Fog computing : from architecture to edge computing and big data processing," *J. Supercomput.*, vol. 75, no. 4, pp. 2070–2105, 2019.

[9] A. Singh and K. Chatterjee, "Cloud security issues and challenges: A survey," J. Netw. Comput. Appl., vol. 79, no. August 2016, pp. 88–115, 2017.

[10] S. R. Gundu, C. A. Panem, and A. Thimmapuram, "Real-Time Cloud-Based Load Balance Algorithms and an Analysis," SN Comput. Sci., vol. 1, no. 4, pp. 1–9, 2020.

[11] C. Li, J. Tang, T. Ma, X. Yang, and Y. Luo, "Load balance based workflow job scheduling algorithm in distributed cloud," J. Netw. Comput. Appl., vol. 152, no. May 2019, p. 102518, 2020.

[12] G. Sridevi, "A Systematic Survey on Load Balancing in the Cloud," in Intelligent Computing and Innovation on Data Science, Springer, 2020, pp. 761–771.

[13] A. Gupta, H. S. Bhadauria, and A. Singh, "SLA-aware load balancing using risk management framework in cloud," J. Ambient Intell. Humaniz. Comput., no. 0123456789, 2020.

[14] M. V. Fard, A. Sahafi, A. M. Rahmani, and P. S. Mashhadi, "Resource allocation mechanisms in cloud computing: A systematic literature review," IET Softw., vol. 14, no. 6, pp. 638–653, 2020.

[15] D. Perez Abreu, K. Velasquez, M. Curado, and E. Monteiro, "A comparative analysis of simulators for the Cloud to Fog continuum," Simul. Model. Pract. Theory, vol. 101, no. November 2019, p. 102029, 2020.

[16] A. A. Alli and M. M. Alam, "The fog cloud of things: A survey on concepts, architecture, standards, tools, and applications," Internet of Things, vol. 9, p. 100177, 2020.

[17] R. Mahmud, K. Ramamohanarao, and R. Buyya, "Application Management in Fog Computing Environments: A Taxonomy, Review and Future Directions," ACM Comput. Surv., vol. 53, no. 4, 2020.

[18] A. A. Sadri, A. M. Rahmani, M. Saberikamarposhti, and M. Hosseinzadeh, "Fog data management: A vision, challenges, and future directions," J. Netw. Comput. Appl., vol. 174, no. November 2020, p. 102882, 2021.

[19] D. Puthal, S. P. Mohanty, S. A. Bhavake, G. Morgan, and R. Ranjan, "Fog Computing Security Challenges and Future Directions [Energy and Security]," IEEE Consum. Electron. Mag., vol. 8, no. 3, pp. 92–96, 2019.

[20] S. B. Nath, H. Gupta, S. Chakraborty, and S. K. Ghosh, "A survey of fog computing and communication: Current researches and future directions," arXiv, no. i, pp. 1–47, 2018.

[21] M. A. Aleisa, A. Abuhussein, and F. T. Sheldon, "Access Control in Fog Computing: Challenges and Research Agenda," IEEE Access, vol. 8, pp. 83986–83999, 2020.

[22] P. Hu, S. Dhelim, H. Ning, and T. Qiu, "Survey on fog computing: architecture, key technologies, applications and open issues," J. Netw. Comput. Appl., vol. 98, no. April, pp. 27–42, 2017.

[23] R. Mahmud, R. Kotagiri, and R. Buyya, "Fog Computing: A taxonomy, survey and future directions," Internet of Things, vol. 0, no. 9789811058608, pp. 103–130, 2018.

[24] C. Mouradian, D. Naboulsi, S. Yangui, R. H. Glitho, M. J. Morrow, and P. A. Polakos, "A Comprehensive Survey on Fog Computing: State-of-the-Art and Research Challenges," IEEE Commun. Surv. Tutorials, vol. 20, no. 1, pp. 416–464, 2018.

[25] M. Mukherjee, L. Shu, and D. Wang, “Survey of fog computing: Fundamental, network applications, and research challenges,” IEEE Commun. Surv. Tutorials, vol. 20, no. 3, pp. 1826–1857, 2018.
[26] P. Y. Zhang, M. C. Zhou, and G. Fortino, “Security and trust issues in Fog computing: A survey,” Futur. Gener. Comput. Syst., vol. 88, pp. 16–27, 2018.
[27] J. Dizdarević, F. Carpio, A. Jukan, and X. Masip-bruin, “A survey of communication protocols for internet of things and Related Challenges of Fog and Cloud Computing,” vol. 1, no. 1, pp. 1–30, 2019.
[28] M. S. Aslanpour, S. S. Gill, and A. N. Toosi, “Performance Evaluation Metrics for Cloud, Fog and Edge Computing: A Review, Taxonomy, Benchmarks and Standards for Future Research,” Internet of Things, p. 100273, 2020.
[29] M. Bendechache, S. Svorobej, P. T. Endo, and T. Lynn, “Simulating resource management across the cloud-to-thing continuum: A survey and future directions,” Futur. Internet, vol. 12, no. 6, pp. 1–25, 2020.
[30] A. R. Arunarani, D. Manjula, and V. Sugumaran, “Task scheduling techniques in cloud computing : A literature survey,” Futur. Gener. Comput. Syst., vol. 91, pp. 407–415, 2019.
[31] E. J. Ghomi and A. M. Rahmani, “Load-balancing algorithms in cloud computing : A survey,” J. Netw. Comput. Appl., vol. 88, no. March, pp. 50–71, 2017.
[32] P. Kumar and R. Kumar, “Issues and Challenges of Load Balancing Techniques in Cloud Computing : A Survey,” no. February, 2019.
[33] S. Singh, Y. Jeong, and J. Hyuk, “Journal of Network and Computer Applications A survey on cloud computing security : Issues , threats , and solutions,” J. Netw. Comput. Appl., vol. 75, pp. 200–222, 2016.
[34] M. Kumar, S. C. Sharma, A. Goel, and S. P. Singh, “A comprehensive survey for scheduling techniques in cloud computing,” J. Netw. Comput. Appl., vol. 143, no. June, pp. 1–33, 2019.
[35] C. Barros, V. Rocio, A. Sousa, and H. Paredes, “Scheduling in Cloud and Fog Architecture: Identification of Limitations and Suggestion of Improvement Perspectives,” J. Inf. Syst. Eng. Manag., vol. 5, no. 3, p. em0121, 2020.
[36] C.H. Hong and B. Varghese, “Resource Management in Fog/Edge Computing: A Survey on Architectures, Infrastructure, and Algorithms,” ACM Comput. Surv., vol. 52, no. 5, Sep. 2019, doi: 10.1145/3326066.
[37] V. Prokhorenko and M. Ali Babar, “Architectural resilience in cloud, fog and edge systems: A survey,” IEEE Access, vol. 8, pp. 28078–28095, 2020.
[38] Mell Peter, Timothy Grance, “The NIST Definition of Cloud Computing”, National Institute of Standards and Technology, 2011.
[39] A. Al-Qamash, I. Soliman, R. Abulibdeh, and M. Saleh, “Cloud, Fog, and Edge Computing: A Software Engineering Perspective,” 2018 Int. Conf. Comput. Appl. ICCA 2018, pp. 276–284, 2018.
[40] R. Buyya, C. Vecchiola, and S. T. Selvi, "Cloud Computing Architecture ", in Mastering Cloud Computing, Eds. Boston: Morgan Kaufmann, 2013, pp. 111-139.
[41] W. Voorsluys, J. Broberg, and R. Buyya, “Introduction to Cloud Computing,” in Cloud Computing, John Wiley & Sons, Ltd, 2011, pp. 1–41.
[42] D. Talia, “Clouds for scalable big data analytics,” Computer (Long. Beach. Calif)., vol. 46, no. 5, pp. 98–101, 2013.
[43] D. Nazir and M. A. Sheheryar, “Data Backup and Recovery Methods in Cloud Computing,” Int. J. Comput. Sci. Eng., vol. 6, no. 5, pp. 540–544, 2018.
[44] E. Gialinou, C. Drosos, M. Papoutsidakis, and K. Kalovrektis, “Study and Analysis of a ‘Disaster Recovery’ Information System using Cloud-computing Technology,” Int. J. Comput. Appl., vol. 177, no. 23, pp. 1–7, 2019.
[45] C. Stergiou, K. E. Psannis, B. G. Kim, and B. Gupta, “Secure integration of IoT and Cloud Computing,” Futur. Gener. Comput. Syst., vol. 78, pp. 964–975, 2018.
[46] A. Praveena and S. Smys, “Ensuring data security in cloud based social networks,” Proc. Int. Conf. Electron. Commun. Aerosp. Technol. ICECA 2017, vol. 2017-January, pp. 289–295, 2017.
[47] S. Ashtari and A. Eydgahi, “Student perceptions of cloud applications effectiveness in higher education,” J. Comput. Sci., vol. 23, pp. 173–180, 2017.
[48] S. J. Miah, J. Hasan, and J. G. Gammack, “On-Cloud Healthcare Clinic: An e-health consultancy approach for remote communities in a developing country,” Telemat. Informatics, vol. 34, no. 1, pp. 311–322, 2017.
[49] M. S. Mekala and P. Viswanathan, “A Survey: Smart agriculture IoT with cloud computing,” 2017 Int. Conf. Microelectron. Devices, Circuits Syst. ICMDCS 2017, vol. 2017-January, pp. 1–7, 2017.
[50] S. Namasudra, D. Devi, S. Kadry, R. Sundarasekar, and A. Shanthini, “Towards DNA based data security in the cloud computing environment,” Comput. Commun., vol. 151, no. January, pp. 539–547, 2020.
[51] B. K. Dewangan, A. Agarwal, M. Venkatadri, and A. Pasricha, “Self-characteristics based energy-efficient resource scheduling for cloud,” Procedia Comput. Sci., vol. 152, pp. 204–211, 2019.
[52] S. S. Gill, R. Buyya, I. Chana, M. Singh, and A. Abraham, “BULLET: Particle Swarm Optimization Based Scheduling Technique for Provisioned Cloud Resources,” J. Netw. Syst. Manag., vol. 26, no. 2, pp. 361–400, 2018.
[53] N. Malarvizhi, G. S. Priyatharsini, and S. Koteeswaran, “Cloud Resource Scheduling Optimal Hypervisor (CRSOH) for Dynamic Cloud Computing Environment,” Wirel. Pers. Commun., no. 0123456789, 2020.
[54] V. Priya, C. S. Kumar, and R. Kannan, “Resource scheduling algorithm with load balancing for cloud service provisioning,” Appl. Soft Comput. J., vol. 76, pp. 416–424, 2019.
[55] W. Zhu, Y. Zhuang, and L. Zhang, “A three-dimensional virtual resource scheduling method for energy saving in cloud computing,” Futur. Gener. Comput. Syst., vol. 69, pp. 66–74, 2017.
[56] H. A. Kholidy, “An Intelligent Swarm Based Prediction Approach For Predicting Cloud Computing User Resource Needs,” Comput. Commun., vol. 151, no. October 2019, pp. 133–144, 2020.
[57] S. Kim, “A New Adaptive Data Center Resource Provisioning Scheme Based on the Dual-Level Cooperative Game Approach,” IEEE Access, vol. 6, pp. 52047–52057, 2018.

[58] Sukhpal Singh Gill, Shreshth Tuli, Adel Nadjaran Toosi, Felix Cuadrado, Peter Garraghan, Rami Bahsoon, Hanan Lutfiyya, Rizos Sakellariou, Omer Rana, Schahram Dustdar, and Rajkumar Buyya. 2020. ThermoSim: Deep Learning based Framework for Modeling and Simulation of Thermal-aware Resource Management for Cloud Computing Environments. J. Syst. Softw. (2020), 110596.
[59] S. Tuli, R. Sandhu, and R. Buyya, "Shared data-aware dynamic resource provisioning and task scheduling for data intensive applications on hybrid clouds using Aneka," Futur. Gener. Comput. Syst., vol. 106, pp. 595–606, 2020.
[60] T. Uchibayashi, B. O. Apduhan, N. Shiratori, T. Suganuma, and M. Hiji, "Policy management technique using blockchain for cloud VM migration," Proc. - IEEE 17th Int. Conf. Dependable, Auton. Secur. Comput. IEEE 17th Int. Conf. Pervasive Intell. Comput. IEEE 5th Int. Conf. Cloud Big Data Comput. 4th Cyber Sci., pp. 360–362, 2019.
[61] L. Li, J. Dong, D. Zuo, and J. Wu, "SLA-Aware and Energy-Efficient VM Consolidation in Cloud Data Centers Using Robust Linear Regression Prediction Model," IEEE Access, vol. 7, pp. 9490–9500, 2019.
[62] X. Liu, C. Xia, T. Wang, L. Zhong, and X. Li, "A behavior-aware SLA-based framework for guaranteeing the security conformance of cloud service," Front. Comput. Sci., vol. 14, no. 6, 2020.
[63] R. Mandal, M. K. Mondal, S. Banerjee, and U. Biswas, "An approach toward design and development of an energy-aware VM selection policy with improved SLA violation in the domain of green cloud computing," J. Supercomput., no. 0123456789, 2020.
[64] Y. Wang, X. Tao, F. Zhao, B. Tian, and A. M. Vera Venkata Sai, "SLA-aware resource scheduling algorithm for cloud storage," Eurasip J. Wirel. Commun. Netw., vol. 2020, no. 1, 2020.
[65] T. Guo, P. Shenoy, K. K. Ramakrishnan, and V. Gopalakrishnan, "Latency-aware virtual desktops optimization in distributed clouds," Multimed. Syst., vol. 24, no. 1, pp. 73–94, 2018.
[66] H. Li, C. Lu, and C. Gill, "Predicting latency distributions of aperiodic time-critical services," Proc. - Real-Time Syst. Symp., vol. 2019-Decem, pp. 30–42, 2019.
[67] A. Rodrigo, M. Dayarathna, and S. Jayasena, "Latency-Aware Secure Elastic Stream Processing with Homomorphic Encryption," Data Sci. Eng., vol. 4, no. 3, pp. 223–239, 2019.
[68] M. Naghshnejad and M. Singhal, "Adaptive Online Runtime Prediction to Improve HPC Applications Latency in Cloud," IEEE Int. Conf. Cloud Comput. CLOUD, vol. 2018-July, pp. 762–769, 2018.
[69] A. Marahatta, Y. Wang, F. Zhang, A. K. Sangaiah, S. K. S. Tyagi, and Z. Liu, "Energy-Aware Fault-Tolerant Dynamic Task Scheduling Scheme for Virtualized Cloud Data Centers," Mob. Networks Appl., vol. 24, no. 3, pp. 1063–1077, 2019.
[70] S. Nasirian and F. Faghani, "Crystal: A scalable and fault-tolerant Archimedean-based server-centric cloud data center network architecture," Comput. Commun., vol. 147, no. August, pp. 159–179, 2019.
[71] V. M. Sivagami and K. S. Easwarakumar, "An Improved Dynamic Fault Tolerant Management Algorithm during VM migration in Cloud Data Center," Futur. Gener. Comput. Syst., vol. 98, pp. 35–43, 2019.
[72] H. Yan, X. Zhu, H. Chen, H. Guo, W. Zhou, and W. Bao, "DEFT: Dynamic Fault-Tolerant Elastic scheduling for tasks with uncertain runtime in cloud," Inf. Sci. (Ny)., vol. 477, pp. 30–46, 2019.
[73] D. Armstrong, K. Djemame, and R. Kavanagh, "Towards energy aware cloud computing application construction," J. Cloud Comput., vol. 6, no. 1, 2017.
[74] X. Zhang, T. Wu, M. Chen, T. Wei, J. Zhou, S. Hu, R. Buyya, "Energy-aware virtual machine allocation for cloud with resource reservation," J. Syst. Softw., vol. 147, pp. 147–161, 2019.
[75] M. Kumar and S. C. Sharma, "PSO-COGENT: Cost and energy efficient scheduling in cloud environment with deadline constraint," Sustain. Comput. Informatics Syst., vol. 19, no. January, pp. 147–164, 2018.
[76] S. K. Mishra, D. Puthal, B. Sahoo, P. P. Jayaramam, S. Jun, A. Y. Zomaya, R. Ranjan, "Energy-efficient VM-placement in cloud data center," Sustain. Comput. Informatics Syst., vol. 20, pp. 48–55, 2018.
[77] Y. Alahmad, T. Daradkeh, and A. Agarwal, "Availability-Aware Container Scheduler for Application Services in Cloud," 2018 IEEE 37th Int. Perform. Comput. Commun. Conf. IPCCC 2018, 2018.
[78] A. Londhe, V. Bhalerao, S. Ghodey, S. Kate, N. Dandekar, and S. Bhange, "Data Division and Replication Approach for Improving Security and Availability of Cloud Storage," Proc. - 2018 4th Int. Conf. Comput. Commun. Control Autom. ICCUBEA 2018, pp. 2018–2021, 2018.
[79] T. M. Mengistu, D. Che, A. Alahmadi, and S. Lu, "Semi-Markov Process Based Reliability and Availability Prediction for Volunteer Cloud Systems," IEEE Int. Conf. Cloud Comput. CLOUD, vol. 2018-July, pp. 359–366, 2018.
[80] Y. Hassanzadeh-Nazarabadi, A. Küpçü, and Ö. Özkasap, "Awake: Decentralized and Availability Aware Replication for P2P Cloud Storage," Proc. - 2016 IEEE Int. Conf. Smart Cloud, SmartCloud 2016, pp. 289–294, 2016.
[81] M. A. Khoshkholghi, M. N. Derahman, A. Abdullah, S. Subramaniam, and M. Othman, "Energy-Efficient Algorithms for Dynamic Virtual Machine Consolidation in Cloud Data Centers," IEEE Access, vol. 5, no. November, pp. 10709–10722, 2017.
[82] Z. Jiang and S. Mao, "Energy Delay Tradeoff in Cloud Offloading for Multi-Core Mobile Devices," IEEE Access, vol. 3, no. July, pp. 2306–2316, 2015.
[83] S. Mubeen, P. Nikolaidis, A. DIdic, H. Pei-Breivold, K. Sandstrom, and M. Behnam, "Delay Mitigation in Offloaded Cloud Controllers in Industrial IoT," IEEE Access, vol. 5, pp. 4418–4430, 2017.
[84] S. Chenthara, K. Ahmed, H. Wang, and F. Whittaker, "Security and Privacy-Preserving Challenges of e-Health Solutions in Cloud Computing," IEEE Access, vol. 7, pp. 74361–74382, 2019.
[85] V. Eramo and F. G. Lavacca, "Optimizing the Cloud Resources, Bandwidth and Deployment Costs in Multi-Providers Network Function Virtualization Environment," IEEE Access, vol. 7, pp. 46898–46916, 2019.
[86] V. Haghighi and N. S. Moayedian, "An Offloading Strategy in Mobile Cloud Computing Considering Energy and Delay Constraints," IEEE Access, vol. 6, pp. 11849–11861, 2018.
[87] S. Krishnaveni and S. Prabakaran, "Ensemble approach for network threat detection and classification on cloud computing," Concurr. Comput. Pract. Exp., vol. 33, no. 3, p. e5272, 2021.
[88] S. K. Mishra, S. Mishra, S. K. Bharti, B. Sahoo, D. Puthal, and M. Kumar, "VM Selection using DVFS Technique to Minimize Energy Consumption in Cloud System," Proc. - 2018 Int. Conf. Inf. Technol. ICIT 2018, pp. 284–289, 2018.

[89] M. Kumar and S. C. Sharma, "Load balancing algorithm to minimize the makespan time in cloud environment," vol. 14, no. 4, pp. 276–288, 2018.
[90] M. Kumar and S. C. Sharma, "Deadline constrained based dynamic load balancing algorithm with elasticity in cloud environment," Comput. Electr. Eng., vol. 69, pp. 395–411, 2018.
[91] M. Kumar, K. Dubey, and S. C. Sharma, "Job Scheduling Algorithm in Cloud Environment Considering the Priority and Cost of Job," in Proc. SocProS2016S, pp. 313–320, 2017.
[92] K. Dubey, M. Kumar, and M. A. Chandra, "A priority based job scheduling algorithm using IBA and EASY algorithm for cloud metaschedular," Conf. Proceeding - 2015 Int. Conf. Adv. Comput. Eng. Appl. ICACEA 2015, pp. 66–70, 2015.
[93] H. F. Atlam, A. Alenezi, A. Alharthi, R. J. Walters, and G. B. Wills, "Integration of cloud computing with internet of things: Challenges and open issues," Proc. - 2017 IEEE Int. Conf. Internet Things, IEEE Green Comput. Commun. IEEE Cyber, Phys. Soc. Comput. IEEE Smart Data, iThings-GreenCom-CPSCom-SmartData 2017, vol. 2018-January, pp. 670–675, 2018.
[94] G. Neagu, S. Preda, A. Stanciu, and V. Florian, "A Cloud-IoT based sensing service for health monitoring," 2017 E-Health Bioeng. Conf. EHB 2017, pp. 53–56, 2017.
[95] M. A. O. Pessoa, M. A. Pisching, L. Yao, F. Junqueira, P. E. Miyagi, and B. Benatallah, "Industry 4.0, how to integrate legacy devices: A cloud IoT approach," Proc. IECON 2018 - 44th Annu. Conf. IEEE Ind. Electron. Soc., pp. 2902–2907, 2018.
[96] M. Tota, "Big Data Privacy and Security Risk and Solutions," sensors, vol. 4, no. 1, 2021.
[97] M. Tao, J. Zuo, Z. Liu, A. Castiglione, and F. Palmieri, "Multi-layer cloud architectural model and ontology-based security service framework for IoT-based smart homes," Futur. Gener. Comput. Syst., vol. 78, pp. 1040–1051, 2018.
[98] Z. Guan, J. Li, L. Wu, Y. Zhang, J. Wu, and X. Du, "Achieving Efficient and Secure Data Acquisition for Cloud-Supported Internet of Things in Smart Grid," IEEE Internet Things J., vol. 4, no. 6, pp. 1934–1944, 2017.
[99] W. Tärneberg, V. Chandrasekaran, and M. Humphrey, "Experiences creating a framework for smart traffic control using AWS IOT," Proc. - 9th IEEE/ACM Int. Conf. Util. Cloud Comput. UCC 2016, pp. 63–69, 2016.
[100] R. Dagar, S. Som, and S. K. Khatri, "Smart Farming - IoT in Agriculture," Proc. Int. Conf. Inven. Res. Comput. Appl. ICIRCA 2018, no. Icirca, pp. 1052–1056, 2018.
[101] Y. Nan, W. Li, W. Bao, F. C. Delicato, P. F. Pires, Y. Dou, A. Y. Zomaya, "Adaptive Energy-Aware Computation Offloading for Cloud of Things Systems," IEEE Access, vol. 5, pp. 23947–23957, 2017.
[102] M. M. E. Mahmoud, J. J. P. C. Rodrigues, K. Saleem, J. Al-Muhtadi, N. Kumar, and V. Korotaev, "Towards energy-aware fog-enabled cloud of things for healthcare," Comput. Electr. Eng., vol. 67, pp. 58–69, 2018.
[103] Z. Ning, X. Kong, F. Xia, W. Hou, and X. Wang, "Green and Sustainable Cloud of Things: Enabling Collaborative Edge Computing," IEEE Commun. Mag., vol. 57, no. 1, pp. 72–78, 2019.
[104] Y. Nan, W. Li, W. Bao, F. C. Delicato, P. F. Pires, and A. Y. Zomaya, "A dynamic tradeoff data processing framework for delay-sensitive applications in Cloud of Things systems," J. Parallel Distrib. Comput., vol. 112, pp. 53–66, 2018.
[105] T. Khan, S. Ghosh, M. Iqbal, G. Ubakanma, and T. Dagiuklas, "RESCUE: A Resilient Cloud Based IoT System for Emergency and Disaster Recovery," Proc. - 20th Int. Conf. High Perform. Comput. Commun. 16th Int. Conf. Smart City 4th Int. Conf. Data Sci. Syst. HPCC/SmartCity/DSS 2018, pp. 1043–1047, 2019.
[106] R. Hasan, M. Hossain, and R. Khan, "Aura: An incentive-driven ad-hoc IoT cloud framework for proximal mobile computation offloading," Futur. Gener. Comput. Syst., vol. 86, pp. 821–835, 2018.
[107] M. Jia, Z. Yin, D. Li, Q. Guo, and X. Gu, "Toward improved offloading efficiency of data transmission in the IoT-cloud by leveraging secure truncating OFDM," IEEE Internet Things J., vol. 6, no. 3, pp. 4252–4261, 2019.
[108] H. W. Kim, J. H. Park, and Y. S. Jeong, "Efficient Resource Management Scheme for Storage Processing in Cloud Infrastructure with Internet of Things," Wirel. Pers. Commun., vol. 91, no. 4, pp. 1635–1651, 2016.
[109] B. Maati and D. E. Saidouni, "CIoTAS protocol: CloudIoT available services protocol through autonomic computing against distributed denial of services attacks," J. Ambient Intell. Humaniz. Comput., no. 2019, 2020.
[110] G. Sharma and S. Kalra, "Advanced lightweight multi-factor remote user authentication scheme for cloud-IoT applications," J. Ambient Intell. Humaniz. Comput., vol. 11, no. 4, pp. 1771–1794, 2020.
[111] M. Wazid, A. K. Das, V. Bhat K, and A. V. Vasilakos, "LAM-CIoT: Lightweight authentication mechanism in cloud-based IoT environment," J. Netw. Comput. Appl., vol. 150, no. July 2018, p. 102496, 2020.
[112] F. Nawaz, N. K. Janjua, O. K. Hussain, F. K. Hussain, E. Chang, and M. Saberi, "Event-driven approach for predictive and proactive management of SLA violations in the Cloud of Things," Futur. Gener. Comput. Syst., vol. 84, pp. 78–97, 2018.
[113] H. Khodkari, S. Ghazi-Maghrebi, and A. Asosheh, "Assurance of QoS in the integration of cloud services and internet of things," 2017 Int. Symp. Networks, Comput. Commun. ISNCC 2017, 2017.
[114] F. Nawaz, O. K. Hussain, N. Janjua, and E. Chang, "A proactive event-driven approach for dynamic QoS compliance in cloud of things," Proc. - 2017 IEEE/WIC/ACM Int. Conf. Web Intell. WI 2017, no. June 2019, pp. 971–975, 2017.
[115] P. Asghari, A. M. Rahmani, and H. H. S. Javadi, "Privacy-aware cloud service composition based on QoS optimization in Internet of Things," J. Ambient Intell. Humaniz. Comput., no. 0123456789, 2020.
[116] C. Badii, P. Bellini, A. Difino, and P. Nesi, "Sii-mobility: An IoT/IoE architecture to enhance smart city mobility and transportation services," Sensors (Switzerland), vol. 19, no. 1, 2019.
[117] H. M. Al-Kadhim and H. S. Al-Raweshidy, "Energy efficient and reliable transport of data in cloud-based IoT," IEEE Access, vol. 7, pp. 64641–64650, 2019.
[118] T. G. Nguyen, T. V. Phan, B. T. Nguyen, C. So-In, Z. A. Baig, and S. Sanguanpong, "SeArch: A Collaborative and Intelligent NIDS Architecture for SDN-Based Cloud IoT Networks," IEEE Access, vol. 7, pp. 107678–107694, 2019.
[119] C. Stergiou, K. E. Psannis, B. B. Gupta, and Y. Ishibashi, "Security, privacy & efficiency of sustainable Cloud Computing for Big Data & IoT," Sustain. Comput. Informatics Syst., vol. 19, no. May, pp. 174–184, 2018.
[120] M. Al-Zihad, S. A. Akash, T. Adhikary, and M. A. Razzaque, "Bandwidth allocation and computation offloading for service specific IoT edge devices," 5th IEEE Reg. 10 Humanit. Technol. Conf. 2017, R10-HTC 2017, vol. 2018-January, pp. 516–519, 2018.
[121] Z. Á. Mann, "Notions of architecture in fog computing," Computing, vol. 103, no. 1, pp. 51–73, 2021.

[122] S. Yi, C. Li, and Q. Li, "A survey of fog computing: Concepts, applications and issues," Proc. Int. Symp. Mob. Ad Hoc Netw. Comput., vol. 2015-June, no. June 2015, pp. 37–42, 2015.
[123] P. Hu, S. Dhelim, H. Ning, and T. Qiu, "Survey on fog computing: architecture, key technologies, applications and open issues," J. Netw. Comput. Appl., vol. 98, no. April, pp. 27–42, 2017.
[124] Y. I. Alzoubi, V. H. Osmanaj, A. Jaradat, and A. Al-Ahmad, "Fog computing security and privacy for the Internet of Thing applications: State-of-the-art," Secur. Priv., vol. 4, no. 2, p. e145, Mar. 2021.
[125] S. Kumari, S. Singh, and M. April, "Fog Computing : Characteristics and Challenges," vol. 6, no. 2, pp. 113–117, 2017.
[126] S. Yi, Z. Hao, Z. Qin, and Q. Li, "Fog Computing : Platform and Applications," no. August, 2016.
[127] I. Lera, C. Guerrero, and C. Juiz, "Availability-aware service placement policy in fog computing based on graph partitions," IEEE Internet Things J., vol. 6, no. 2, pp. 3641–3651, 2019
[128] A. Mseddi, W. Jaafar, H. Elbiaze, and W. Ajib, "Joint Container Placement and Task Provisioning in Dynamic Fog Computing," IEEE Internet Things J., vol. 6, no. 6, pp. 10028–10040, 2019.
[129] Z. Pooranian, M. Shojafar, P. G. V. Naranjo, L. Chiaraviglio, and M. Conti, "A Novel Distributed Fog-Based Networked Architecture to Preserve Energy in Fog Data Centers," Proc. - 14th IEEE Int. Conf. Mob. Ad Hoc Sens. Syst. MASS 2017, pp. 604–609, 2017.
[130] R. Oma, S. Nakamura, D. Duolikun, T. Enokido, and M. Takizawa, "An energy-efficient model for fog computing in the Internet of Things (IoT)," Internet of Things, vol. 1–2, pp. 14–26, 2018.
[131] Y. Xiao and M. Krunz, "Distributed optimization for energy-efficient fog computing in the tactile internet," IEEE J. Sel. Areas Commun., vol. 36, no. 11, pp. 2390–2400, 2018.
[132] A. Karimiafshar, M. R. Hashemi, M. R. Heidarpour, and A. N. Toosi, "Effective Utilization of Renewable Energy Sources in Fog Computing Environment via Frequency and Modulation Level Scaling," IEEE Internet Things J., vol. 4662, no. c, pp. 1–1, 2020.
[133] J. wen Xu, K. Ota, M. xiong Dong, A. feng Liu, and Q. Li, "SIoTFog: Byzantine-resilient IoT fog networking," Front. Inf. Technol. Electron. Eng., vol. 19, no. 12, pp. 1546–1557, 2018.
[134] K. Wang, Y. Shao, L. Xie, J. Wu, and S. Guo, "Adaptive and Fault-Tolerant Data Processing in Healthcare IoT Based on Fog Computing," IEEE Trans. Netw. Sci. Eng., vol. 7, no. 1, pp. 263–273, 2020.
[135] R. Mahmud, K. Ramamohanarao, and R. Buyya, "Latency-aware application module management for fog computing environments," ACM Trans. Internet Technol., vol. 19, no. 1, 2018.
[136] Q. D. La, M. V. Ngo, T. Q. Dinh, T. Q. S. Quek, and H. Shin, "Enabling intelligence in fog computing to achieve energy and latency reduction," Digit. Commun. Networks, vol. 5, no. 1, pp. 3–9, 2019.
[137] I. Martinez, A. Jarray, and A. S. Hafid, "Scalable design and dimensioning of fog-computing infrastructure to support latency-sensitive IoT applications," IEEE Internet Things J., vol. 7, no. 6, pp. 5504–5520, 2020.
[138] M. Mukherjee, S. Kumar, C. X. Mavromoustakis, G. Mastorakis, R. Matam, V. Kumar, Q. Zhang , "Latency-Driven Parallel Task Data Offloading in Fog Computing Networks for Industrial Applications," IEEE Trans. Ind. Informatics, vol. 16, no. 9, pp. 6050–6058, 2020.
[139] J. P. Martin, A. Kandasamy, and K. Chandrasekaran, "Mobility aware autonomic approach for the migration of application modules in fog computing environment," J. Ambient Intell. Humaniz. Comput., no. 0123456789, 2020.
[140] O. Skarlat, M. Nardelli, S. Schulte, and S. Dustdar, "Towards QoS-Aware Fog Service Placement," Proc. - 2017 IEEE 1st Int. Conf. Fog Edge Comput. ICFEC 2017, pp. 89–96, 2017.
[141] K. Cao, J. Zhou, G. Xu, T. Wei, and S. Hu, "Exploring Renewable-Adaptive Computation Offloading for Hierarchical QoS Optimization in Fog Computing," IEEE Trans. Comput. Des. Integr. Circuits Syst., vol. 0070, no. c, pp. 1–1, 2019.
[142] C. H. Hong, K. Lee, M. Kang, and C. Yoo, "QCon: QoS-aware network resource management for fog computing," Sensors (Switzerland), vol. 18, no. 10, pp. 1–21, 2018.
[143] Y. Sun, F. Lin, and H. Xu, "Multi-objective Optimization of Resource Scheduling in Fog Computing Using an Improved NSGA-II," Wirel. Pers. Commun., vol. 102, no. 2, pp. 1369–1385, 2018.
[144] G. Li, Y. Liu, J. Wu, D. Lin, and S. Zhao, "Methods of resource scheduling based on optimized fuzzy clustering in fog computing," Sensors (Switzerland), vol. 19, no. 9, 2019.
[145] H. Rafique, M. A. Shah, S. U. Islam, T. Maqsood, S. Khan, and C. Maple, "A Novel Bio-Inspired Hybrid Algorithm (NBIHA) for Efficient Resource Management in Fog Computing," IEEE Access, vol. 7, pp. 115760–115773, 2019.
[146] J. Yao and N. Ansari, "Fog Resource Provisioning in Reliability-Aware IoT Networks," IEEE Internet Things J., vol. 6, no. 5, pp. 8262–8269, 2019.
[147] J. Santos, T. Wauters, B. Volckaert, and F. De Turck, "Resource provisioning in fog computing: From theory to practice," Sensors (Switzerland), vol. 19, no. 10, pp. 1–25, 2019.
[148] S. Feng, Z. Xiong, D. Niyato, and P. Wang, "Dynamic Resource Management to Defend Against Advanced Persistent Threats in Fog Computing: A Game Theoretic Approach," IEEE Trans. Cloud Comput., vol. 7161, no. c, pp. 1–12, 2019.
[149] W. Ben Daoud, M. S. Obaidat, A. Meddeb-Makhlouf, F. Zarai, and K. F. Hsiao, "TACRM: trust access control and resource management mechanism in fog computing," Human-centric Comput. Inf. Sci., vol. 9, no. 1, 2019.
[150] H. K. Gill, V. K. Sehgal, and A. K. Verma, "A context sensitive security framework for Enterprise multimedia placement in fog computing environment," Multimed. Tools Appl., vol. 79, no. 15–16, pp. 10733–10749, 2020.
[151] S. A. Hussein, A. I. Saleh, H. E. D. Mostafa, and M. I. Obaya, "A hybrid security strategy (HS2) for reliable video streaming in fog computing," Wirel. Networks, vol. 26, no. 2, pp. 1389–1416, 2020.
[152] M. Mukherjee, R. Matam, L. Shu, L. Maglaras, M. A. Ferrang, N. Choudhary, V. Kumar, "Security and Privacy in Fog Computing: Challenges," IEEE Access, vol. 5, pp. 19293–19304, 2017.
[153] E. Balevi and R. D. Gitlin, "Optimizing the Number of Fog Nodes for Cloud-Fog-Thing Networks," IEEE Access, vol. 6, pp. 11173–11183, 2018.
[154] M. M. Shahriar Maswood, M. R. Rahman, A. G. Alharbi, and D. Medhi, "A Novel Strategy to Achieve Bandwidth Cost Reduction and Load Balancing in a Cooperative Three-Layer Fog-Cloud Computing Environment," IEEE Access, vol. 8, pp. 113737–113750, 2020

[155] N. N. Dao, J. Lee, D. N. Vu, J. Paek, J. Kim, S. Cho, K.S. Chung, C. Keum, “Adaptive Resource Balancing for Serviceability Maximization in Fog Radio Access Networks,” IEEE Access, vol. 5, pp. 14548–14559, 2017
[156] H. Sun, H. Yu, G. Fan, and L. Chen, “Energy and time efficient task offloading and resource allocation on the generic IoT-fog-cloud architecture,” Peer-to-Peer Netw. Appl., vol. 13, no. 2, pp. 548–563, 2020
[157] L. Bittencourt et al., “The Internet of Things, Fog and Cloud continuum: Integration and challenges,” Internet of Things, vol. 3–4, pp. 134–155, 2018.
[158] N. Moustafa, “A Systemic IoT-Fog-Cloud Architecture for Big-Data Analytics and Cyber Security Systems: A Review of Fog Computing,” arXiv Prepr. arXiv1906.01055, 2019, [Online]. Available: https://arxiv.org/abs/1906.01055.
[159] S. Zahoor, S. Javaid, N. Javaid, M. Ashraf, F. Ishmanov, and M. K. Afzal, “Cloud-fog-based smart grid model for efficient resource management,” Sustain., vol. 10, no. 6, pp. 1–21, 2018.
[160] F. Y. Okay and S. Ozdemir, “Routing in Fog-Enabled IoT Platforms: A Survey and an SDN-Based Solution,” IEEE Internet Things J., vol. 5, no. 6, pp. 4871–4889, 2018.
[161] G. Caiza, M. Saeteros, W. Oñate, and M. V. Garcia, “Fog computing at industrial level, architecture, latency, energy, and security: A review,” Heliyon, vol. 6, no. 4, p. e03706, 2020.
[162] R. Deng, R. Lu, S. Member, C. Lai, T. H. Luan, and H. Liang, “Optimal Workload Allocation in Fog-Cloud Computing Toward Balanced Delay and Power Consumption,” IEEE Internet of Things Journal, vol. 3, no. 6, pp. 1171–1181, Dec. 2016.
[163] M. Adhikari and H. Gianey, “Internet of Things Energy efficient offloading strategy in fog-cloud environment for IoT applications,” Internet of Things, vol. 6, p. 100053, 2019.
[164] J. Du, L. Zhao, X. Chu, “Enabling Low-Latency Applications in LTE-A Based Mixed Fog / Cloud Computing Systems,” IEEE Transactions on Vehicular Technology, vol. 68, no. 2, pp. 1757–1771, 2019.
[165] M. Abbasi, E. Mohammadi Pasand, and M. R. Khosravi, “Workload Allocation in IoT-Fog-Cloud Architecture Using a Multi-Objective Genetic Algorithm,” J. Grid Comput., vol. 18, no. 1, pp. 43–56, 2020.
[166] J. Yang, “Low-latency cloud-fog network architecture and its load balancing strategy for medical big data,” J. Ambient Intell. Humaniz. Comput., 2020.
[167] S. K. Sood, “SNA based QoS and reliability in fog and cloud framework,” no. October 2017, pp. 1601–1616, 2018.
[168] M. Emami and K. Saeed, “A modified water cycle evolutionary game theory algorithm to utilize QoS for IoT services in cloud - assisted fog computing environments,” J. Supercomput., vol. 76, no. 7, pp. 5578–5608, 2020.
[169] M. Taneja and A. Davy, “Resource aware placement of IoT application modules in Fog-Cloud Computing Paradigm,” Proc. IM 2017 - 2017 IFIP/IEEE Int. Symp. Integr. Netw. Serv. Manag., pp. 1222–1228, 2017.
[170] J. Du, L. Zhao, J. Feng, and X. Chu, “Computation Offloading and Resource Allocation in Mixed Fog / Cloud Computing Systems With Min-Max Fairness Guarantee,” IEEE Transactions on Communications , vol. 66, no. 4, pp. 1594–1608, April 2018.
[171] R. A. C. da Silva and N. L. S. d. Fonseca, “Resource Allocation Mechanism for a Fog-Cloud Infrastructure,” in 2018 IEEE ICC, pp. 1–6, 2018.
[172] G. L. Stavrinides and H. D. Karatza, “A hybrid approach to scheduling real-time IoT workflows in fog and cloud environments,” Multimed. Tools Appl., vol. 78, no. 17, pp. 24639–24655, 2019.
[173] K. Fan, J. Wang, X. Wang, H. Li, and Y. Yang, “A Secure and Verifiable Outsourced Access Control Scheme in Fog-Cloud Computing,” Sensors, 2017.
[174] A. A. Abd El-latif, B. Abd-el-atty, M. S. Hossain, and S. Member, “Secure Quantum Steganography Protocol for Fog Cloud Internet of Things,” IEEE Access, vol. 6, pp. 10332–10340, 2018.
[175] A. A. Alli and M. Mahbub, “Internet of Things SecOFF-FCIoT : Machine learning based secure offloading in Fog-Cloud of things for smart city applications,” Internet of Things, vol. 7, no. 2019, p. 100070, 2021.
[176] J. P. D. Comput, R. Amin, S. Kunal, A. Saha, D. Das, and A. Alamri, “CFSec : Password based secure communication protocol in cloud-fog environment,” J. Parallel Distrib. Comput., vol. 140, pp. 52–62, 2020.
[177] X. Meng, W. Wang, and Z. Zhang, “Delay-Constrained Hybrid Computation Offloading with Cloud and Fog Computing,” IEEE Access, vol. 5, pp. 21355–21367, 2017.
[178] P. K. Sharma, M. Y. Chen, and J. H. Park, “A Software Defined Fog Node Based Distributed Blockchain Cloud Architecture for IoT,” IEEE Access, vol. 6, pp. 115–124, 2018.
[179] M. Abedi and M. Pourkiani, “Resource allocation in combined fog-cloud scenarios by using artificial intelligence,” in 2020 Fifth International Conference on Fog and Mobile Edge Computing (FMEC), pp. 218–222, 2020.
[180] D. C. Nguyen, P. N. Pathirana, M. Ding, and A. Seneviratne, “Integration of Blockchain and Cloud of Things: Architecture, Applications and Challenges,” IEEE Commun. Surv. Tutorials, vol. 22, no. 4, pp. 2521–2549, 2020,.
[181] R. Buyya et al., “A Manifesto for Future Generation Cloud Computing: Research Directions for the Next Decade,” ACM Comput. Surv., vol. 51, no. 5, Nov. 2018.
[182] S. S. Gill et al., “Transformative effects of IoT, blockchain and artificial intelligence on cloud computing: Evolution, vision, trends and open challenges,” arXiv, vol. 8, p. 100118, 2019.
[183] U. K.C., S. Garg, J. Hilton, J. Aryal, and N. Forbes-Smith, “Cloud Computing in natural hazard modeling systems: Current research trends and future directions,” Int. J. Disaster Risk Reduct., vol. 38, 2019.
[184] M. K. Yogi, K. Chandrasekhar, and G. V. Kumar, “Mist computing: Principles, trends and future direction,” arXiv, vol. 4, no. 7, pp. 19–21, 2017.
[185] G. S. S. Chalapathi, V. Chamola, A. Vaish, and R. Buyya, “Industrial internet of things (IIoT) applications of edge and fog computing: A review and future directions,” arXiv, pp. 1–15, 2019.
[186] B. Varghese and R. Buyya, “Next generation cloud computing: New trends and research directions,” Futur. Gener. Comput. Syst., vol. 79, pp. 849–861, 2018.
[187] I. Baldini et al., “Serverless computing: Current trends and open problems,” Res. Adv. Cloud Comput., pp. 1–20, 2017.
[188] J. M. Hellerstein et al., “Serverless Computing: One Step Forward, Two Steps Back,” arXiv, vol. 3, 2018.
[189] N. Hassan, K. L. A. Yau, and C. Wu, “Edge computing in 5G: A review,” IEEE Access, vol. 7, pp. 127276–127289, 2019.

[190] R. K. Barik et al., “Mist Data: Leveraging Mist Computing for Secure and Scalable Architecture for Smart and Connected Health,” Procedia Comput. Sci., vol. 125, pp. 647–653, 2018.
[191] K. Qazi and I. Aizenberg, “Towards Quantum Computing Algorithms for Datacenter Workload Predictions,” IEEE Int. Conf. Cloud Comput. CLOUD, vol. 2018-July, pp. 900–903, 2018.
[192] B. Liu, Y. Zhang, G. Zhang, and P. Zheng, “Edge-cloud orchestration driven industrial smart product-service systems solution design based on CPS and IIoT,” Adv. Eng. Informatics, vol. 42, no. September, p. 100984, 2019.
[193] Y. Wu, H.-N. Dai, and H. Wang, “Convergence of Blockchain and Edge Computing for Secure and Scalable IIoT Critical Infrastructures in Industry 4.0,” IEEE Internet Things J., vol. XX, no. X, pp. 1–1, 2020.
[194] Q. V. Pham et al., “A Survey of Multi-Access Edge Computing in 5G and Beyond: Fundamentals, Technology Integration, and State-of-the-Art,” IEEE Access, vol. 8, pp. 116974–117017, 2020.
[195] F. Wang and X. Zhang, “Secure Resource Allocation for Polarization-Based Non-Linear Energy Harvesting Over 5G Cooperative CRNs,” IEEE Wirel. Commun. Lett., vol. 2337, no. c, pp. 1–1, 2020.
[196] A. El Azzaoui, S. K. Singh, Y. Pan, and J. H. Park, “Block5GIntell: Blockchain for AI-Enabled 5G Networks,” IEEE Access, vol. 8, pp. 145918–145935, 2020.
[197] P. Das, S. Ghosh, S. Chatterjee, and S. De, “Energy Harvesting-enabled 5G Advanced Air Pollution Monitoring Device,” 2020 IEEE 3rd 5G World Forum, 5GWF 2020 - Conf. Proc., pp. 218–223, 2020.
[198] H. Gao, S. Zhang, Y. Su, and M. Diao, “Energy Harvesting and Information Transmission Mode Design for Cooperative EH-Abled IoT Applications in beyond 5G Networks,” Wirel. Commun. Mob. Comput., vol. 2020, 2020.
[199] S. K. Sharma, I. Woungang, A. Anpalagan, and S. Chatzinotas, “Toward Tactile Internet in beyond 5G Era: Recent Advances, Current Issues, and Future Directi ons,” IEEE Access, vol. 8, pp. 56948–56991, 2020
[200] Q. Liang, T. S. Durrani, X. Gu, J. Koh, Y. Li, and X. Wang, “IEEE Access Special Section Editorial: New Waveform Design and Air-Interface for Future Heterogeneous Network towards 5G,” IEEE Access, vol. 8, pp. 160549–160557, 2020.
[201] M. Bhatia, S. K. Sood, and S. Kaur, “Quantum-based predictive fog scheduler for IoT applications,” Comput. Ind., vol. 111, pp. 51–67, 2019.
[202] T. M. Fernandez-Carames and P. Fraga-Lamas, “Towards Post-Quantum Blockchain: A Review on Blockchain Cryptography Resistant to Quantum Computing Attacks,” IEEE Access, vol. 8, pp. 21091–21116, 2020.
[203] A. A. A. El-Latif et al., “Providing End-to-End Security Using Quantum Walks in IoT Networks,” IEEE Access, vol. 8, pp. 92687–92696, 2020
[204] K. Shankar, “Improving the Security and Authentication of the Cloud with IoT using Hybrid Optimization Based Quantum Hash Function,” vol. 1, no. 2, pp. 61–71, 2020.
[205] Lindsay, Dominic, Sukhpal Singh Gill, Daria Smirnova, and Peter Garraghan. "The evolution of distributed computing systems: from fundamental to new frontiers." Computing (2021) : 1-20.
[206] H. Chegini, R. K. Naha, A. Mahanti, and P. Thulasiraman, “Process Automation in an IoT–Fog–Cloud Ecosystem: A Survey and Taxonomy,” *IoT*, vol. 2, no. 1, pp. 92–118, Feb. 2021, doi: 10.3390/iot2010006.
[207] F. E. F. Samann, S. R. M. Zeebaree, and S. Askar, “IoT Provisioning QoS based on Cloud and Fog Computing”, *JASTT*, vol. 2, no. 01, pp. 29 - 40, Mar. 2021.
[208] D. C. Nguyen, P. N. Pathirana, M. Ding and A. Seneviratne, "Integration of Blockchain and Cloud of Things: Architecture, Applications and Challenges," IEEE Communications Surveys & Tutorials, vol. 22, no. 4, pp. 2521-2549, Fourthquarter 2020, doi: 10.1109/COMST.2020.3020092.
[209] A. Toth, "Cloud of Things Security Challenges and Solutions," 2021 Communication and Information Technologies (KIT), 2021, pp. 1-6, doi: 10.1109/KIT52904.2021.9583760.
[210] A. A. A. Ari et al., “Enabling privacy and security in Cloud of Things: Architecture, applications, security & privacy challenges,” Appl. Comput. Informatics, no. xxxx, 2019, doi: 10.1016/j.aci.2019.11.005.
[211] D. C. G. Valadares, N. C. Will, M. A. Spohn, D. F. de S. Santos, A. Perkusich, and K. C. Gorgônio, “Confidential computing in cloud/fog-based Internet of Things scenarios,” Internet of Things (Netherlands), vol. 19, no. March, p. 100543, 2022, doi: 10.1016/j.iot.2022.100543.
[212] S. S. Gill *et al.*, “AI for next generation computing: Emerging trends and future directions,” *Internet of Things (Netherlands)*, vol. 19, no. March, p. 100514, 2022, doi: 10.1016/j.iot.2022.100514.
[213] S. Roy, J. Li, and Y. Bai, “A Two-layer Fog-Cloud Intrusion Detection Model for IoT Networks,” Internet of Things (Netherlands), vol. 19, no. May, p. 100557, 2022, doi: 10.1016/j.iot.2022.100557.
[214] M. A. Aleisa, A. Abuhussein, F. S. Alsubaei, and F. T. Sheldon, “Novel Security Models for IoT–Fog–Cloud Architectures in a Real-World Environment,” Appl. Sci., vol. 12, no. 10, 2022, doi: 10.3390/app12104837.
[215] Y. Yang, L. Zhang, Y. Zhao, K. K. R. Choo, and Y. Zhang, “Privacy-Preserving Aggregation-Authentication Scheme for Safety Warning System in Fog-Cloud Based VANET,” IEEE Trans. Inf. Forensics Secur., vol. 17, pp. 317–331, 2022, doi: 10.1109/TIFS.2022.3140657.
[216] Y. I. Alzoubi, A. Al-Ahmad, and H. Kahtan, “Blockchain technology as a Fog computing security and privacy solution: An overview,” Comput. Commun., vol. 182, pp. 129–152, 2022, doi: https://doi.org/10.1016/j.comcom.2021.11.005.