AI-empowered Fog/Edge Resource Management for IoT Applications: A Comprehensive Review, Research Challenges and Future Perspectives

Guneet Kaur Walia , Mohit Kumar, *Member, IEEE* and Sukhpal Singh Gill

Abstract— **The proliferation of ubiquitous Internet of Things (IoT) sensors and smart devices in several domains embracing healthcare, Industry 4.0, transportation and agriculture are giving rise to a prodigious amount of data requiring ever-increasing computations and services from cloud to the edge of the network. Fog/Edge computing is a promising and distributed computing paradigm that has drawn extensive attention from both industry and academia. The infrastructural efficiency of these computing paradigms necessitates adaptive resource management mechanisms for offloading decisions and efficient scheduling. Resource Management (RM) is a non-trivial issue whose complexity is the result of heterogeneous resources, incoming transactional workload, edge node discovery, and Quality of Service (QoS) parameters at the same time, which makes the efficacy of resources even more challenging. Hence, the researchers have adopted Artificial Intelligence (AI)-based techniques to resolve the above-mentioned issues. This paper offers a comprehensive review of resource management issues and challenges in Fog/Edge paradigm by categorizing them into provisioning of computing resources, task offloading, resource scheduling, service placement, and load balancing. In addition, existing AI and non-AI based state-of-the-art solutions have been discussed, along with their QoS metrics, datasets analysed, limitations and challenges. The survey provides mathematical formulation corresponding to each categorized resource management issue. Our work sheds light on promising research directions on cutting-edge technologies such as Serverless computing, 5G, Industrial IoT (IIoT), blockchain, digital twins, quantum computing, and Software-Defined Networking (SDN), which can be integrated with the existing frameworks of fog/edge-of-things paradigms to improve business intelligence and analytics amongst IoT-based applications.**

Index Terms— **Edge Computing, Resource Management, Fog Computing, Artificial Intelligence, Machine Learning, Cloud computing, IoT.**

Guneet Kaur Walia is with Department of Information Technology, Dr. B.R. Ambedkar National Institute of Technology, Jalandhar, Punjab, India. (e-mail: guneetkw.it.22@nitj.ac.in).

Mohit Kumar is with Department of Information Technology, Dr. B.R. Ambedkar National Institute of Technology, Jalandhar, Punjab, India. (e-mail: kumarmohit@nitj.ac.in).

Sukhpal Singh Gill is with School of Electronic Engineering and Computer Science, Queen Mary University of London, London, UK. (e-mail: s.s.gill@qmul.ac.uk)

I. INTRODUCTION

Over the past three decades, applications characterized by varying workloads and substantial datasets have been the driving force behind transformative developments in distributed computing. This computing paradigm gained prominence due to its capability to cater to both compute and data-intensive tasks, driven by its inherent characteristics such as fault tolerance, resource sharing, load balancing, robustness, scalability etc. However, there are challenges, including comprising data movement overhead, synchronization and the complexity involved in handling data distribution and communication amongst nodes, which makes it unsuitable for high-performance scientific and engineering applications. After that, High-Performance Computing (HPC) has been introduced to address the mentioned issues, and it plays a vital role in power systems by optimizing grid control, cost minimization, reducing losses and transmission investment planning [1]. This paradigm provides high-quality solutions within reasonable time, primarily due to its ability to deliver high computational performance. Market-driven advanced computing systems have strategically shifted from HPC to High Throughput Computing (HTC). This transition aims to enhance not only processing speed but also to address critical issues like cost efficiency, energy savings, system reliability, and security [2]. Nonetheless, the escalating trend of data sharing across networks makes it imperative to employ programming models for executing programs across multiple distributed infrastructures. In order to manage the same, leveraging Virtual Machines (VMs) and virtualization has proven to be crucial in effectively handling the growing data quantities within distributed infrastructure. Advances in virtualization have paved the way for the emergence of Internet clouds as a novel paradigm. Significantly, the advancement of technologies like Radio Frequency Identification (RFIDs), Global Positioning System, and various sensors has catalyzed the emergence of the Internet of Things (IoT).

A. Unfolding Emerging Computing Paradigm

With digitization revolutionizing the world at an expeditious rate, IoT is emerging as a broad and multifaceted term encompassing several components and protocols, leading to a dominant technological shift. This rapid evolution is being utilized in domains such as smart factories, structural healthcare, smart cities, smart transportation, supply chain control, intelligent shopping applications, smart agriculture and many more in the form of autonomous IoT applications [3]. This paradigm shift has empowered self-driving drones to carry out home deliveries of groceries, enabled healthcare experts to conduct continuous health monitoring through wearable sensors, facilitated real-time monitoring of equipment and processes in supply chain management, and many more. The predictions regarding the potential influence of IoT are indeed remarkable.

The term "IoT" is defined as a network of physical objects which comprises embedded technologies to communicate, sense and perceive data from their external environment. The IoT applications mainly consist of three components: things also known as devices, insights and actions. The devices are embedded with sensors or actuators. The sensors detect changes in the ambient conditions or in the state of the system and forward this information to the designated destination. In other words, this paradigm aims at building a smart environment by employing smart devices that autonomously generate data and transmit it via the Internet to facilitate decision-making [4]. Finally, Business Intelligence (BI) is utilized to draw appropriate actions from the insights. However, this paradigm imposes a few challenges such as: (1) Security; (2) Energy-efficiency; (3) Data storage and analytics (4) Resource-constrained. To illustrate these challenges, consider smart devices such as RFID tags used for asset identification and tracking in the Industry 4.0 scenario [5]. Smart manufacturing aims at predicting future conditions of manufacturing to ameliorate asset management and quality control of manufactured equipment. For example, NIROTech is a manufacturing company specializing in access and safety control devices for homes. Such devices include surveillance cameras, biometric and facial recognition door locks, fire alarms etc. These devices become susceptible to attacks by malicious users as smart home services are provided over a wireless network. Hence, it becomes important to preserve the integrity (prevent the insertion of malignant software applications into the IoT network, which might change the service purpose), availability (prevention against injecting fabricated data which has the ability to overload device causing financial losses) and authentication (which implies safeguarding the service environment from Denial of Service (DoS) Attack, Distributed DoS, and personnel information leakage etc.) aspects of a safe home [6]. The security aspect of Autonomous Vehicles (AVs) in Intelligent Transportation Systems (ITS) is considered, which interact with other vehicles and Road Infrastructure Units (RSUs) using telecommunication technology. AVs are now being embraced due to their driverless nature, fuel efficiency (AVs are capable of travelling at high speed because of Intelligence and quick sensors), adaptive behavior, remote monitoring and control, optimal path planning etc. The connected AVs contain sensors such as Light Detection and Ranging (LiDAR), Inertial Measurement Unit (IMU), Radio Detection and Ranging (Radar), GPS, cameras, thermal imaging etc., along with connection mechanisms (cellular connections, Wi-Fi, Bluetooth etc.). These components enable AVs to navigate efficiently in an environment by identifying obstacles. The information shared from vehicle sensors to their peer vehicles, or RSUs is vulnerable to being exploited by illegitimate users, raising concerns about data security and privacy in connected ITS. These unauthorized users have the potential to access an AV through various entry points, including USB connections, Bluetooth technology, in-car navigation systems, and other monitoring components. Hence, it becomes challenging to manage such a huge amount of data from potential attacks. Generally, such users conduct two types of attacks: spoofing and jamming [7]. Spoofing attacks include radar spoofing and GPS spoofing, in which counterfeit data is fed to AVs, aiming to gain significant control over the system's behavior. It might drag the AV in the wrong direction. Sometimes attackers send blocking signals which prevent AVs from receiving authentic information from their counterparts, which constitutes jamming. Therefore, it becomes important for developers and manufacturers to develop strategies for mitigating the dangers associated with such attacks to reduce such cyberattacks.

To understand the energy perspective of IoT devices, consider the scenario of Industrial IoT (IIoT), which comprises devices such as Computational RFID (CRFID) tags, ZigBee or LoRa-based sensors. These devices generate a tremendous amount of data and signals, which are used for controlling, sensing, predictive maintenance, and data analysis [8]. However, the communication and computation tasks of these "things" consume a substantial amount of energy, leading to a carbon footprint. For example, smart grids incorporate a large number of sensors that autonomously report their information to grid infrastructure. The sensing and communication tasks consume a lot of energy, that ultimately impacts the lifetime of smart grids [9]. In addition, executing task on optimal destination becomes significant as it incurs a different amount of computation and communication costs [10]. Another challenge is that IoT devices are generally energy-constrained since they are powered by batteries. To increase the lifetime of IoT devices, battery replacements and recharging don't serve as an optimal solution due to increased cost, and in some situations, the location of IoT devices might be inaccessible [11]. Last but not least, the substantial mobility exhibited by dynamic autonomous vehicles leads to increased fluctuations in the Received Signal Strength Indicator (RSSI) at the base station for wireless connections, which results in increased energy dissipation [12]. Therefore, addressing energy efficiency concerns becomes essential for achieving long-term sustainability in real-life deployments of IoT use cases.

IoT devices generate both structured (numbers and values) and unstructured (text, video and audio files) types of data in massive amounts. As per an estimation by Cisco, 500 billion devices will fall under the expansive IoT paradigm, accumulating as much as 79.4 ZB of data by the year 2030 [13]. This necessitates finding suitable solutions for data storage and processing since the physical resources of these devices are limited in terms of power, memory, and computational resources [14]. However, IoT devices execute one task at a time, and in some scenarios, the resource requirements of incoming applications cannot be merely served by IoT paradigm. For instance, meeting user demands while viewing video content of emergency patients in vehicles (such as ambulances) is challenging considering the resource-constrained nature of IoT devices. Therefore, considering all these challenges, it becomes evident that a single computing paradigm is not sufficient to address all the needs of consumers or IoT devices. It further necessitates the incorporation of appropriate resource management strategies via Artificial Intelligence (AI) leveraging the computation capabilities of other emerging paradigms to make it worthwhile in real-world scenarios.

The processing and analysis of compute-intensive IoT devices' data requires high-end storage, network, and computational capabilities, which can be accomplished by utilizing resource-rich cloud infrastructure [15]. Cloud computing is possibly the most impoverished paradigm, which evolves from the capability to harness utility computing, enabling pay-as-you-go models, parallel computation, load balancing, and the data-intensive nature of tasks [16]. Its ability to endorse a Service-Oriented Architecture (SOA) empowers it to create, incorporate, and generate new services by seamlessly integrating with existing ones. This results in the provisioning of services at three distinct levels: Platform-as-a-Service (PaaS), Infrastructure-as-a-Service (IaaS), and Software-as-a-Service (SaaS) [17], leading towards a terminology called Everything-as-a-service (XaaS) [18]. A new computing paradigm has emerged, referred to as the Cloud of Things (CoT), which transforms ubiquitous computing and allows the utilization of cloud architecture in the processing and analysis of extensive IoT data [19], [20]. The combination of both technologies will provide robust, seamless, and agile services for Next-Generation Networks (NGNs) [21].

In CoT, dynamic provisioning of underlying resources and the creation of VMs on demand are the key solutions for managing physical machine resources [22]. Nonetheless, it results in engaging the limited resources of the host machine. CoT envisions device management, including brokering messages between devices and the cloud. Message Queuing Telemetry Transport (MQTT) works as a broker that enables machine-to-machine (M2M) communication by enabling publish/subscribe services [23]. Data generated by devices is published, and data in the cloud-IoT will subscribe to it. Containers, on the other hand, are gaining prominence in multi-cloud platforms to manage and orchestrate applications into portable containers, especially in PaaS models [24]. Within no time, CoT has started harnessing Containers-as-a-Service (CaaS) to provide services in the fields of transportation and Next-Generation Sequencing (NGS) bioinformatics [25]. Furthermore, it motivates the computation and processing of big data residing in cloud environments by efficiently processing spark jobs and hence, improving workload makespan as compared to traditional virtualization [26]. An integral approach involving the synergy of both technologies is going to be embraced by future cloud architectures.

Despite the predominance of the CoT paradigm, it faces numerous challenges in hosting real-time IoT applications. Processing IoT requests at the cloud layer results in high latency due to bandwidth constraints, making it inadequate to cater the demands of real-time applications. This issue has been resolved by fog computing, which brings the computational capability of its underlying resources within close proximity of the end-user. It leverages cloud infrastructure in a decentralized manner, placing storage, computing, and processing components at the edge of the network [27]. Instead of being a replacement for the cloud, this computing paradigm introduced by Cisco acts as a complement to the existing framework.

Fog/Edge computing emphasizes processing data in close proximity to the service-consumer, bringing new advantages such as greater context-awareness amongst nodes, real-time data processing, lower bandwidth consumption, and so on [28]. This computing paradigm makes the cloud truly distributed. However, IoT devices are battery-driven and resource-constrained, which elicits the need

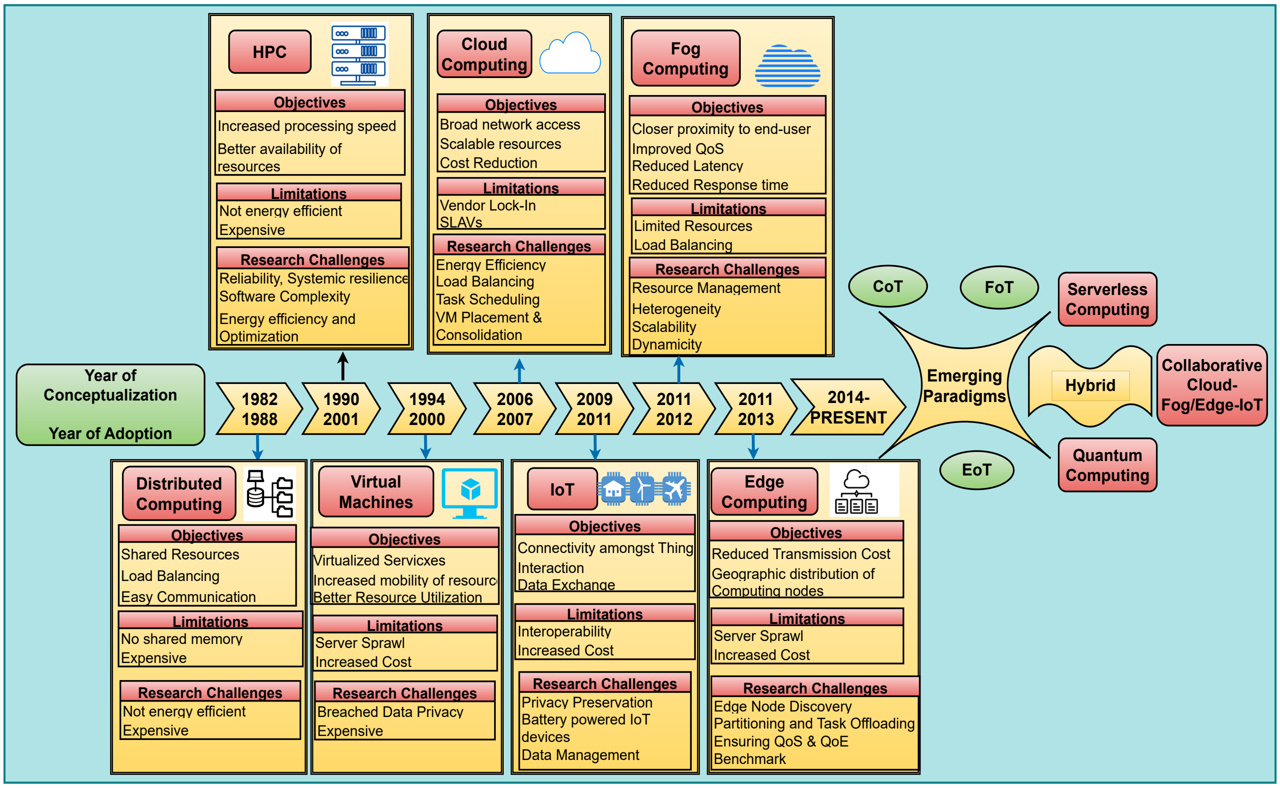

Fig. 1. A Timeline of the Evolution of Computing Paradigms

to incorporate other emerging paradigms such as fog and/or edge, serverless, and quantum computing to enrich their resource capabilities and make them capable of incorporating business logic, which is responsible for running lightweight computation. Resource management in the emerging computing paradigm for real-time applications is a critical issue due to the incoming transactional workload (of a complex and diverse nature), scalability across data centers, and last but not least, fluctuating interactions and managing Service Level Agreements (SLAs) along with QoS parameters [29]. In contrast to the cloud, resource allocation in fog computing is more complicated because workloads are distributed among fog nodes due to its de-centralized architecture. Furthermore, determining the suitable destination node for executing incoming IoT workloads is also a critical consideration. For example, the latency-sensitive data collected is analyzed at the edge of the network, whereas if the application thrives for high computation service and storage, then the data is offloaded to the cloud for processing and analysis. Hence, the problem of resource management in such a collaborative environment thrives for leveraging compute, storage, networking, and intelligence capabilities from resource-enriched cloud and fog layers. This eventually calls for integrating the intelligence tier, which runs machine learning and deep learning models at the cloud layer while inferencing is carried out at the intermediate fog layer. This approach has now been envisioned as a potential platform for ameliorating services in smart cities, smart industries, connected vehicles, UAVs, Wireless Sensors, and Actuators Networks (WSANs), further boosting the development of advanced applications. Figure 1 depicts the evolution of various prominent computing paradigms with a timeline along with their objectives, limitations, and research challenges. The subsequent pointers highlight various emerging paradigms such as Fog of Things (FoT), Edge of Things (EoT) and most importantly, the hybrid computing paradigm, which provides a collaborative framework for various layers to work in a harmonized manner.

1) Fog of Things

Fog computing exploits its virtualization characteristics in order to deliver network, storage, and computing resources to end-users at the network edge. It offers services to latency-sensitive IoT applications with tolerable delay. Fog computing, also known as fogging, is an architectural framework that utilizes edge resources usually, but not entirely situated at the edge of the underlying network [27]. These nodes are designed to carry out a substantial number of local computing and storage tasks while also managing data routing over the network. Delay-sensitive applications, for instance, Mobile Augmented Reality (MAR), thrive on up-to-date data processing and computational requirements, which cannot be fulfilled with a cloud scenario [30]. Hence, to ensure seamless transmission of real-time gaming data, the fog paradigm provides a competent solution. Furthermore, offloading IoT data over fog optimizes Quality of Service (QoS) parameters such as latency, energy consumption, reliability, throughput, and bandwidth utilization.

Furthermore, if the edge device sends latency-sensitive data to the cloud for analysis and waits for appropriate action, then it might result in unwanted delays due to its geographically distant nature as comparison with the fog/edge node. To overcome the mentioned limitations, fog computing provides limited processing, computation capability and storage services closer to the end user. Appropriate placement of services under IoT applications for execution in a fog or cloud environment is challenging, which often leads to inappropriate distribution of workload amongst VMs. To ensure optimal QoS and Quality-of-Experience (QoE), efficient distribution and tuning of IoT application workloads amongst Fog Nodes (FN), is complicated because of the distributed and heterogeneous architecture of resources in FNs [31].

2) *Edge of Things*

As data is increasingly generated at the network edge, the most efficient processing of the data can be done on the edge device itself. The downstream flow of data is triggered by cloud datacenters, whereas the upstream data flow is managed on behalf of IoT devices [32]. In addition, moving massive amounts of potentially useless raw data to the cloud is expensive to transmit and store, and can even disable a network [33]. It introduces a heavy load on network transmission bandwidth, leading to data latency and backhaul, which makes it unfit for real-time applications, especially in the automotive industry. To exemplify, edge computing plays a vital role in the domain of vehicular networks by enabling a smooth exchange of data amongst vehicles and coordinating uniform flow to enrich user proficiency [34]. The rationale of edge computing lies in the fact that computation will happen in proximity to the data source. Edge computing is an unprecedented term that planted its seed in enhancing the speech recognition process in mobile devices under finite resources by computational offloading to a proximate server [35].

In the edge paradigm, content is pushed out geographically so that the data is more readily available to network clients with low latency. The compute capacity amongst edge servers and devices is introduced in distinctive environments, including warehouses, retail stores, banks, etc., where communication is enabled via the 5G network. The edge nodes prompt the 5G network by ensuring latency (one millisecond or less). Despite improved services in comparison to CoT, this paradigm witnesses a few drawbacks, like seamless migration of workloads, limited storage, heat dissipation, battery life and the limited computational power of mobile devices in an IoT environment. To deal with it, a cyber-foraging framework has been proposed [36]. This technique empowers communication by offloading the data to powerful datacenters residing in the cloud or to edge nodes lying in proximity.

3) *Fog vs Edge*

Edge and fog are frequently used interchangeably by most researchers; however, it's important to note that edge processes the task at the device itself whereas fog processes the request on near-end devices such as smart routers, gateways and network switches. The decision to incorporate a fog layer within a particular SOA rests entirely with the service provider, which is influenced by factors like application type, network architecture, data characteristics, and the location of essential network tools and resources. Although both fog and edge work towards leveraging storage and computational capabilities closer to the end user instead of pushing them to cloud datacenters, they still differ from one another in the following context: (1) *Where does data processing take place?* It happens at the network edge or device itself from the point of data generation in edge computing, but in the case of fog computing, the processed data is relocated to processors connected to a Local Area Network (LAN) relatively farther from sensors, gateways, and actuators. (2) *possession of processing and storage capabilities?* Fog nodes are comparatively more powerful than edge nodes. (3) *On which layer do they work?* Where edge computing emphasizes edge devices, fog computing sways at the infrastructure level.

4) *Hybrid Computing Paradigm*

Modern research is trending in the direction of exploiting the collaborative layered architecture of cloud-fog/edge computing in order to cater to delay-sensitive and compute-intensive tasks, as depicted in Figure 2. Some prominent application areas include AR/VR gaming, 24x7 video surveillance, maritime engineering, electronic health and activity tracking, autonomous vehicle management and Wireless Sensors and Actuator [37] [38]. This collaborative framework fits well for such applications where each layer performs distinctive functions, as explained below:

- *Collaborative Cloud-Fog/Edge-IoT:* It constitutes the following layers:

Perception Layer: The IoT ecosystem encompasses a wide range of components, such as sensors, and mobile IoT devices like smartphones, smartwatches and consumer electronics, as well as household appliances like refrigerators, televisions, microwaves, and ovens. These devices gather and transmit data from their operational surroundings. This data generation spawns some tasks that require a timely response. For instance, these tasks can be categorized into hard and soft deadline-based tasks

This article has been accepted for publication in IEEE Communications Surveys & Tutorials. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/COMST.2023.3338015

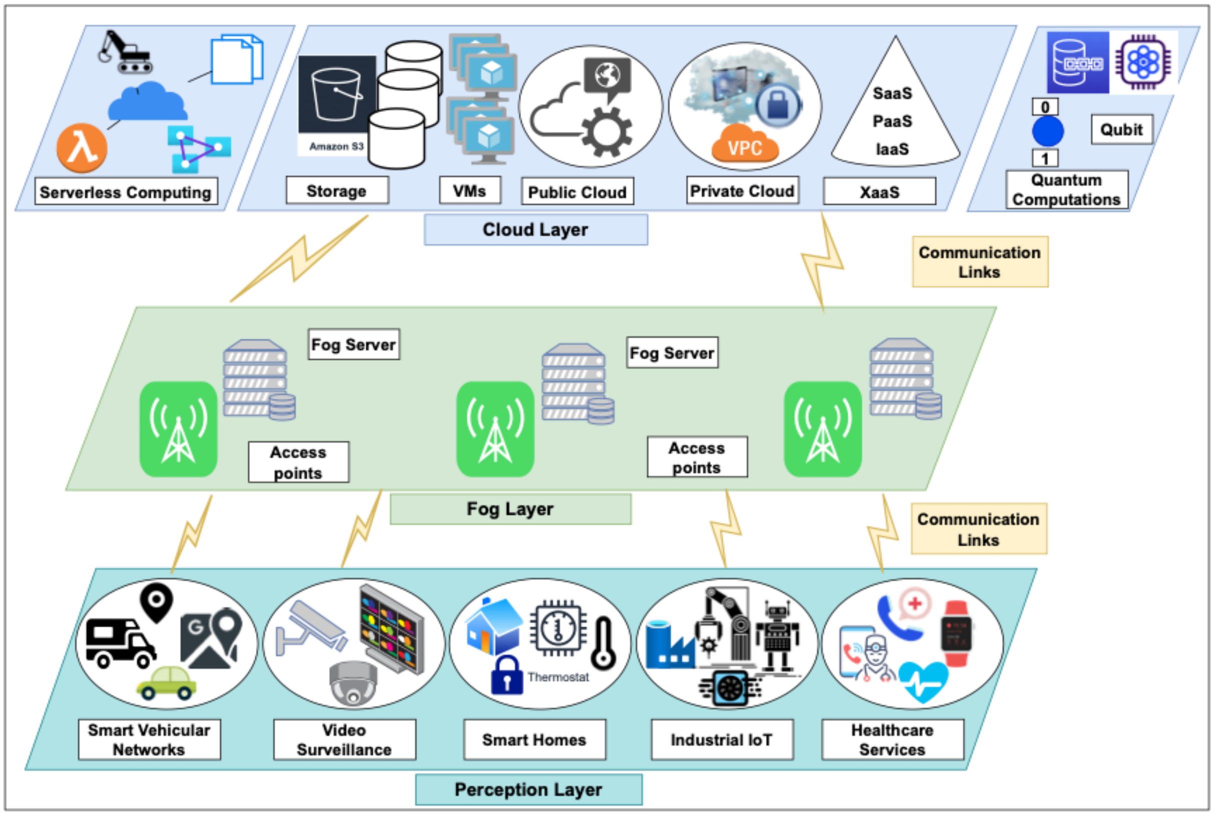

Fig. 2. Hybrid Computing Paradigm

Fog Layer: This layer is characterized by features such as location awareness, low latency, support for mobility, heterogeneity and interoperability. Apart from providing a decentralized and ubiquitous form of computing, this layer acts as a platform to harmonize coordination amongst heterogeneous fog nodes and the programmability of networking resources, to name a few [40]. The fog nodes utilize the capabilities of cellular base stations, network routers, Wi-Fi gateways, etc., to perform operations such as managing and analyzing data and other time-sensitive actions in close proximity to the device user. In addition to enhancing the service response time, its capability to process data saves network bandwidth by reducing the need to upload it to the cloud every time.

Cloud Layer: This layer enables ubiquitous, on-demand access to a shared pool of configurable computing resources that can be provisioned and scaled as per application needs. Despite this, IoT applications require genuine response services, which cannot be fulfilled by the cloud. However, its limitless computational capabilities make it an irreplaceable choice for catering to compute-sensitive applications such as social networking, video conferencing, and so forth. Therefore, we consider this framework to explore various issues relating to resource management in the resource-constrained fog layer and the computationally equipped cloud layer.

Serverless Computing: The cloud layer is considered the optimal destination for hosting serverless computing due to its predominant characteristics which include centralized architecture, elasticity and scalability etc. However, fog nodes can communicate with serverless functions deployed on the cloud, thus acting as event sources or consumers [41].

Quantum Computing (QC): It is a fascinating and powerful technology that promises to revolutionize a considerable range of intellectual and economic factors in our society. In contrast to classical computing, this paradigm is based on quantum mechanics, which uses Qubits that can be in superpositions of both states at the same time [42]. Although the cloud layer is considered best for QC, quantum computations and insights can be leveraged by fog nodes and IoT devices. The in-depth incorporation of QC is depicted in Section IV.

B. Google Trends

The Google trends depict the pattern of emerging computing paradigms over the past six years (2018 to the present), which is visually presented in Figure 3. It has been observed that Fog, Edge, and AI/Machine Learning are in vogue for integration with the cloud computing paradigm. Our study investigates recent research trends based on the state-of-the-art for newly fangled thrust technology, including blockchain, 5G, quantum computing,

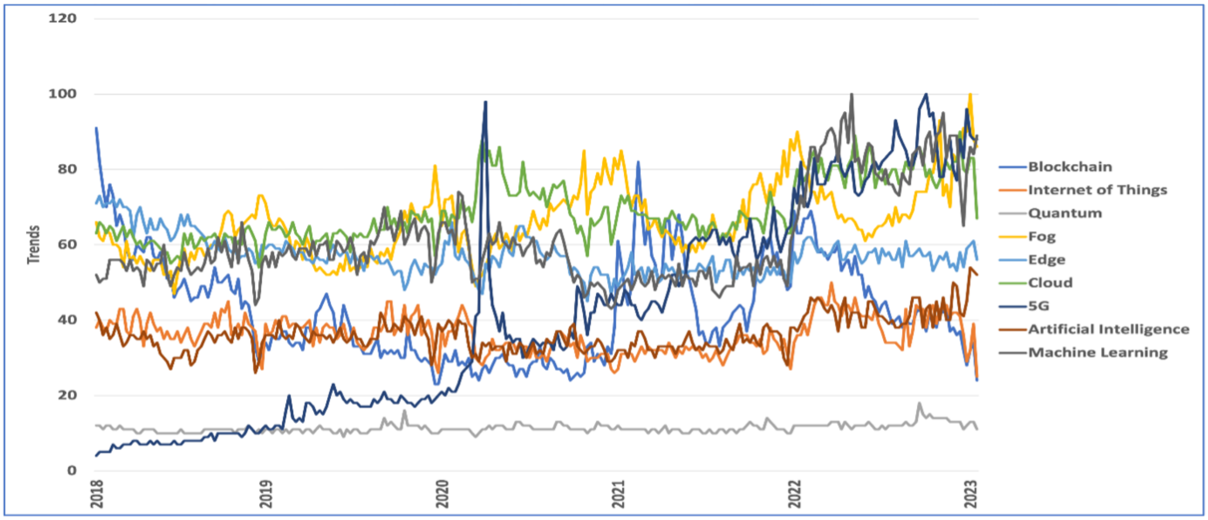

Fig. 3. Google Trends of Emerging Paradigms during last 6 years

and many more. It is observed that the usage and practical applicability of the term quantum computing are still at an infancy stage. Only a few works have presented its usage, which identifies it as one of the future potential research directions. However, terminology such as blockchain, 5G, and the IoT is taking the lead as compared to other emerging technologies. Also, the escalating trend amongst the cutting-edge technologies of fog and edge showcases the integration of these technologies in the era of modern computing paradigms.

C. Motivation for conducting the Survey

With the advancements in the sphere of technology, new frontiers harnessing cloud computing have come up, encompassing fog and edge computing, whose intent resides in extending the cloud's compute, storage, and ubiquitous characteristics nearer to IoT devices or mobile users. The existing literature lacks a clearly defined vision and concept of the emerging computing paradigm, resulting in a restricted comprehension of the role of AI in this domain. AI includes techniques such as metaheuristics, Machine Learning (ML), Deep Learning (DL), and Reinforcement Learning (RL) under its umbrella, which can automate and reshape the existing traditional resource management capabilities to reach new heights.

For instance, the decision to map and manage the execution of tasks on fog nodes is challenging as multiple incoming tasks might compete for the same resources, and the time slot desired by one IoT application task might be occupied by another application. Hence, to resolve such issues, AI provides an adaptive system that is programmed intelligently to handle the dynamic workload requirements of incoming applications. Keeping all these factors in mind, this paper explores the architectural framework of emerging paradigms and provides insights into incorporating AI components with computing paradigms such as Fog-of-Things and Edge-of-Things. This review tries to familiarize its readers with the various state-of-the-art integrating technologies (AI-employed and non-AI-employed). This includes an exploration of research objectives, advantages, disadvantages, and a comparative analysis of QoS metrics.

Despite the significance of utilizing AI techniques in fog/edge computing, we have found few surveys that emphasize the efforts made in the management of resources. One of the considerable studies has been done by Ghobaei-Arani et al. [43]. This survey presents resource management problems in taxonomical form, which is divided into six categories. It includes application placement at the appropriate destination node, task scheduling, provisioning of resources, allocation, load balancing, and task offloading. In-depth studies have been done in all categorized areas, along with discussions of open issues. However, this survey doesn't provide research trends for integrating thrust technologies like serverless computing, blockchain, quantum computing and Software-Defined Networking (SDN). Abdulkareem et al. [44], have studied the role of ML approaches in addressing the problems of resource management pertaining to fog computing. The paper highlights the application areas, challenges, and open issues covering the security aspect. Nevertheless, the work discussed ignores the classification of resource management techniques. Other survey parameters like taxonomy and QoS-based comparison are not mentioned. Also, a comprehensive study done by Hong et al. [45] presents an architectural classification for effective management of resources in Fog/Edge. It discusses future research perspectives to address various challenges. However, they have ignored some topics such as resource provisioning, resource scheduling, allocation, etc. QoS parameters regarding algorithmic categorization of resource management have also been ignored.

TABLE I
COMPARISON OF OUR SURVEY WITH EXISTING SURVEYS BASED UPON INTEGRATION WITH THRUST TECHNOLOGY

Year & Reference	**Emerging Computing Paradigm**	**AI**	**Taxonomy**	**RM**	**QoS based parameters Comparison**	**Integration with Thrust Technology**								**Open issues and Future Challenges**
						Serverless	**5G**	**IIoT**	**Blockchain**	**Digital Twins**	**QC**	**FL**	**SDN**	
2019 [43]	Fog	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✓
2019[44]	Fog	✓	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓
2019 [45]	Fog/Edge	✗	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗
2020 [51]	Cloud-Fog-Edge	✓	✓	✗	✗	✗	✗	✓	✗	✓	✗	✓	✗	✓
2020 [52]	Cloud-Fog/Edge	✓	✗	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✓
2020 [47]	Fog	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓
2020 [46]	Edge	✓	✗	✓	✓	✗	✗	✗	✗	✗	✗	✓	✗	✓
2021 [48]	Fog	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✓
2021 [49]	Fog/Edge	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✓
2021 [56]	Edge	✓	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✓
2022 [42]	Cloud Fog-Edge	✓	✗	✓	✗	✓	✓	✗	✗	✗	✓	✗	✗	✓
2022 [53]	Cloud-Fog/Edge	✓	✓	✓	✗	✗	✗	✗	✗	✓	✗	✗	✗	✓
2022 [54]	Edge	✓	✗	✗	✗	✗	✗	✓	✗	✗	✗	✓	✗	✓
2022 [50]	Fog-IoE-Cloud	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✓
2023 [57]	Fog/Edge	✓	✓	✗	✓	✗	✗	✗	✗	✗	✗	✓	✗	✓
Our Survey (this work)	Cloud-Fog-Edge	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Another review by Deng et al. [46] addresses the convergence of edge computing with AI. The work presents two-faced objectives: using AI for edge computing and on the other side, using AI on edge computing environments. However, it only discusses the concept of task offloading along with mobility management using AI techniques. Reviewing the resource provisioning and allocation schemes in lieu of studying their efficacy to reinforce both static and dynamic IoT applications in fog computing has been done by Martinez et al. [47]. They presented four distinct phases for the implementation of fog infrastructure and highlighted some current challenges and future directions. Although it provides an in-depth study of the necessary steps vital for the practical implementation of infrastructure, it ignores important resource management concepts such as scheduling underlying resources, offloading IoT-based tasks and load balancing. Furthermore, Nayeri et al. [48] have surveyed the role of AI-based solutions for placing application workload in fog computing in the form of a taxonomy of AI algorithms (ML, evolutionary and combinational). However, they have studied only service placement techniques without categorizing them, and other important aspects of resource management such as resource provisioning, resource allocation, service placement and load balancing have not been scrutinized. Also, no discussion has been done regarding new-fangled technologies like SDN, serverless computing, quantum computing, blockchain etc. A study by Shakarami et al. [49] presents a systematic review by proposing a classification for resource provisioning. However, the work has been divided into five classes based on framework, heuristic or meta-heuristic, model-based, ML and game theory. Still, other important prospects for resource management have not been discussed. In another study done by Bushra et al. [50], QoS-based comparisons are evaluated in fog and IoE environments. The author describes the role of heuristic and metaheuristic algorithms in resource allocation and task scheduling. Nevertheless, it has not considered all aspects of resource management.

Considering some recent research, Zhang J et al. [51] endow the concept of Artificial Intelligence of Things (AIoT) in a cloud-fog-edge environment. However, the importance of resource management in resource-constrained IoT applications is not discussed. Another study by Lin et al. [52] highlights the issue of Resource Management (RM) in wireless communication networks. But it doesn't discuss the implications of the proposed work as real-life case studies relating to self-driving cars, smart logistics, the role of Collaborative Robots (COBOTS) in Industry 5.0 etc. AI-augmented technology integration and its impact are discussed in the Fog/Edge paradigm in context with resource management. But some important aspects of resource management, such as task offloading, which leverages IoT applications by incorporating orchestrators into edge devices, are ignored [53]. The work done by Gill S et al. [42] provides the concept of quantum computation in lieu of managing large amounts of data and describes how quantum computing solutions will take over the existing world's economy. But nevertheless, other cutting-edge technologies such as blockchain, Digital Twins, Federated Learning (FL), and Industrial IoT (IIoT) have not been discussed. In another study by Su et al. [54] the author emphasizes training and inference at the edge, utilizing FL to train models. The study remains confined to training the models via FL, ignoring possible solutions for edge intelligence such as security, consumer privacy, scheduling, etc., which can be accomplished by incorporating thrust technology. Another study brings forth a practical AIoT approach in a real-life scenario, discussing training and inferencing at the edge [55].

After extensively surveying all the related state-of-the-art works, the author realizes that none of the works have considered resource management in depth in the form of taxonomical bifurcation, its related challenges, or its proposed solutions. Therefore, to the best of the author's knowledge, this is the first comprehensive review in this domain, encompassing all facets of resource management. It delves into suggested integrations, their complexities, upcoming trends, pros and cons, and underscores their incorporation with cutting-edge technologies. Hence, our work outlines the mentioned incorporation in addition to highlighting an in-depth survey of research challenges and recognizing various issues in resource management. Our study emphasizes the integration of thrust technologies like Serverless computing, 5G, SDN, Blockchain, QC, FL, Digital twins, the IIoT as a prospect for future research directions which the author believes is of paramount importance. This survey will make an impactful impression on readers by articulating needs and recommending solutions to various problems, including the social and ethical impacts of IoT-enabled computing paradigms in a real-world scenario. We also present a formulation of existing resource management problems in mathematical form, which aims to equip the researchers by articulating needs and further devising solutions to this problem. Table I summarizes the comparison study of thrust technology that can be integrated with the Fog/Edge-of-Things paradigm.

D. Our Contributions

The main contributions to this work are as mentioned below:

- Presents a comprehensive overview of state-of-the-art driving dynamic resource management, encompassing controlling over and under-provisioning, offloading IoT-based tasks, scheduling, and service placement, including allocation of resources and load balancing.
- This work classifies and analyses existing AI-based solutions, emphasizing machine learning, metaheuristics, and combinational techniques for fog/edge resource management.
- Mathematically formulate the problem of provisioning adequate resources, offloading task requests at optimal destinations, scheduling the task, service placement and finally balancing workload amongst nodes, considering latency and cost optimization models.
- We have analyzed the existing solution in the form of QoS metrics along with their limitations and considered all aspects of resource management with classification, description, and limitations.
- The article emphasizes fruitful discussions of recently published articles incorporating AI-enriched techniques, which can work as a stepping stone for potential future researchers.
- This survey represents and evaluates the integration of the computing paradigm with thrust technologies such as Serverless computing, FL, IIoT, Digital Twin, Industry 4.0, SDN, Blockchain and Quantum Computing in the form of frameworks and applications.

This article has been accepted for publication in IEEE Communications Surveys & Tutorials. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/COMST.2023.3338015

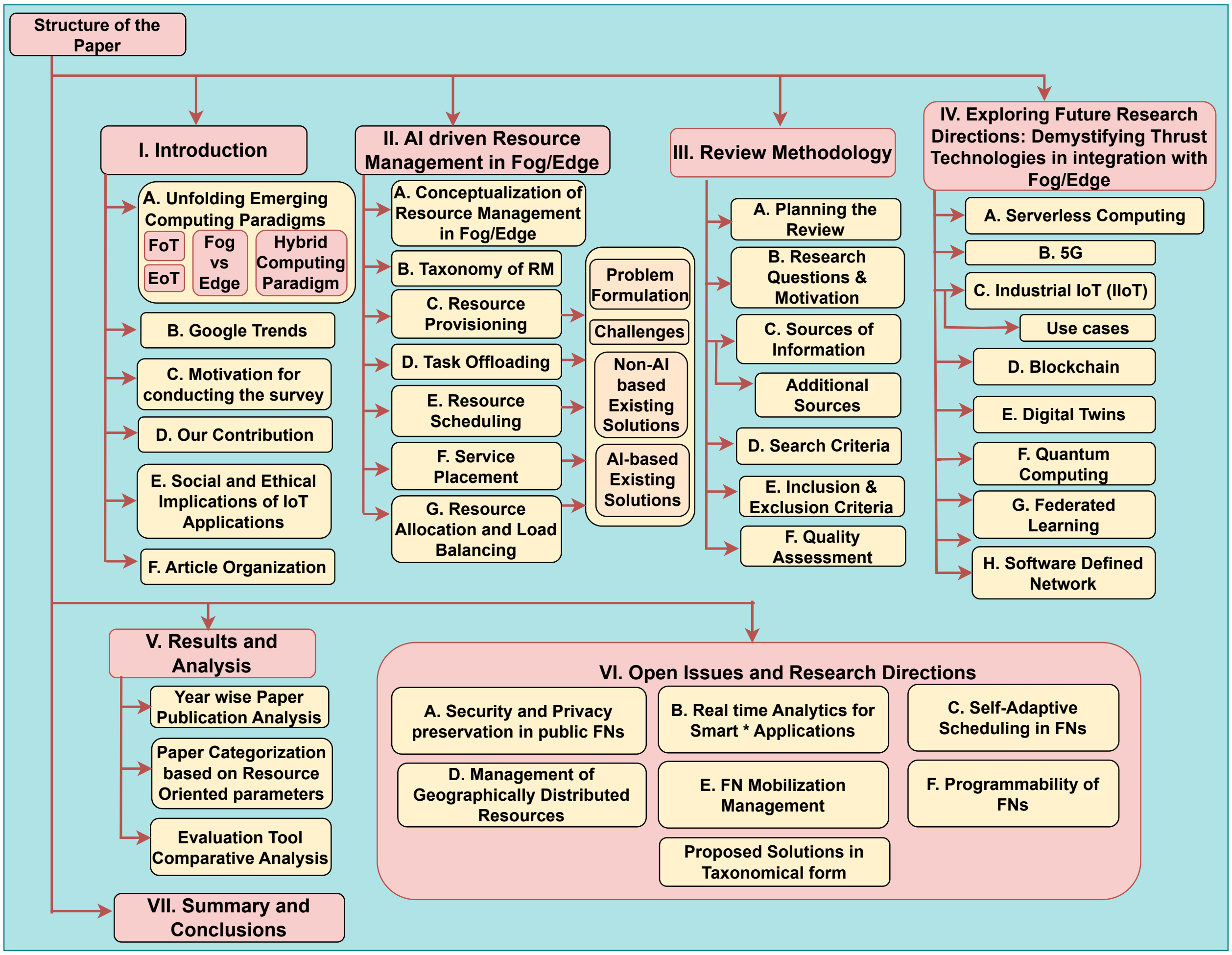
Structure of the Paper
I. Introduction
A. Unfolding Emerging Computing Paradigms
FoT
EoT
Fog vs Edge
Hybrid Computing Paradigm
B. Google Trends
C. Motivation for conducting the survey
D. Our Contribution
E. Social and Ethical Implications of IoT Applications
F. Article Organization
II. AI driven Resource Management in Fog/Edge
A. Conceptualization of Resource Management in Fog/Edge
B. Taxonomy of RM
C. Resource Provisioning
D. Task Offloading
E. Resource Scheduling
F. Service Placement
G. Resource Allocation and Load Balancing
Problem Formulation
Challenges
Non-AI based Existing Solutions
AI-based Existing Solutions
III. Review Methodology
A. Planning the Review
B. Research Questions & Motivation
C. Sources of Information
Additional Sources
D. Search Criteria
E. Inclusion & Exclusion Criteria
F. Quality Assessment
IV. Exploring Future Research Directions: Demystifying Thrust Technologies in integration with Fog/Edge
A. Serverless Computing
B. 5G
C. Industrial IoT (IIoT)
Use cases
D. Blockchain
E. Digital Twins
F. Quantum Computing
G. Federated Learning
H. Software Defined Network
V. Results and Analysis
Year wise Paper Publication Analysis
Paper Categorization based on Resource Oriented parameters
Evaluation Tool Comparative Analysis
VI. Open Issues and Research Directions
A. Security and Privacy preservation in public FNs
B. Real time Analytics for Smart * Applications
C. Self-Adaptive Scheduling in FNs
D. Management of Geographically Distributed Resources
E. FN Mobilization Management
F. Programmability of FNs
Proposed Solutions in Taxonomical form
VII. Summary and Conclusions

Fig. 4. The organization of this Systematic Literature Review (SLR)

- Endows solutions based upon thrust technology corresponding to futuristic challenges in resource management in collaborative Cloud-Fog-IoT paradigm.

E. Social and Ethical Implications of IoT Applications

The escalating use of sensors and corresponding enhancement of smart environments have driven the world to integrate sensors within the system, eventually resulting in data being exchanged between environments, humans, and various objects [58]. IoT has brought about a transformative wave in the world of technology, promising unparalleled convenience and connectivity in our daily lives. As IoT applications continue to proliferate, they extend their influence far beyond the realm of technology, touching upon the very fabric of our society, ethics, and the ways we interact with the world. This exploration delves into the multifaceted landscape of the social and ethical implications arising from the widespread adoption of IoT applications. This implies having a significant impact on citizens, society, and government organizations by imparting round-the-clock medical assistance by eminent doctors and healthcare experts, improving quality of life by cutting the carbon footprint, enabling access to education in remote, unsupervised areas, and the list goes on and on. The utilization of automation and analytics within IoT devices enhances customer service and refines business management strategies by facilitating the tracking and monitoring of both employees and products [58]. Nonetheless, the adoption of IoT gives rise to certain non-technical consequences, encompassing social and legal risks as well as ethical considerations.

The ethical aspect comprises several factors such as transparency, accountability, sustainability, consumer safety etc. [55]. The transparency principle ensures that decision-making and underlying techniques work in a transparent and understandable manner. Accountability guarantees that the developer or manufacturer bears the responsibility for the consequences and effects of AI-powered IoT applications. To demonstrate, consider the scenario of an autonomous driverless car that accelerates, gains momentum, applies brakes, slows down etc. in response to other vehicles on the road (heavy vehicles, pedestrians, cyclists etc.) and other traffic-related parameters.

TABLE II
LIST OF ACRONYMS USED IN THE ARTICLE

Acronym	Description	Acronym	Description
ACO	Ant Colony Optimization	**MEC**	Mobile Edge Computing
APSO	Accelerated PSO	**MEL**	Manufacturing Edge Layer
ARIMA	Autoregressive Integrated Moving Average	**MMAS**	Max-Min Ant System
BFD	Best Fit Decreasing	**MMPA**	Modified Marine Predators Algorithm
BLOT	Bandit Learning Based Offloading	**MOO**	Multi Objective Offloading
BOA	Butterfly Optimization Algorithm	**MOPSO**	Multi Objective Particle Swarm Optimization
BP	Back Propagation	**MPA**	Marine Predators Algorithm
BPSO	Binary Particle Swarm Optimization	**NANN**	Nonlinear Autoregressive Neural Network
BS	Base Station	**NGN**	Next Generation Networks
CAPEX	Capital Expenditure	**PSO**	Particle Swarm Optimization
CART	Classification and Regression Tree Algorithm	**QC**	Quantum Computing
CCR	Communication to Computation Ratio	**QoE**	Quality of Experience
CEA	Cultural Evolution Algorithm	**QoS**	Quality of Service
CoT	Cloud of Things	**RA**	Resource Allocation
DAG	Directed Acyclic Graphs	**RFID**	Radio Frequency Identification
DRL	Deep Reinforcement Learning	**RL**	Reinforcement Learning
DVFS	Dynamic Voltage Frequency Scaling	**RM**	Resource Management
EDF	Earliest Deadline First	**RP**	Resource Provisioning
ESC	Edge Server Cluster	**RSU**	Road Side Units
FN	Fog Node	**SDN**	Software Defined Network
GWO	Grey Wolf Optimizer	**SLA**	Service Level Agreement
HCF	High Computing Fog	**SLAV**	Service Level Agreement Violations
HSOM	Hierarchical SOM	**SLO**	Service Level Objective
IIoT	Industrial Internet of Things	**SOA**	Service Oriented Architecture
IoE	Internet of Everything	**SOM**	Self-Organizing Maps
IoT	Internet of Things	**VFC**	Vehicular Fog Computing
IWO	Invasive Weed optimization	**VM**	Virtual Machine
LCF	Low Computing Fog	**WOA**	Whale Optimization Algorithm
MDC	Micro Data Center	**WRT**	Weighted Response Time
MDP	Markov Decision Process	**WSANs**	Wireless Sensors, and Actuators Networks

Despite the functionalities of IoT-integrated paradigms, any malfunction with the driverless vehicle might potentially harm the safety of other vehicles on the road [59]. Henceforth, it becomes ethically foremost to mitigate such harms in the working environment of IoT, protecting the physical space [60].

Hence, accountability fosters trust between service provider and consumer, ensuring that the service provider will be held liable for their actions. In a similar manner, the safety principle assures consumer data safety from potential breaches and malicious attacks. To exemplify, recent advancements delve into the realm of robot ethics in the context of COBOTS in Industry 5.0 and autonomous vehicles. This aspect underscores the importance of upholding the safety aspects of physical spaces, ensuring that sensors and actuators operate appropriately to accurately interpret environmental conditions, thereby mitigating potential accidents [61]. These parameters collectively contribute towards an ethical, IoT-driven smart world. Nevertheless, the legal and ethical implications of this technology in the practical world encourage the government to enforce meticulous strategies, ensuring safety standards are met by IoT-driven devices. These devices should undergo periodic safety assessments to detect potential hazards and system failures.

F. Article Organization

The paper is structured as follows: *Section II* conceptualizes the concept of resource management in depth and presents the significance of incorporating AI-empowered techniques in the Fog/Edge of Things paradigm via state-of-the-art advantages and open issues. Also, a few non-AI approaches in the existing literature have been reviewed. *Section III* presents the review methodology. *Section IV* provides integration of emerging computing paradigms with thrust technology as a future research scope. *Section V* depicts an analytical study of selected articles in the form of a graphical representation, based upon categorization and comparative evaluation. *Section VI* includes the research gaps and challenges identified. Finally, we summarize the work in the form of a conclusion in *Section VII*. The detailed organization of this comprehensive review is illustrated in Figure 4. Table II lists various acronyms used in this comprehensive review.

II. AI-DRIVEN RESOURCE MANAGEMENT IN FOG/EDGE

The stochastic nature of the fog environment is influenced by multiple factors, including the rate of job arrivals, delay-focused data, interdependencies among incoming requests, dynamicity, status of resources (busy or idle), the number of tasks within IoT applications, varying resource requirements of real-time applications, and the accessibility of computational resources. Thus, conventional or heuristic methods do not work well in a dynamic fog computing environment due to their inability to adapt to constant changes. Moreover, the implications of fog computing in physical world applications necessitate fault-tolerant and adaptive resource management mechanisms which can be achieved through efficient and optimized task or workload scheduling. Hence, to acquire efficiency at the infrastructural level, optimal management of resources can't be compromised. Presently, research is trending in the direction of inculcating AI with computing paradigms, hence making the system autonomous.

AI is increasingly ingrained in our daily lives, contributing to informed decision-making through the utilization of meta-heuristics, ML and DL approaches. It is being utilized by recommendation systems for companies such as Facebook, Instagram, Amazon, and Google and in use cases such as healthcare [62], earthquake prediction [63], Industry 4.0, etc., which require the efficient handling of gigantic amounts of data generated from sensors. This data needs to be efficiently analyzed in order to extract certain features for accurate training of AI models. The process of training a fully equipped model can be complex, especially in context to the time required for training a machine learning-based model. In contrast, deep learning offers a key advantage over classical machine learning by delivering superior performance, especially when dealing with extensive datasets. Since many IoT applications generate vast amounts of data, deep learning methods are particularly well-suited for such systems [64]. For example, Deep Reinforcement Learning (DRL) methods are gaining insight in various forms, including convolutional Deep Neural Networks (DNN), deep belief networks, Recurrent Neural Networks (RNN) etc., for enhancing the computational intelligence of systems. It also provides a solution for predicting extensive workloads and aids the system even where ML techniques fail. Nevertheless, metaheuristic solutions provide promising results for scheduling tasks to appropriate nodes in the distributed architecture of Fog/Edge of Things [65]. Hence, our subsequent section highlights an exhaustive study implicating non-AI and AI-based (ML, DL, metaheuristics, and hybrid methods) for dynamic resource management.

A. Conceptualization of Resource management in Fog/Edge computing paradigm

Resource Management (RM) is a major challenge in the emerging computing paradigm that includes device heterogeneity, resource-constrained nature, large-scale geographical distribution, edge node discovery, dynamic workload, unpredictable demand, and diversity [66] [67]. In contrast to the cloud, allocating resources for upcoming requests is quite tough in the case of fog computing, where the load has to be distributed amongst fog nodes due to its decentralized architecture [68]. In addition to this, it is the most preferred platform for performing complex computing for scientific workflows, where even one small wrong decision in resource allocation can lead to substantial monetary loss [69]. Moreover, the increasing number and complexity of IoT applications make the process even more challenging. Hence, in order to orchestrate IoT applications, it becomes important to optimally provision the resources without compromising the application's performance and user satisfaction level.

B. Taxonomy of Resource Management

The trundle of fog RM begins with provisioning resources, scheduling, service placement, allocating, and load balancing [43]. It becomes noteworthy to define the term "resource" as a collection of hardware (processor, storage, network, power, memory and communication media) and software components (VMs, instances, and containers) [70]. The RM module comprises components that are responsible for resource allocation among incoming tasks, followed by scheduling. The task offloading mechanism decides where to offload the IoT requests for performance enhancement by determining the algorithm and trade-offs to be taken into consideration [71]. The trade-off comprises fault monitoring and incentives to be paid in accordance with the device type. The process of Resource Allocation (RA) reserves and defines resources for a particular end user, whereas the resource provisioning module provisions and deprovisions the resources as and when required by the service consumer. Resource provisioning comprises the effective allocation of IaaS resources among applications running over distributed platforms. Such applications require orchestration and fencing, which traditional methods can't support. It intends to minimize SLA violations by implying efficient server allocation strategies amid interactive and real-time jobs.

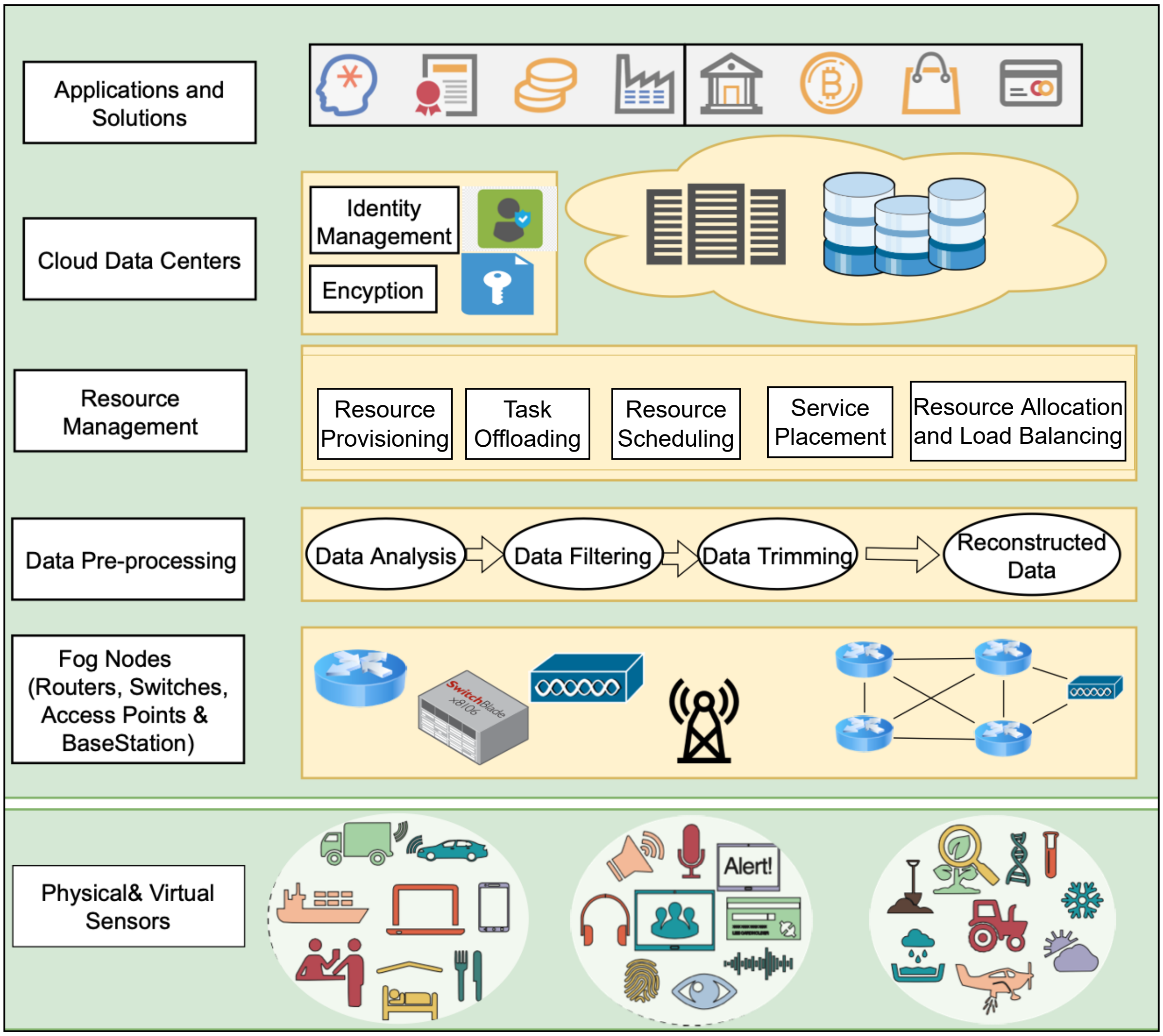

Fig. 5. Resource Management in Fog Computing Environments

The resource allocation module is also responsible for managing the allocation, re-allocation, and de-allocation of all the dependent and independent tasks. This process is followed by the scheduling component, which ensures energy efficiency, optimal resource usage, and minimize the operational costs of task execution. All these components reside in the complex architecture of the fog computing environment, as depicted in Figure 5. The bottommost layer, comprising clusters of physical and virtual sensors residing in intelligent homes, devices, CCTVs, smart cities, grids, automated driving vehicles and so on, is networked to fog devices. The monitoring level keeps track of system performance, resource utilization, utility, and feedback [72]. Data pre-processing and reduction represent promising concepts that enhance the efficiency of IoT data processing and analysis. Implementing data reduction at the edge layer can effectively decrease network bandwidth and latency at the gateway, thereby mitigating I/O bottlenecks in the broader network connection [73]. For instance, consider the case of managing IoT-based industrial data in smart manufacturing. This IIoT data caters to prognosis and predictive maintenance tasks, which calls for unnecessary data to be trimmed due to limited transmission, computational, storage and processing capabilities. The trimming procedure guarantees the exclusion of faulty, incomplete, and redundant data [74]. Finally, the provision of services to real-time applications must maintain a high level of security through the incorporation of encryption. In addition, it is imperative to acknowledge that while encryption plays a pivotal role in safeguarding data confidentiality, it is not a comprehensive solution for addressing all security considerations. Encryption primarily focuses on data confidentiality, ensuring that unauthorized access to data is prevented [75]. However, several other crucial aspects of security warrant attention. Firstly, in applications with real-time constraints, relying solely on encryption can introduce unacceptable latency, compromising the immediate responsiveness required in critical systems. Secondly, the concept of data integrity is central to security, ensuring that data remains unaltered during transmission or storage [76]. Encryption, in isolation, does not address the challenge of data tampering, making additional integrity checks essential, especially in the context of thwarting attacks involving false data injection. Authentication is another vital component, as it establishes trust between communicating entities. Encryption, on its own, does not verify the identities of participants, potentially leaving room for attackers to impersonate legitimate entities. Certain applications necessitate non-repudiation, where senders cannot deny their actions. Encryption, while valuable, does not inherently provide non-repudiation features. Furthermore, specific security threats, such as false data injection attacks, fall beyond the scope of encryption. These attacks require supplementary security measures like data validation and intrusion detection to detect and prevent potential harm [77], [78]. Lastly, effective key management is critical for encryption's success. Inadequate key management can compromise the entire security infrastructure. Thus, a holistic security approach combines encryption with these additional measures to ensure comprehensive protection against a wide array of security challenges.

In order to effectively manage resource-constrained fog layer devices, the author has proposed this problem in the form of a taxonomical representation, as depicted in Figure 6, by articulating the needs and research challenges and recommending possible AI and non-AI-based solutions. The subsequent content presents a brief overview of existing non-AI and AI-based solutions in context to resource management in a Fog/edge-enabled IoT scenario.

- *Existing non-AI-based Solutions:* These solutions have been categorized as static approaches, heuristics, mathematical models (Euclidean formulation), mathematical optimization-based methods such as Integer Linear Programming (ILP), Mixed Linear Programming (MLP), etc. The static algorithms are characterized by pre-information schedule creation for the incoming jobs, such as time required to complete the job, resources required etc. [79]. In other words, a general resource schedule is generated beforehand, which, however, leads to resource waste if reserved instances are not utilized in that particular time period. It includes First Come First Serve (FCFS), in which the task that arrives first, gets the resources first [60] [61]. Another static approach is Min-Min, which works by determining the Minimum Completion Time (MCT) for each job, and then, based on the MCT value, the right resources are allocated to the respective jobs. On the contrary, the Max-Min approach selects the job with the maximum execution time and accordingly allocates resources.

However, in a real-world scenario, tasks can appear at runtime, and resources can be dynamically introduced or removed at runtime. To work effectively in such an environment, resource management strategies often rely on optimization methods, which can be divided into two categories: exact and heuristic [81]. Although exact techniques such as Branch and Bound find optimal solutions, however they are not well-suited for extensive problems, particularly those associated with IoT applications. Hence, heuristic algorithms perform comparatively better at finding a near-optimal solution in minimal time. Mathematical optimization methods such as Integer programming are categorized as ILP in cases where both the optimization problem and constraints are linear, whereas when continuous decision variables are introduced, the problem is characterized as MLP.

- *Existing AI-based Solutions:* In order to resolve the limitations of traditional IoT systems, such as enabling real-time response, poor Internet connectivity and data gravity which evolves a better way to find insights than shipping all the data to the cloud. AI heralds' momentum in IoT-enabled smart applications, which is further boosted with the emergence of 5G with the aim of addressing such issues. Recent research trends are gravitating towards incorporating machine learning, DL, DRL, and other hybrid techniques at the network edge for IoT applications. ML-based techniques utilized in the context of optimizing resource utilization include k-means clustering [82], Decision Tree Regression (DTR) [83], Multiple Linear Regression [84], Naïve Bayes [85]. However, the latest works are approaching RL-based techniques due to the need to dynamically adapt and fix parameters changing in IoT-based scenarios [86]. The main benefit of this technique lies in the fact that RL-based techniques such as Q-Learning and State-Action-Reward-State-Action (SARSA) do not require any dataset for training the model. Furthermore, such techniques operate iteratively with the goal of maximizing rewards for the agent, which takes actions based on the environmental state. The drawbacks such as the inability to handle high-dimensional state information regarding incoming IoT tasks, can be resolved by utilizing the competence of Deep learning in RL. Such techniques include DRL, Deep Q-Learning (DQN) and DNN. DRL works well with larger data sets by predicting appropriate action from action space [87]. On the other hand, DQN utilizes deep neural nets typically Convolutional Neural Networks (CNN) for calculating the Q-values.

This article has been accepted for publication in IEEE Communications Surveys & Tutorials. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/COMST.2023.3338015

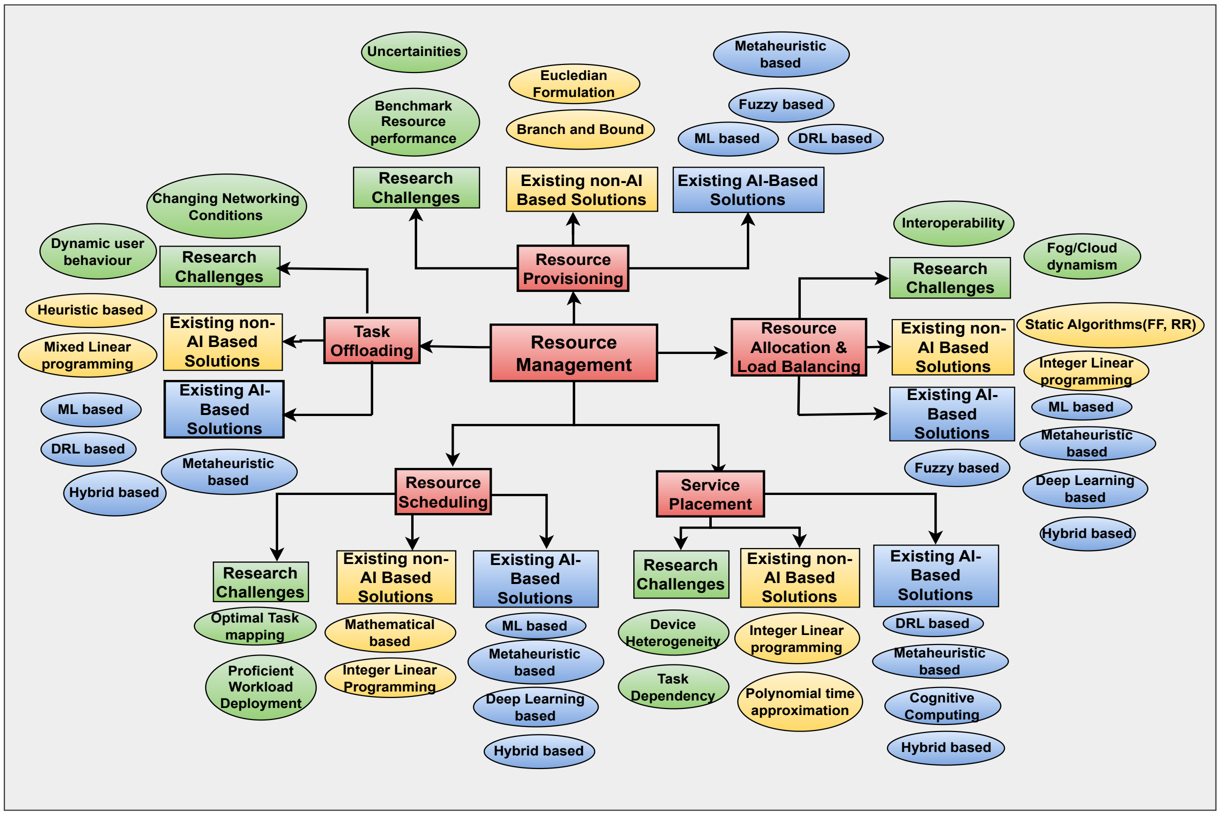

Fig. 6. Taxonomy of Resource Management in Fog/Edge Computing

C. Resource Provisioning

In order to facilitate energy-efficient and low-latency solutions for real-time IoT applications, a thorough analysis of the forecasting network, storage, and computational resource needs for these applications is essential. These applications are accompanied by fluctuating workloads, so static allocation methodologies won't simply serve the purpose. In straightforward terminology, Resource Provisioning (RP) is described as a process that controls resource allocation and de-allocation in a manner such that QoS parameters should not be compromised even in a fluctuating incoming workload environment. This stipulates the need for an automated mechanism that can effectively manage the over-provisioned and under-provisioned system state issues in the fog ecosystem. In under-provisioning, the number of allocated resources is less in comparison to the actual task execution needs, which further leads to SLA violations. Therefore, the basic idea behind RP resides in the detection and selection of judicious resources for the user in accordance with incoming application requests [88]. Alongside, it maps the incoming requests to the VMs to deliver the services with the minimum cost and time.

1) *Problem Formulation for Resource Provisioning*

The collaborative cloud-fog-IoT scenario constitutes collection of n IoT devices $\mathfrak{D}=\{d_1, d_2 \dots d_d \dots d_n\}$ which spawn m tasks $\mathfrak{I}=\{\mathfrak{I}^1, \mathfrak{I}^2, \dots, \mathfrak{I}^j, \mathfrak{I}^m\}$. Set of x fog nodes which serve as micro data centers is represented as $\mathcal{FN} = \{\mathcal{FN}^1, \mathcal{FN}^2 \dots \mathcal{FN}^x\}$. In addition, the author considers the cloud data center ($\mathcal{DS}$) for offering highly computational-oriented services. The objective of resource provisioning is to search for optimal resources for end users within defined constraints. This module of RM accounts for the validation of various constraints, which acts as a preliminary before actual resource allocation and task service take place. The problem of resource provisioning is formulated as follows [89]:

IoT device constraint: $1 \leq d \leq n$ (1)

Task count constraint: $1 \leq \mathfrak{I} \leq m$ (2)

FN constraint: $1 \leq \mathcal{FN} \leq x$ (3)

Deadline constraint: It specifies that each task must not surpass its maximum latency, which corresponds to the task's deadline, $\mathbb{D}^{d,j}$.

$$\sum_{j=1}^{m} \mathcal{T}_{Compute}^{d,j} \leq \mathbb{D}^{d,j} \quad (4)$$

Here, $\mathcal{T}_{Compute}^{d,j}$ denotes the compute time of task.

Bandwidth constraint: This constraint ensures the avoidance of network congestion while performing various resource-centric tasks. It further states that the allocated bandwidth of an IoT device, fog node or cloud data center must not exceed the maximum bandwidth.

$$\sum_{j=1}^{m} \mathfrak{B}_j \leq \mathfrak{B}_{Max} \quad (5)$$

Computational constraint: The required CPU cycles to accomplish the task must not be greater than the available processing capacity of the resource (R). R can be an IoT device, fog node or cloud data center. It can be illustrated as follows:

$$\sum_{j=1}^{m} \Im_{req}^{j} \leq \text{Ꞧ}_{\max\_cpu} \qquad (6)$$

Where $\Im_{req}^{j}$ depicts about the required cpu cycle and $\text{Ꞧ}_{\max\_cpu}$ represents about the max cpu capacity of resource.

Non-preemption: It states that once a task is allocated to a computational node, that task must be completed first before starting the execution of another task at the same resource.

2) *Resource Provisioning Challenges*

The following are the main challenges of resource provisioning:

- **Benchmark Resource Performance:** It comprises some critical aspects such as resource utilization, predictability, and fault tolerance. Predictability defines the system's ability to predict its future behavior. Complementing the existing infrastructure with futuristic knowledge enhances the system's ability to predict future behavior. Nevertheless, fault tolerance underlines the system's capability to withstand the failover without compromising service delivery.
- **Uncertainties:** The inherent characteristic of dynamism in fog environment leads to uncertainties. A powerful resource provisioning mechanism is capable of mapping incoming requests arriving in a stochastic manner to an available set of resources. Furthermore, due to the significant computational complexities involved in this decision-making process, it is considered an NP-Hard problem. Hence, to solve such a complex issue, new AI-based approaches are being proposed based on the problem and underlying environment.

The subsequent sub-section discusses the framework for evaluating QoS study for the work, as shown in Table III, which concludes that the most targeted parameters include cost, delay, and utilization. Whereas, few studies have focused on failover ratio, response time, SLA violation etc. Furthermore, most recent state-of-the-art based on non AI-based and AI-based (Metaheuristics, ML, and DL) along with hybrid techniques is presented, which is summarized in Table IV. Concerning Resource Provisioning, non-AI models are based upon Markovian-Decisions, mathematical-based mechanisms, and various types of linear programming models. Alongside a Noteworthy, Table V depicts various datasets that have been used to evaluate the performance of AI approaches in the domain of resource management. In addition, it aims to deliver dataset-related aspects such as accessibility in the form of links, descriptions and types, facilitating researchers in accessing the dataset and gaining valuable information for their future investigations.

3) *Existing Solutions for non AI-based Resource Provisioning*

The reviewed articles address the problem of over- and under-provisioning of resources in fog/edge computing with various heuristic and mathematical-based models. Yao et al. [90] depict the challenge of provisioning multiple VMs for incoming tasks as a multi-objective problem in a heterogeneous IoT-enabled environment. It aims at minimizing the total cost involved in renting VMs while nevertheless maximizing reliability to confront the issue of VM failovers. The objective has been accomplished by using a modified version of the Best Fit Decreasing (BFD) algorithm; where the incoming tasks are arranged in decreasing order of corresponding length and then provisioned to VMs as per weight capacity (price). The simulation results demonstrate the improvement in results in terms of cost and reliability as compared to state-of-the-art algorithms. The authors propose a software-based approach for workload partitioning among computing layers [93]. Partitioning the workloads provides a prime requisite for optimal provisioning of resources, especially bandwidth consumption to link factories with cloud data centers. Moreover, it assists the existing framework in determining the minimum and maximum number of locally situated servers to be integrated with available FNs in order to provide exquisite resources to real-time applications subjected to time and memory constraints.

It becomes noteworthy to mention the problem of predicting and provisioning resources for serving the enormous number of devices in existing cellular networks. For instance, cellular relay networks thrive on strategies enabling ubiquitous coverage and providing a fair share to respective users, enabling optimal resource provisioning [91]. To maintain the QoS parameters, the service providers are working towards expanding their Base Stations (BS). BS consistently allocates a fixed number of resources, while mobile users access services at discrete intervals, often leading to inefficient resource allocation. In the context of vehicular networks and aerial units, a dynamic resource provisioning approach has been suggested in reference [104]. Additionally, a two-stage algorithm has been introduced to optimize the management of RSU workloads. Briefly, the vehicular computing framework manages incoming workload spikes via flying fog units, hence the introduction of RSUs as a computing paradigm. The overloaded RSU can provision the required resources from nearby base stations based on the calculated lease period, utilizing local and global workloads. The work done is capable of deadline and capacity-aware offloading, handling peak loads, and supporting dynamic resources in the form of vehicles or UAVs using various mathematical models. The results illustrate reduced energy consumption, waiting time and computational time in comparison to the state-of-the-art.

This article has been accepted for publication in IEEE Communications Surveys & Tutorials. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/COMST.2023.3338015

TABLE III
COMPARISON OF PERFORMANCE METRICS FOR RESOURCE PROVISIONING IN FOG/EDGE COMPUTING

Year & Reference	**Dataset**	**QoS Parameters**														
		Cost	**Reliability**	**Failure Recovery Ratio**	**Bandwidth**	**Delay Violation**	**Energy Consumption**	**SLAV Ratio**	**CPU Utilization**	**Waiting Time/ Makespan**	**System Efficiency**	**Response Time**	**No. Requests Rejected**	**Latency**	**Throughput**	**Accuracy**
2010 [91]	NA	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗
2016 [92]	Animation rendering Dataset	✗	✗	✗	✗	✗	✗	✗	✓	✓	✗	✗	✗	✗	✓	✓
2019 [90]	NA	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
2019 [93]	NA	✗	✓	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
2020 [94]	MNIST, Fashion MNIST, and LEAF	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓
2020 [95]	New York City Taxi trip	✓	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗
2020 [96]	Microsoft T-Drive Trajectory & Chicago Taxi Trips	✓	✗	✗	✗	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗
2020 [97]	NA	✓	✗	✗	✗	✓	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗
2020 [98]	Taobao App, Cloud Theme click from Tianchi, Clark Net	✓	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗
2020 [99]	NA	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓	✗	✗	✗	✗	✗
2021 [100]	Google cluster Usage Trace	✓	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✓	✗
2021 [101]	NA	✓	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗
2021 [102]	Fast Fourier Transformation & historical temperature records	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✓	✓	✗	✗	✓
2022 [103]	Real Google Traces comprising 25 mil- lion tasks	✗	✗	✗	✗	✓	✗	✗	✓	✗	✗	✗	✗	✓	✗	✓

4) *Existing Solutions for AI-based Resource Provisioning*

Due to the prevalence of AI, we decided to explore the existing work done in this challenging domain using metaheuristics, ML, DL, and hybrid approaches. Hatti et al. [101] implemented distributed provisioning using Multi-objective Particle Swarm Optimization (MOPSO). Multiple jobs have been categorized into different swarms based upon similarity in response time requirements. Further, each swarm is mapped to a single or multiple FNs taking into consideration the resource capacity, response time, and distance. It works by computing the task fitness value corresponding to a particular swarm. PSO is a nature-inspired evolutionary and stochastic technique for the optimization of computationally hard problems. It is inspired by the swarms in nature, which include bird flocking, fish schooling etc., and holds an edge over other optimization techniques as it utilizes a combination of both local and global search [117].

Considering the ML approaches, the author in [95] proposed an IoT service autonomous mechanism possessing self-sustainability in fog nodes using Bayesian learning along with the incorporation of IBM's MAPE-k model for better decisions regarding resource planning. This technique of Bayesian inference is based upon the Bayes theorem, which considers the probability distribution of each quantity, hence providing effective decision-making [118]. The proposed model withstands different workloads with a minimum error rate. The authors highlight the importance of resource provisioning in multimedia applications [92]. The work signifies the role of edge networks in satisfying the resource-hungry nature of modern applications. Sophisticated algorithms based on ML techniques have been discussed, which focus on the predictor module for the completion time of rendering jobs in context with the underlying available fog resources. Furthermore, the prediction accuracy of job completion time is improved using the multi-fold cross-validation method.

TABLE IV

STATE-OF-THE-ART SOLUTIONS FOR RESOURCE PROVISIONING IN FOG/EDGE COMPUTING

Year & Reference	Objective	FN type	Technique Utilized	Evaluation Tool	Limitations
2016 [92]	To propose a resource provisioning mechanism for edge enabled distributed multimedia applications.	Heterogeneous	Completion Time Prediction (CTP) Algorithm based upon ML	Simulator (java)	The issue of duplicate job assignments is not addressed.
2019 [90]	Formulating a multi-objective problem to handle the trade-off between system cost and reliability.	Heterogeneous	Modified Best Fit Decreasing (MBFD) Algorithm	Simulation (MATLAB)	The author has stated the scope for heterogeneous fog nodes; however, in simulation, homogeneous FNs have been considered.
2019 [93]	Reliability-aware partitioning of real-time workloads in smart factories.	Heterogeneous	Heuristic partitioning Lowest-Laxity First (LLF)	Simulation	Factors such as application active time could have been included for better results.
2020 [95]	IoT service automation in a fog computation environment for delay and cost minimization.	Heterogeneous	MAPE-k	Simulation (iFogSim) workload	Low prediction accuracy of the time series model
2020 [96]	To propose a learning-based resource provisioning technique in a three-tier fog ecosystem for energy optimization.	Homogeneous	NAR and Hidden Markov Model	Simulation (iFogSim)	The LSTM model gives better predictions. High computational complexity
2020 [97]	Workload clustering-based RP for minimization of delay and cost.	Homogeneous	BBO, k-means, Bayesian learning	Simulation (CloudSim)	CPU utilization of machines is not taken into consideration.
2020 [98]	Heterogeneity-aware elastic provisioning in a cloud fog environment for balancing workload	Heterogeneous	Neural network	Experimentation using real workloads.	The proposed approach involves high operational costs and complexity.
2020 [99]	Dynamic resource provisioning using flying Fog for vehicular networks to improve efficiency.	Heterogeneous	Euclidean three-space formulation	Simulation (Anylogic)	Utilization has not been evaluated. The proposed work has not been evaluated as a real-life case study.
2021[100]	To implement proactive service placement and IoE provisioning in Mobile Edge Computing (MEC).	Homogeneous	DRL MDP	Real world simulation Google Cluster Usage Trace	The data-hungry nature of DRL overloads the limited resource capacity of edge nodes. EC is not evaluated
2021 [101]	Optimal provisioning of requests to fog nodes for energy efficiency.	Homogeneous	Multiobjective PSO	Simulation (CloudSim Plus)	The proposed approach is not presented as a real-life case study.
2021 [102]	To implement dynamic Resource Provisioning for containerized microservices in order to mitigate SLA violations.	Homogeneous	Machine learning (EN, DTR)	Microdata centre (MDC) testbed	An effective burst traffic handling approach using an integrated cloud computing paradigm with existing fog architecture.
2022 [103]	Admission Control Manager (ACM) for Handling and Classification of Heterogeneous Jobs.	Heterogeneous	Fuzzy logic	Simulation	Practical implementation of proposed study amongst IoT applications is not illustrated.

It uses various algorithms for prediction problems, such as Random Forest (RF), Support Vector Machine (SVM), DL and Gradient Boosting Tree (GBT), and finally R-square scores, to shortlist the best-performing approach. Finally, the proposed predictor algorithm is compared with various state-of-the-art algorithms. The work done by the authors [102] brings forward a predictive autoscaling technique using ML for containerized Microdata Center (MDC) in fog environments. A workload forecasting mechanism has been used to determine the number of containers required to serve incoming IoT workloads without compromising Service-level Objectives (SLOs).

TABLE V
DATASET AVAILABILITY AND DESCRIPTION

RM Categorization	Reference	Dataset Name and Availability	Description	Type
Resource Provisioning	[92]	Animation Rendering Dataset [105]	-It comprises of 33,078 records containing resource utilizations -Shows the number of jobs arrived each hour from a sample day	-Each record contains a rendering job belonging to animation studio. -CPU usage (%), RAM usage (KB), number of frames, number of polygons, size of image (pixels), completion time (sec)
	[94]	MNIST: https://www.tensorflow.org/datasets/catalog/mnist Fashion MNIST: https://www.tensorflow.org/datasets/catalog/fashion_mnist LEAF: https://github.com/TalwalkarLab/leaf/	-Modified National Institute of Standards and Technology (MNIST) is a database of handwritten digits. -Article images divided into test and train examples -LEAF is an open-source benchmark for federated settings.	-Each example is a gray-scale image. -Image and Text datasets
	[95]	New York City Taxi trip: https://databank.illinois.edu/datasets/IDB-9610843	-Real world IoT workload trace - 697,622,444 records of taxi-trip in New York city collected between 2010 and 2014	Drop off, pick-up dates, time, distance, coordinates recorded by taximeter.
	[96]	Microsoft T-Drive Trajectory : https://www.microsoft.com/en-us/research/publication/t-drive-trajectory-data-sample/ Chicago Taxi Trips: https://data.cityofchicago.org/	-GPS trajectories of 10,357 taxis -contains taxi trip information in Chichago	-each record contains pick-up and drop-off dates, times, coordinates and trip duration.
	[98]	Clark Net: http://ita.ee.lbl.gov/html/traces.html. Cloud Theme click from Tianchi: https://tianchi.aliyun.com/. Taobao App: https://tianchi.aliyun.com/dataset/	-Clark Net comprises workload data of web server from Clark Internet Services -Tianchi is cloud theme click dataset -Taobao App is the largest retail in China.	-Clarknet is extracted from Web servers in Washington, America, which includes 33,28,587 Hypertext Transfer Protocol (HTTP) requests during two weeks -data is in the form of clicks
	[100]	Google cluster Usage Trace: https://research.google/pubs/pub43438/	-It contains a trace of workloads running on 8 Google Borg compute clusters	-Each trace describes job submission, scheduling decision and resource usage.
	[103]	Real Google Traces [GoCJ]: https://data.mendeley.com/datasets/b7bp6xhrcd/1	-comprising 25 million tasks of approximately 930 users	-Tasks (id-value pair)
Task Offloading	[106]	Real time dataset from fog cloud environment	6 Synthetic dataset	-Energy consumption -computational time
	[107]	Vehicular traces [108]	-considers a highway environment where each vehicle maintains a constant speed within an RSU	-Vehicle ID, Timestamp, GPS data, Traffic conditions, Sensor data
Resource Scheduling	[109]	CERIT Trace: https://jsspp.org/workload/index.php?page=cerit	-contains mixed workload of jobs	-Description of cloud VMs -Description of grid worker -Number of available CPUs
	[110]	Intel Berkeley research lab (Not mentioned)	-It consists of 2.3 million sensor readings from 54 sensors	Humidity, light, voltage and temperature data
Service Placement	[111]	Synthetic DAGs using parameters as stated in [112]	-IoT applications modelled as DAGs, where nodes depict the tasks and edges represent the data communication amongst dependent tasks.	-testbed created similar to scenario in
	[113]	Google Cluster Trace and Nasa Server Logs (NSL) https://github.com/google/cluster-	-provides real life deployment scenario of services on available servers. -it constitutes source IP having the	-It provides a set of hosts along with resources available and a set of resources having services requirements.

		data/blob/master/ClusterData2011_2.md -NASA dataset - https://ita.ee.lbl.gov/html/contrib/ NASA-HTTP	same subnet mask as that of requesting services. - NASA dataset comprises two months of HTTP logs from a busy WWW Server:	-It consists of list of changing demands corresponding to GCT service.
Resource Allocation and Load Balancing	[114]	SAIVT Multi-Camera Surveillance Database: https://www.gov.uk/guidance/imagery-library-for-intelligent-detection-systems	-It consists of 8 cameras and contains movement of more than 150 people in cafeteria.	-dataset provides the target tracks in the form of videos in MJPEG file format.
	[115]	MOBILE Health (MHEALTH): https://archive.ics.uci.edu/dataset/319/mhealth+dataset	-Used to detect the possibility of heart attack. -This contains vital signs and body movements recorded corresponding to 10 volunteers during several physical activities.	-Multivariate -Time-series data with 23 attributes and 161,280 instances. -constitutes various attribute values along with probability of occurrence of heart attack.
	[116]	PlanetLab workload: https://planetlab.cs.princeton.edu/datasets.html	-it is used for modelling energy usage pattern of fog nodes.	-it contains CPU utilization

The proposed framework has been evaluated under synthetic and realistic workloads. The work presents potential application areas comprising real-time traffic monitoring, smart healthcare, self-driving cars, and the IoE. The task classification is done on the basis of task priority and scheduling class. Sham et al. [103] proposed a fuzzy-based admission control and resource provisioning system that places request analysis parameters such as CPU, memory, storage, job priority and time sensitivity. The presented methodology considers request parameters in the form of crisp values, which are then passed through the Fuzzy Inference System (FIS), and then aggregation is performed to select the best computing node.

Deep learning-based Solutions: In contrast to classical machine learning techniques, RL-based approaches have the ability to perform linear and non-linear approximations. These approaches dynamically adapt to the changes in the environment and possess the capability to learn the system without prior knowledge. These capabilities make this technique well-suited for resource management problems. An AI-enabled resource provisioning architecture has been proposed for IoE services in 6G networks. It uses DRL along with the Markov Decision Process (MDP) for resource provisioning [119]. The MDP framework is employed for addressing challenges that are resolved through RL. In the context of a multi-application scaling solution, the author explores scenarios where each application consists of a collection of services. MDP considers a tuple ($\mathcal{S}$,$\mathcal{A}$,$\mathcal{P}$,$\mathcal{R}$,γ), where $\mathcal{S}$ is the finite set of states, $\mathcal{A}$ represents the finite state of actions, $\mathcal{P}$ depicts the probability of state transition, $\mathcal{R}$ represents the immediate reward and γ is the discount factor. The state space '$\mathcal{S}$' is formed by the dynamic user demands and resource availability for each host at timestamp 't.' The action space '$\mathcal{A}$' is of fixed size, comprises a pair of elements that determine the CPU and memory scaling decisions. The probability transition matrix '$\mathcal{P}$' quantifies the likelihood of transitioning to the next state 's' when a specific action 'a' is taken. Usually, this value is not given in advance, hence model-free RL technique is utilized to predict this value. Finally, the immediate reward or cost function $\mathcal{R}$, aims at selecting the best action which results in the minimum cost. For instance, minimization of application load and optimizing the available resources are a few examples of cost functions.

However, the ultimate objective of an RL agent is to acquire knowledge about the probability distribution governing transitions from a given state to all possible subsequent states and to determine the optimal policy π^*. The following equations outline how to update the Q-value for a specific state-action pair $\mathcal{Q}(s,a)$, by taking into account the immediate reward R and the minimum Q-value. This value is computed for the next state, s′ which is weighted by the discount factor γ. The equation below iteratively improves the agent’s Q-value estimates.

$$\mathcal{Q}(s,a) := \mathcal{Q}(s,a) + \alpha[\mathcal{R} + \gamma \min_{\alpha'} \mathcal{Q}(s',a')] \quad (7)$$

Where $\mathcal{Q}(s,a)$: represents the current assessment of the expected cumulative reward for taking action a
$\min_{\alpha'} \mathcal{Q}(s',a')$: signifies the lowest expected cumulative reward across all feasible actions a′ in the next state s'.
s': final state after taking transition from state s.
a': action chosen by agent in the next state s'.
All these Q-values are stored in a tabular manner, as a result of which the state space grows with the increasing number of containers/ hosts. Therefore, it becomes computationally very expensive to store and further maintain the updated Q-values. Hence, Optimal Q-values can be derived from adjustable

weights (θ). Gradient descent can be employed to adjust weights in the right direction. Consequently, Q becomes close to optimal Q^*which is represented as:

$$Q^*(s,a) = Q(s,a,\theta) \quad (8)$$

But linear approximations suffer from the following drawbacks:

- Inability to capture complex, non-linear relationships between states and optimal actions.
- Approximation errors accumulate over time, which leads to unstable policies.
- Not suitable for continuous state space, where states can be similar.
- May struggle to adapt to the dynamics of the environment.

Linear approximations can help with large state spaces, but suffer from the curse of dimensionality, which increases with an increase in the number of features. Consequently, non-linear approximations such as DNNs are being utilized which empowers agents to employ deep learning for weight updates and customized learning adjustments.

Etemadi et al. [96] work states an automatic and scalable resource provisioning framework for fog architecture that stores the incoming workload parameters (response time, resource requirements) in a shared database. An autonomous learning-based provisioning model has been proposed utilizing Nonlinear AutoRegressive (NAR) neural networks. In another study done by Li et al. [98], a provisioning strategy in a cloud-enabled fog computing scenario has been proposed, considering tenanted and overhead costs. The exquisite decision-making capability regarding whether to add more instances or release confers the data migration strategy. This is implemented using the hybrid Autoregressive Integrated Moving Average (ARIMA) model and a Back Propagation (BP) neural network. ARIMA is a univariate time-series forecasting model used to predict the number of resources needed to support the incoming workload of smart applications [120]. The hosts located at the edge layer are equipped with hybrid ARIMA and BP neural network algorithms. This forecasted workload acts as a basis for determining whether the request will be processed at the edge tier or rented from cloud data centers. A module known as the workload analyzer is supplied with historical workload data, offering workload estimates with a one-time interval forecast. This interval should be long enough in order to provision an appropriate number of VMs in advance [120]. ARIMA blends autoregressive (AR) and moving average (MA) elements in conjunction with differencing to achieve stationarity in the time series. Despite the capabilities of ARIMA for time series forecasting, it may not capture complex, non-linear patterns or sudden changes in workload behavior. Nevertheless, its univariate nature limits its capabilities to analyze and forecast single-time series variables. Henceforth, it is combined with ML-based models such as BP-NN to improve prediction accuracy. During the training phase of BP-NN, the input patterns are processed in two stages. In the first stage, a predicted value corresponding to each input pattern is provided. Any prediction-related error is propagated backward to update the weights of the hidden and output layers. This update process aims to minimize the prediction error that is associated with it [121]. Finally, the effectiveness of the work done has been validated using extensive experimentation on real-world datasets.

The hybrid approaches provide an amalgamation of any two (or even more) of the above-mentioned solution types. The implementation of RP using Bayesian networks and workload clustering using Biogeography Based Optimization (BBO) in integration with K-means clustering has been proposed by Ghobaei-Arani et al. [97]. The author has emphasized the significance of classification and analysis of the workload of incoming user requests, which acts as a prerequisite for effective resource provisioning. The workflow of the proposed work comprises three phases: preprocessing of incoming workload, clustering, and finally provisioning the resources. The pre-processing stage eliminates noise and filters out workloads with incomplete attributes, as well as artificial users like robots that may propagandize and desolate actual purposes. In addition, it also includes making SLAs. A hybrid approach is followed for workload clustering which works in four phases defined as: (1) Random initialization of habitats comprising the population, (2) Using k-means for habitat evaluation, (3) Immigration and emigration rate calculation and (4) Best habitat selection. Lastly, Bayesian learning serves as a classifier for categorizing RP decisions based on clustered workloads. It utilizes a state table containing SLA cost, response time, workload attributes, and resource scaling decisions to derive Bayesian-based rules.

D. Task Offloading

The best possible task offloading strategy is characterized by its capability to choose an optimal offloading decision, subjected to the incoming IoT workload. In cases of ample resource availability, the job is processed; otherwise, the architecture is scaled up and allocations are updated. In case the fog nodes are incompetent to execute the incoming application request, it is then offloaded to the cloud layer. Task offloading from edge devices to nearby FNs not only reduces latency but also significantly reduces energy consumption [122]. Figure 7 demonstrates the complexity of making the optimal offloading decision due to various factors such as offloading fraction, offloading constraints, and offloading objectives. The objectives focus on improving QoS parameters like cost, throughput, resource utilization rate, energy, etc. On the contrary, offloading constraints comprise

This article has been accepted for publication in IEEE Communications Surveys & Tutorials. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/COMST.2023.3338015

bounds on bandwidth utilization, offloading fraction, and other task-specific criteria [123]. In certain scenarios, only a portion of the task is offloaded, requiring the processed outcome to be subsequently transmitted back along the same route or an alternative one, based on the most favorable option.

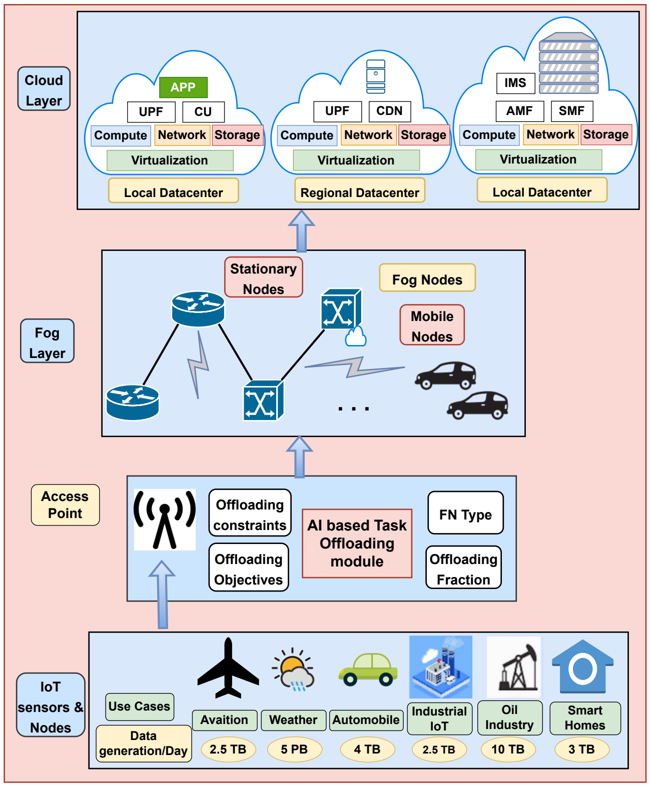

Fig. 7. Task offloading Problem in Fog Computing

1) *Problem Formulation for Task Offloading*

This section highlights the significance of mathematical analysis in managing task offloading, particularly in relation to reducing delays for applications that require instantaneous responses. It aims to jointly minimize task delay/latency and simultaneously economizing energy consumption. Task offloading not only helps to enrich the consumer experience but also accelerates job execution. The authors introduce the offloading latency models to investigate the delay parameter, computational demand and communication demand, corresponding to each incoming request. Moreover, deciding how to offload tasks is intricate due to the diverse wireless networks (including WLAN and MAN) involved, as well as the necessity to choose the most efficient offloading strategy from among multiple edge/fog nodes, and cloud data centers. The problem for offloading tasks at optimal destinations is as follows [10], [124]:

Local Computation at Device: It considers the task processing at local edge device itself, where the latency corresponding to local computational $\mathcal{L}_{IoT}^{d,j}$ is computed as follows:

$$\mathcal{L}_{IoT}^{d,j} = \frac{\mathfrak{I}_{size}^{d,j}}{\psi_{d\_mips}} \tag{9}$$

$\mathfrak{I}_{size}^{d,j}$: Task size corresponding to device d and j^{th} task.
ψ_{d_mips}:computational capacity of IoT device

Offloading to Fog node

The task offloading at local fog delivers ultra-low latency services due to avoidance of network backhaul delay. The latency $\mathcal{L}_{FN}^{d,j}$ can be computed as:

$$\mathcal{L}_{FN}^{d,j} = \mathcal{T}_{FN\_Uptime}^{d,j} + \mathcal{T}_{FN\_Compute}^{d,j} + \mathcal{T}_{FN\_Downtime}^{d,j} \tag{10}$$

Where, $\mathcal{T}_{Offloading}^{k}$ represents the time to offload request to edge node. This time is also known as task uptime, which is represented as $\mathcal{T}_{FN\_Uptime}^{d,j}$

$$\mathcal{T}_{FN\_Uptime}^{d,j} = \frac{\mathfrak{I}_{size}^{d,j}}{\mathfrak{B}_{FN}} \tag{11}$$

Here: $\mathfrak{I}_{size}^{d,j}$: Task size corresponding to device d and j^{th} task
$\mathfrak{B}_{FN}$: Bandwidth of underlying link (WLAN, MAN)

$$\mathcal{T}_{FN\_Compute}^{d,j} = \frac{\mathfrak{I}_{size}^{d,j}}{\psi_{FS\_mips}} \tag{12}$$

Where: ψ_{FS_mips} denotes the computational capacity of Fog server node, and further the authors have considered that the downtime is equivalent to the uptime.

Offloading to cloud data centers

Noteworthy, our work considers full offloading scheme, where task is either on fog or cloud deployed node. Latency in full task offloading case $\mathcal{L}_{DS}^{d,j(Full)}$, which transmits whole task to cloud for processing is computed as :

$$\mathcal{L}_{DS}^{d,j(Full)} = \mathcal{T}_{DS\_Uptime}^{d,j} + \mathcal{T}_{DS\_Compute}^{d,j} + \mathcal{T}_{DS\_Downtime}^{d,j} \tag{13}$$

where, $\mathcal{T}_{DS_Uptime}^{d,j}$represents the uptime of cloud data center, which can be also referred as offloading time.

$$\mathcal{T}_{DS\_Uptime}^{d,j} = \frac{\mathfrak{I}_{size}^{d,j}}{\mathfrak{B}_{DS}} \tag{14}$$

where $\mathfrak{B}_{DS}$: depicts the bandwidth of network (WAN).
$\mathcal{T}_{DS_Compute}^{d,j}$represents the computational time by cloud datacenter node, which can be calculated as:

$$\mathcal{T}_{DS\_Compute}^{d,j} = \frac{\mathfrak{I}_{size}^{d,j}}{\psi_{DS\_mips}} \tag{15}$$

where ψ_{DS_mips}signifies computational capacity of cloud datacenter node.

Noteworthy, the offloading latency model functions in accordance with offloading decision variable $\Theta^{d,j}$ is defined as follows [125]:

$$\Theta^{d,j} = \begin{cases} 0; & task\ computed\ by\ d \\ -1; & task\ offloaded\ to\ \mathcal{FN} \\ 1; & task\ offloaded\ to\ \mathcal{DS} \end{cases} \tag{16}$$

This equation represents the offloading decision corresponding to j^{th} task generated by IoT device. The value of $\Theta^{d,j} = 0$; signifies that the task $\mathfrak{I}^{j}$ is processed at

IoT device itself, whereas the value $\Theta^{d,j}$= -1; indicates that the task is offloaded to local fog server. Finally, the task is being offloaded to central cloud server in case $\Theta^{d,j}$= 1.

2) Task Offloading Challenges

- **Changing Networking Conditions:** The edge/fog computing environment is characterized by dynamic network and traffic conditions. The presence of different noise and interference levels can significantly impact the overall efficiency and latency of wireless transmission. It demands having an analysis and prediction mechanism for underlying networking conditions in order to estimate the right time for task offloading decisions. Such a scenario works in an architecture comprising mobile devices, Mobile Edge Servers (MES) and Fixed Edge Servers (FES), in which the mobile device layer includes various sensors and devices embedded in vehicles, whereas autonomous vehicles serve as MES, and RSUs serve as FES [126]. In this situation, the presence of mobility amongst vehicles and MES, as well as factors such as increased traffic during peak hours and complex road networks, inject a dynamic element into the network conditions. This, in turn, has a direct influence on the decision-making process for optimizing task offloading and route planning for MES operations. To handle this issue, DRL-based techniques are gaining prominence due to their proficiency to self-learn from the environment and incorporate the updated parameters in subsequent iterations [93].
- **Dynamic User Behavior**: The randomized behavior of mobile users adds another level of complexity to task offloading decisions. Currently, research trends are being diverted to machine learning and data analytics techniques in contemplation of prediction and forecasting user behavior [128]. For illustration, consider the urban transportation scenario which comprises latency-sensitive tasks spawned from self-driving vehicles, remote fleet monitoring, etc. Such tasks need to be cautiously handled in order to incorporate appropriate environmental perception, vehicle motion control, opportune decision-making and action, and collaborative Simultaneous Localization And Mapping (SLAM), to name a few. Such versatility among tasks makes the task-offloading decision even more complex. The majority of work done to address this challenge has utilized DRL approaches [126], [129]–[131]. Apart from DRL approaches, some authors have implemented meta-learning to handle dynamic IoT application requirements in the form of incorporating diverse task types [98], [99].
- **Highly Latency-sensitive Requests:** Certain applications cannot endure even milliseconds of delay and require immediate access to computational resources for processing upcoming requests. Such types of requests are either processed by IoT devices or offloaded over the fog/edge nodes. Nowadays, prominent video applications, particularly those involving VR/AR, are becoming increasingly significant in areas like the gaming industry, education, medicine, and more, owing to their immersive visual features [134], [135]. For instance, AR application tasks are effectively handled by utilizing DRL, MES and the capabilities of the 5G network in order to cater to the highly sensitive latency requirements of the task. This is carried out by splitting the incoming task into a Directed Acyclic Graph (DAG) [135].

3) Existing Solutions for non AI-based Task Offloading

This section reviews the state-of-the-art works in detail based on task offloading and optimization objectives. Munoz et al. [136] proposed a framework for the optimization of computational resources in a cellular network. It utilizes the suboptimal approach and the optimal statistical approach. In the former class, the complete dataset is divided into smaller subsets, and an offloading decision is taken corresponding to each subset. In this suboptimal approach, a global decision is taken in conjunction, although in practice it is challenging to predict the channel state in advance. The latter category makes statistical offloading task decisions that adapt based on the real-time updates of channel statistics. Another work by Han et al. [137] proposed an online job dispatching and task scheduling problem in an edge-cloud environment. It assumes that incoming jobs are released in a random order and at irregular time intervals, without any initial statistical information about the jobs. It aims to minimize Weighted Response Time (WRT). As far as the offloading decision is concerned, OnDisc heuristically dispatches the job to cloud server with a minimum WRT. Extensive simulation results utilizing real-world Google Cluster signify better WRT results. Also, incorporating a fairness knob ensures uniform WRT amongst all jobs. Table VI consolidates QoS comparison study of surveyed work. Whereas Table VII summarizes the non AI-based and AI-based solutions to task offloading problem in collaborative cloud-fog-IoT paradigm.

4) Existing Solutions for AI-based Task Offloading

The author emphasizes that the process of choosing the most suitable offloading destination must prioritize the judicious utilization of resources along with meeting the QoS requirements of the incoming request. It proposes a Classification and Regression Tree Algorithm (CART)-based solution for module placement [138]. The proposed approach begins by evaluating the power usage of all mobile devices and, if the power consumption surpasses that of Wi-Fi, it offloads the incoming request to the fog node. The traditional decision trees have limitations when dealing with input spaces. To address this, the CART method is employed, as it accommodates real-valued parameters and helps identify the best feature conditions. After executing Module Placement by Classification and regression tree Algorithm (MPCA), Poisson process is used to calculate the module arrival rate. The Poisson process aligns with Markov chain properties, providing systematic manageability. In addition, the Markov chain is memoryless, indicating that its probability distribution relies solely on the present state and does not take into account past events. It can be used to analyze arrival rates of fog devices by modeling the arrival process of data or tasks at these devices as a stochastic process with specific states and transition probabilities. Finally, this probability matrix is utilized to place the incoming modules at their best destinations for execution. The presented work is compared with First Fit (FF) and local mobile processing models.

Satish et al. [107] have highlighted the problem of efficient offloading of IoT-based tasks in Vehicular Fog Computing (VFC) due to heterogeneity and mobility amongst vehicles. An RL-based agent explores all the possible actions in a greedy manner until it exploits the best action for maximizing long-term reward. The author interprets states as time slots and defines actions as representations of fog vehicles. The reward is determined by a function that combines service time and the energy consumption (EC) of RSUs. The work aims to reduce the response time and computational overhead of RSUs in VFC by utilizing a fuzzy-based RL technique. The presented incorporation of fuzzy with RL addresses the issue of high dimensionality, which occurs due to the increasing vehicle count under RSUs. In addition, the fuzzy if-then rule base is applied to calculate the vehicle weight (Very low, low, low medium, high medium and very high). This is inferred on the basis of input parameters such as process rate {low, medium, high}, staying period (Dwell time) {low, medium, high}, and distance to RSU {far, middle, near}. The suggested method not only expedites the learning process but also enhances long-term rewards compared to prominent Q-Learning, by incorporating greedy heuristics.

TABLE VI

COMPARISON OF PERFORMANCE METRICS FOR TASK OFFLOADING IN FOG/EDGE COMPUTING

Year & Reference	Dataset	Performance Matrix									
		Cost	Throughput	Power Consumpti	Response Time	Resource Utilization	Delay	Energy Consumption	Task Service Time	Computational Time	Weighted Response
2015 [136]	NA	✗	✗	✗	✗	✗	✓	✓	✗	✗	✗
2019 [137]	Google Cluster Trace	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓
2019 [139]	NA	✗	✓	✗	✓	✓	✓	✓	✗	✗	✓
2019 [140]	NA	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗
2019 [138]	NA	✗	✗	✓	✓	✗	✗	✗	✗	✗	✗
2019 [106]	Real time dataset from fog cloud environment	✓	✗	✗	✗	✓	✓	✗	✗	✓	✗
2020 [141]	NA	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗
2021 [107]	Vehicular traces	✗	✗	✗	✓	✗	✗	✓	✓	✗	✗

A secure task offloading mechanism has been proposed by Adam et al. [139]. The author utilizes machine learning techniques for implementing secure offloading in a cloud-fog environment. In the system model under consideration, IoT gateways function as a bridge, facilitating communication between IoT devices and the upper layers (Fog and cloud). To accommodate the resource-constrained nature of IoT devices, a Neuro-fuzzy model is deployed on smart gateways. This hybrid computational model is a fusion of ANN and fuzzy logic systems, which brings out the best useful traits from both models. The capabilities of NNs such as adaptability, learning and generalization can be applied to fuzzy logic, which further imparts transparency to the hybrid model.

- *Adaptability*: It deals with tuning the parameters of a fuzzy logic system via learning or training.
- *Training*: The NN learns from the incoming data and adjusts the membership functions and rule parameters of the Fuzzy Inference System (FIS).
- *Generalizability*: It refers to the system's ability to make accurate predictions or decisions on new, unseen data that was not part of the training dataset.
- *Transparency*: The fuzzy logic component provides interpretability through linguistic variables and fuzzy

rules, making it easier to understand and modify the system.

Hence, this model harnesses the strengths of both approaches to improve modeling and control uncertain and complex use cases. The work employs a Neuro-fuzzy model which considers two factors: (1) Sensor value, and (2) Time. For security evaluations, the predicted value is derived correspondingly. A value exceeding 1.00 indicates a valid reading, while any value below this threshold is deemed invalid. As a result, the neuro-fuzzy logic knowledge base is trained to adjust to incoming IoT workloads. By categorizing the predicted values as valid/invalid, only data from trusted devices is retained. Afterwards, Q-Learning is used for dynamic offloading decisions. Finally, the resource allocation decision is carried out by the PSO algorithm.

TABLE VII

STATE-OF-THE-ART SOLUTIONS FOR TASK OFFLOADING IN FOG/EDGE COMPUTING

Year & Reference	Offload type	Objective Addressed	Technique	Experimental Configuration/ Dataset used	Limitations
2015 [136]	Full	To optimize computational resources in application offloading	MLP	Simulation	The proposed approach is not presented as real-life case study.
2019 [137]	Independent	To optimize Weighted Response Time (WRT) in online task dispatching and scheduling algorithm	OnDisc- speed augmentation model (Heuristic)	Simulation using real-world workload.	Ignores network congestion and checks for accurate predictions.
2019 [139]	Partial	To propose a secure software-based offloading solution for effective QoS in the Fog Cloud environment	PSO RL	Simulation (NS-3)	No implementation in the real scenario
2019 [140]	Full	To minimize cost and energy. Bandit Learning-based offloading	RL	Simulation	Increased number of switching in frequently changing fog environment reduces estimation accuracy.
2019 [138]	Independent	To propose a technique for selecting the best fog node for offloading to minimize response time and power consumption.	CART	Simulation (CloudSim)	The offloading decision did not consider the issues of trust and fault tolerance. Offloading in real IoT applications has been ignored.
2019 [106]	Full	To propose a multiple-tier fog-cloud model for real-time task processing	Accelerated PSO	Simulation	Real-time IoT application parameters are not taken into consideration.
2020 [141]	Partial	To reduce service response time in IoT task offloading for delay-sensitive applications	ACO and PSO	Simulation	Important QoS parameters like power consumption, computation costs have been ignored. Task dependency has been ignored.
2021 [142]	Full	To implement energy efficient IoT-based task offloading in VFCs	Fuzzy RL	Simulation	Overwhelming computational overhead

Another competent online IoT-based task offloading technique has been discussed by Zhu et al. [140], which aims at minimizing the cost corresponding to a specific node along with the latency incurred and energy consumption at the time of task computation. The problem considered is framed as a stochastic programming model in order to handle dynamic system-related parameters. The proposed Bandit Learning-based Offloading of Tasks (BLOT) is based on the Upper Confidence Bound (UCB) which assists in selecting the optimal fog node to offload tasks. This technique holds an edge over the other techniques in the context that it doesn't consider prior knowledge about the system parameters, highlighting the fact that some system values called as bandit feedback appear only at the time of querying the nodes. After offloading the tasks to resource-rich fog nodes, the First In First Out (FIFO) strategy is used to schedule unfinished tasks. The numerical experimentation demonstrates that BLOT serves as an optimal task offloading strategy in dynamic online mode situations. Another work by Adhikari et al. [106], proposed an Accelerated PSO for performing real-time task offloading to apt computing devices as per resource requirements in hierarchical fog-cloud environments. Multi-Objective Offloading (MOO) based on Adaptive PSO (APSO) categorizes the real-time incoming tasks into resourceinsensitive and delay-sensitive category. The former ones are allocated High Computing Fog (HCF) nodes, whereas the later ones concerning latency and cost are executed on Low Computing Fog (LCF) for faster response. The principal reason for using APSO is to reduce error rates and maximize accuracy, along with optimizing other QoS parameters. Further, advanced machine learning strategies can be employed to improve the performance and precision of offloading methods. A smart city-based scenario is considered by Hussein et al. [141] in which a framework has been presented which aims at enabling synchronized actions by efficiently processing voluminous data produced by IoT sensors.

Task offloading is a problem of NP-hard complexity, which means that as the number of IoT devices and fog nodes grows, the complexity level increases exponentially. Therefore, the author presents two nature-inspired metaheuristic techniques, Ant Colony Optimization (ACO) and PSO to assure low-latency services. For instance, in PSO, each particle depicts a potential solution to the task offloading problem. Every iteration moves closer to the global best for the entire population and to its own local best. Typically, a large number of particles are used, making it an ideal choice for such problem scenarios. The results demonstrate significant improvement by ACO task offloading in terms of response time.

E. Resource Scheduling

In general, *scheduling* is the process of sequencing incoming tasks in some order, which is carried out by a special program known as a scheduler. Scheduling in cloud computing is defined at two levels: physical host to VMs and tasks to VMs. Scheduling in the fog landscape involves complexity because of heterogeneous devices and the resource-constraint nature of end devices. This heterogeneity makes resource scheduling an NP-hard optimization problem [143]. It can also be described as the optimal placement of different IoT-based tasks on fog nodes in order to meet real-time QoS requirements by minimizing task execution time and abiding by user SLAs [144].

1) Problem Formulation for Resource Scheduling

An optimal scheduling solution intends to schedule a set of input tasks $\Im=\{\ \Im^1,\Im^2,\ldots,\Im^{j},\ \Im^m\}$ to improve various QoS requirements considering constraints (latency, deadline, SLA, cost). The distributed fog nodes $\mathcal{FN}=\{\ \mathcal{FN}^1,\mathcal{FN}^2\ldots\mathcal{FN}^x\}$ accomplished the demands of IoT requests in the form of computational capability, memory and network usage. Figure 8 depicts resource scheduling in fog environment. In order to schedule the incoming task to its optimal destination, the overall communication cost of executing the task is computed, based on which task scheduling and allocation occur.

Processing Cost Analysis at Local device: Considering the latency $\mathcal{L}_{IoT}^{d,j}$ computed from Eq.7, now energy consumption of processing task

$$\mathcal{EC}_{IoT}^{d,j} = \frac{\Im_{size}^{d,j}}{\psi_{d\_mips}} * \wp_{IoT}^{j} \qquad (17)$$

Where, $\wp_{IoT}^{j}$ denotes the per unit power consumption of j_{th}task.

Now, computing the total processing cost at $Io\mathcal{T}$device can be depicted as:

$$\mathfrak{C}_{IoT}^{d,j} = \omega_1 * \mathcal{L}_{IoT}^{d,j} + \omega_2 * \ \mathcal{EC}_{IoT}^{d,j} \qquad (18)$$

Here, ω_1 and $\omega_2$are weight parameters such that $\omega_1 + \omega_2 = 1$

Processing Cost Analysis at Fog Node: When an IoT device doesn't possess the capacity to handle high-end, latency-critical tasks, it is offloaded to a fog node. The execution delay for FN is computed from Eq. 8 and Eq. 9, where the uptime refers to the offloading time.

$$\mathcal{D}_{\mathcal{FN}}^{d,j} \ = \ \mathcal{T}_{\mathcal{FN}\_Uptime}^{d,j} \ + \ \mathcal{T}_{\mathcal{FN}\_Compute}^{d,j} \qquad (19)$$

Now let $\wp_{\mathcal{FN}}^{j}$ denote the power consumption to process the task at fog node and correspondingly $\wp_{IoT\rightarrow\mathcal{FN}}^{j}$ denote the power consumed during transferring task from IoT device to fog node. Energy consumption of execution request at this layer can be computed as [145]:

$$\mathcal{EC}_{\mathcal{FN}}^{d,j} = (\mathcal{T}_{\mathcal{FN}\_Uptime}^{d,j} \ * \ \wp_{IoT\rightarrow\mathcal{FN}}^{j}) + (\ \mathcal{T}_{\mathcal{FN}\_Compute}^{d,j} \ * \ \wp_{\mathcal{FN}}^{j}) \qquad (20)$$

Hence, the total processing cost at this layer is computed by combining Eq. 19 and 20.

$$\mathfrak{C}_{\mathcal{FN}}^{d,j} = \omega_1 * \mathcal{D}_{\mathcal{FN}}^{d,j} \ + \omega_2 * \mathcal{EC}_{\mathcal{FN}}^{d,j} \qquad (21)$$

Processing Cost Analysis at Cloud data center: It considers offloading compute-intensive tasks to a cloud data center. Computing the execution delay for FN from Eq. 13 and Eq. 14, where the uptime refers to the offloading time.

$$\mathcal{D}_{\mathcal{DS}}^{d,j} \ = \ \mathcal{T}_{\mathcal{DS}\_Uptime}^{d,j} + \mathcal{T}_{\mathcal{DS}\_Compute}^{d,j} \qquad (22)$$

Now let $\wp_{\mathcal{DS}}^{j}$ denote the power consumption to process the task at cloud datacenter and correspondingly $\wp_{IoT\rightarrow\mathcal{DS}}^{j}$ denote the power consumed during transferring task from IoT device to cloud data center. Hence, the energy consumption of execution requests at this layer can be computed as follows:

$$\mathcal{EC}_{\mathcal{DS}}^{d,j} = \ (\ \mathcal{T}_{\mathcal{DS}\_Uptime}^{d,j} * \wp_{IoT\rightarrow\mathcal{DS}}^{j}) + (\mathcal{T}_{\mathcal{DS}\_Compute}^{d,j} \ * \ \wp_{\mathcal{DS}}^{j}) \qquad (23)$$

Finally, total processing cost,

$$\mathfrak{C}_{\mathcal{DS}}^{d,j} = \omega_1 * \mathcal{D}_{\mathcal{DS}}^{d,j} + \omega_2 * \mathcal{EC}_{\mathcal{DS}}^{d,j} \qquad (24)$$

The processing cost is evaluated at all the possible task offloading destinations in order to schedule it to the optimal destination.

2) Resource Scheduling Challenges

The following are the main challenges of resource scheduling:

This article has been accepted for publication in IEEE Communications Surveys & Tutorials. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/COMST.2023.3338015

- **Proficient Workload Deployment:** In scenarios involving transactional workloads with unpredictable job arrivals, such as in e-commerce traffic, resource scheduling becomes increasingly challenging, particularly when there is no prior information available to make optimal decisions [146]. Such situations required AI-based solutions, particularly DRL, in which agents are supplemented with historical information on incoming jobs for effective training. Based on the value of reward, agents improve their decision strategies by updating the model parameters.
- **Optimal Task Mapping:** The goal of optimal task mapping is to find the most suitable allocation of tasks to resources in order to optimize various performance metrics, like minimizing execution time, energy consumption, or enhancing resource utilization. This is a complex problem because it often involves dynamic workloads, varying resource capabilities, and changing task requirements. However, finding the truly optimal task mapping is often computationally expensive and may require considering a large number of variables and constraints. As a result, various algorithms and heuristics are employed to address this challenge and provide effective solutions for resource scheduling in dynamic computing environments.

There are many kinds of task scheduling algorithms, mainly categorized as Static and Dynamic. The static algorithm required advance information about incoming requests along with available resources, including memory, processing capability, bandwidth etc. This category covers FCFS, Shortest Job First (SJF), Round Robin (RR), Minimum Completion Time (MCT), Minimum Execution Time (MET), and many more, which are usually preferred when workloads have small variation. However, when dealing with real-time applications and multicore processors, achieving optimal resource utilization using these deterministic algorithms becomes a challenging task. On the contrary, dynamic task scheduling doesn't require any advance information about the tasks or available resources.

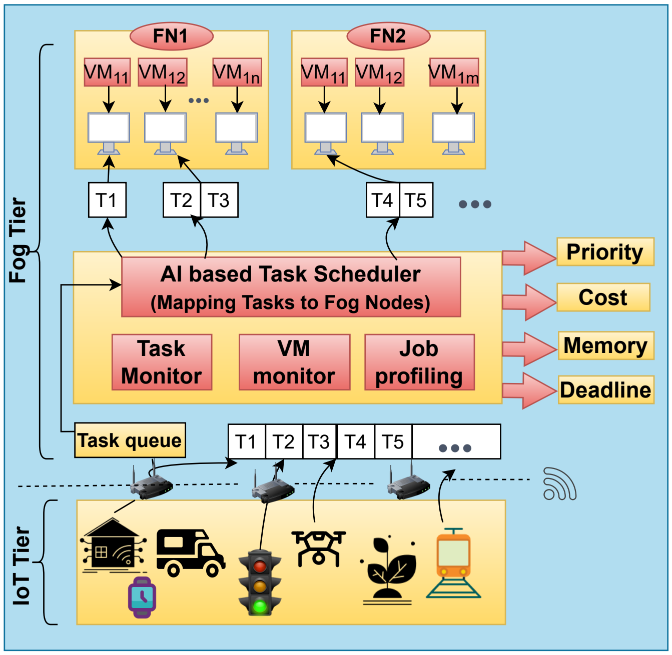

Fig. 8. Resource Scheduling in Fog/Edge computing Environment

3) *Existing Solutions for non AI-based Resource Scheduling*

This section discusses the work that uses static scheduling algorithms along with mathematical models for resource scheduling problems. Li et al. [147] proposed a hybrid computing system for smart factories and Industry 4.0 by proposing a four-level architecture that integrates the historical heritage of computational resources. Furthermore, a two-phase resource scheduling strategy is introduced; the selection of edge computing servers is done by taking into consideration different factors corresponding to low real-time constraints in phase 1. Whereas phase 2 manages cooperation amongst multiple edge servers in order to construct an Edge Server Cluster (ESC), which further comprises the Manufacturing Edge Layer (MEL) cloud. Selecting Algorithms are utilized for ESC known as SAE and CEC (cooperation of edge computing clusters), which works at accomplishing real-time requirements.

A deadline-based framework has been proposed by Fizza et al. [109], in which the incoming tasks are ramified into hard, firm, and soft real-time tasks and then assigned to the most suitable processor for their successful execution. The work has been carried out in an environment consisting of local embedded fog and cloud datacenters. Earliest Deadline First (EDF) is used for task scheduling on the appropriate processor, which works by sorting all the tasks in ascending order of their deadlines. The hard real-time tasks are allocated to run on embedded systems, the firm tasks are executed on fog nodes, and, finally, the soft real-time tasks are processed on cloud-based processors. The proposed architecture demonstrates a 62% improvement in Success Ratio (SR) and a 35% reduction in response time in comparison to scheduling tasks exclusively on the cloud infrastructure. This work focuses on its implications for autonomous cars. Another work by Boveiri et al. [148] proposed a robust solution based on the Max-Min Ant System (MMAS) to solve the multiprocessor task-graph scheduling problem. The algorithm determines the optimal task sequence from the provided task graph and subsequently assigns tasks to available processors in accordance with their sequence. The proposed algorithm outshined traditional methods in terms of makespan time.

4) Existing Solutions for AI-based Resource Scheduling

The process of feature engineering involves the extraction of features, which are characteristics, properties, and attributes, from raw data through domain knowledge [149]. However, the machine learning techniques are based upon manual feature extraction. Afterwards, the model is selected based on chosen features for carrying out variable categorization. Overall, this process becomes time-consuming as the model created depends entirely on the designer's discretionary knowledge. On the contrary, the complex architecture and multiple layers in deep learning models are capable of automatically learning relevant features directly from raw data. This ability to perform feature extraction as part of the learning process has made deep learning particularly powerful in the IoT domain. The potential feature of DL resides in its capability of self-improvement and expansion with increasing data, which is not the same in the case of machine learning. Henceforth, DL is promising for quick feature extraction from voluminous amounts of IoT sensor data. The DL methods have been implemented in the form of different architectures, which are as follows:

- *Convolutional Deep Neural Networks:* These models belong to a family inspired by the way the human brain's visual cortex recognizes objects.

TABLE VIII

COMPARISON OF PERFORMANCE METRICS FOR RESOURCE SCHEDULING IN FOG/EDGE COMPUTING

Year & Reference	Dataset	Performance Matrix														
		Cost	Number of Missed Tasks	Normalized Scheduled Length	Satisfaction Degree	Resource Utilization	Energy Saving Ration	Energy Consumption	Delay	Network Utilization	Makespan	CO2 Emission rate	Fitness	Communication Overhead	Latency	Response time
2019 [148]	Synthetic	✖	✖	✓	✖	✖	✖	✖	✖	✖	✖	✖	✖	✖	✖	✖
2019 [147]	NA	✖	✖	✖	✓	✖	✖	✓	✖	✖	✖	✖	✖	✖	✓	✖
2020 [154]	NA	✖	✖	✖	✖	✓	✓	✓	✖	✖	✖	✖	✖	✖	✖	✖
2021 [156]	NA	✖	✖	✖	✖	✖	✖	✖	✓	✓	✖	✖	✖	✖	✖	✖
2021 [153]	None	✓	✓	✖	✖	✖	✖	✖	✖	✖	✖	✖	✖	✖	✖	✖
2021 [155]	Synthetic	✖	✖	✖	✖	✖	✖	✓	✖	✖	✓	✓	✓	✖	✖	✖
2021 [157]	NA	✖	✖	✖	✖	✖	✖	✓	✖	✖	✖	✖	✖	✓	✖	✖
2022 [109]	CERIT Trace	✖	✖	✖	✖	✖	✖	✖	✓	✖	✖	✖	✖	✖	✖	✓
2022 [110]	Intel Berkeley research lab	✖	✖	✖	✖	✓	✖	✖	✓	✖	✖	✖	✖	✖	✖	✓

Traditional machine learning models depend on input features that can be provided by domain experts or generated through computational feature extraction techniques. In contrast, neural networks automate the feature extraction process. For example, multilayer neural networks create a feature hierarchy by progressively combining low-level features to build high-level features in a layer-wise manner. This approach is well-suited for processing images, where initial layers extract low-level features that are then aggregated to create high-level features. A standard Convolutional Neural Network (CNN) consists of multiple convolutional and pooling layers, followed by one or more fully connected (FC) layers toward the end [150]. For example, surveillance systems employed at various crowded indoor and outdoor locations aim at recognizing abnormal human behavior in society. Carrying out stream analysis manually becomes time-consuming. Hence, CNNs are being utilized to identify objects, people, or specific patterns within video feeds, which is valuable in ensuring safety amongst people (elderly, patients) , decreasing harassment at public places, and safeguarding government assets [151]. Apart from this, CNNs can be optimized and deployed at the edge (on IoT devices) to perform local data processing without the need for constant communication with the cloud.

- *Recurrent Neural Networks (RNNs)*: These are particularly suitable for tasks that involve sequential data and dependencies over time. Predictive models built with RNNs can help optimize resource usage by anticipating demand fluctuations and adjusting resources accordingly. They allow IoT systems to make informed resource allocation decisions, improve operational efficiency, and enhance overall resource management in various domains. For example, these models excel at precisely recognizing complex, non-linear patterns within the input data. They are also effective in capturing temporal workload and node patterns, with their layers facilitating faster learning. Leveraging these models can lead to the optimization of stringent QoS metrics tailored to the application type by employing an adaptive loss function [152].

- *Autoencoder Neural Networks:* These networks operate in an unsupervised manner and are employed for feature extraction and reduction. They have the capacity to identify intrinsic patterns within a dataset and then assign labels to these discovered patterns. Hence, autoencoders reconstruct the datasets, discovering their inherent structure and eventually carrying out dimensionality reduction. This technique can also be utilized for time-series forecasting in IoT applications.

Shadroo et al. [153] proposed a framework to improve performance in terms of response time in an IoT environment. The tasks are clustered based on clustering methods using three Self-Organizing Maps (SOM) methods. In the first method, the primary feature is divided into multiple clusters by SOM. In the second approach, the initial features are transmitted to the Hierarchical Self-Organizing Map (H-SOM) cluster. Subsequently, a deep learning technique, the Autoencoder, which falls under unsupervised learning, is employed for feature extraction. Various features such as task type, priority, task arrival time, data privacy, data heterogeneity, and more are taken into account and then grouped based on these features. The simulation results highlight the superior cost-effectiveness and reduced number of missed tasks achieved by the deep learning method.

TABLE IX

EXISTING SOLUTIONS FOR RESOURCE SCHEDULING IN FOG/EDGE COMPUTING

Year & Reference	Objective	Fog/ Edge Node type	Technique Utilized	Evaluation Tool/ Datasets used	Limitations
2019 [148]	To optimize Normalized Schedule length (NSL) for effective static task-graph scheduling in homogeneous processors	Homogeneous	MMAS	Testbed	Heterogeneous multi-processors are not considered; Implementation covers only limited infrastructural parameters with low edge nodes.
2019 [147]	To design an efficient computing system to maximize resource utilization in industry 4.0	Heterogeneous	Mathematical solution	Real prototype	No real implication; Implemented only as a prototype model.
2020 [154]	To improve resource efficiency in fog computing	Homogeneous	Metaheuristic-Hybrid of IWO and Culture Evolution algorithm	Not mentioned	Trust and privacy of data are not considered.
2021 [156]	To improve network utilization in Mobility-aware approach for fog task scheduling	Homogeneous	Hybrid: PSO + Fuzzy model	Simulation (iFogSim)	The energy-efficiency of fog nodes is not taken into consideration.
2021 [153]	Cost-effective optimal task location selection based on clustering for IoT applications	Heterogeneous	Autoencoder (deep learning method) ANN	Simulation (Matlab)	Only a few numbers of cloud and fog nodes are considered, whereas in a practical scenario, thousands of fog nodes are there.
2021 [155]	To propose an energy-aware model for scheduling tasks in fog computing	Heterogeneous	Metaheuristic (MPA)	Simulation	The proposed work is not implemented as a real case study.
2021 [157]	Efficient management of workflow scheduling in the cloud via MEC	Homogeneous	Metaheuristic (BOA)	Simulation (iFogSim)	Security attacks like DDoS are not considered in the MEC environment.
2022 [109]	Priority-based framework for cost-effective mapping between tasks and processors	Heterogeneous	EDF and Integer Linear Programming Model (DYNAMIC)	Testbed	Task execution in the case of processor failure is not taken into consideration.
2022 [110]	efficient clustering approach to reduce computational complexities	Heterogeneous	DRL and spectral clustering algorithm	Simulation	No real implementation: The issue of energy-efficiency is not considered.

In recent years, metaheuristic methods have become increasingly popular for their ability to discover optimal or nearly optimal solutions to task scheduling challenges in the context of fog computing [65]. The majority of the real-life IoT application scenarios consider task dependencies, which are depicted in the form of DAGs. In such cases, task scheduling is carried out in two phases: (1) Ordering the incoming tasks in some valid sequence; (2) Mapping the tasks to available computing resources. Hence, metaheuristics or evolutionary algorithms are preferred in place of traditional heuristics due to their capability to discover an optimized solution. In lieu of the same, Hosseinian et al. [154] emphasized a power-aware solution utilizing Dynamic Voltage Frequency Scaling (DVFS) and a hybrid metaheuristic-algorithm enabled processor in a fog computing scenario. The purpose of DVFS is to provide appropriate voltage and frequency to the servers, which enables a sustainable solution for scheduling resources. The incoming tasks in the applications are ordered using a hybrid version of Invasive Weed optimization (IWO) and the Cultural Evolution Algorithm (CEA) to retain the precedence constraints for mapping them to an optimum number of resources. The work done not only maximizes task utilization but also provides an energy-efficient approach to task scheduling. Abdel-Basset et al. [155] highlighted the issue of energy efficiency while offloading tasks in fog computing using a metaheuristic technique called the Marine Predators Algorithm (MPA). The original algorithm is modified to propose a modified version (MMPA), which remediates the exploitation capability by utilizing the most recently updated position instead of considering the best one. The problem of resource scheduling is solved using certain steps, such as initialization, evaluation, normalization and scaling, and finally the application of the proposed MMPA technique. The initialization step establishes the predator's vector size, which corresponds to the number of tasks to be scheduled by the fog nodes. The evaluation phase evaluates four parameters: makespan, energy, flow time, and CO_2 emission rate. Then a bi-objective function is formulized, considering makespan and energy-efficiency as significant parameters. Normalization updates the continuous space into discrete values ranging from 0 to 1, and finally, the proposed technique is implicated.

Furthermore, to enhance the MMPA, half of the population is reinitialized and subjected to mutation in the direction of the current best solution, while the remaining half is randomly generated to avoid potential local optima. The algorithm demonstrates improved results in comparison to other existing metaheuristic algorithms in terms of the QoS metrics considered. Hosseinzadeh et al. [157] use an improved version of the Butterfly Optimization Algorithm (BOA) for effective workflow scheduling in the Mobile Edge Computing (MEC) landscape. In addition to enhancing the convergence speed, the proposed technique also solves the problem of local optima by incorporating the Levy flight method. A diverse range of chaotic maps have been applied in the Discrete Version of the Butterfly Optimization (DBOA) algorithm, which results in discrete and randomization in the initial population. In addition, DVFS is incorporated into workflow scheduling to ensure optimal processor frequency and voltage for MEC virtual resources.

A fog task scheduler utilizing the PSO algorithm with fuzzy logic incorporated into the fitness function is mentioned in work done by Javanmardi et al. [156]. The aim of the work is to optimally utilize fog resources to minimize application loop delays. The work refines the scheduling process by considering fog node characteristics (CPU compute, RAM and bandwidth) and task attributes (CPU need, memory required) for meeting the QoS requirements of delay-sensitive and delay-tolerant incoming IoT applications. The fuzzy logic assures that the task scheduler does not get stuck in local minima. Vijayasekaran et al. [110] proposed a two-phase solution. In the first phase, they incorporated the spectral clustering algorithm, an efficient clustering approach, to reduce data overlap and computational complexities within the edge computing framework. In the second phase, they employed deep learning-based resource scheduling to enhance resource utilization and decrease latency in processing user IoT requests. The comparison of performance matrix for resource scheduling problems for various state-of-the-art works is depicted in Table VIII. Along with this, all the investigated studies are depicted in Table IX.

F. Service Placement

Once the incoming requests are mapped by the resource manager, the main challenge is to place the incoming application on a specific fog node in a collaborative cloud-fog IoT computing environment. To resolve this issue, the fog broker, comprising task schedulers and other components as depicted in Figure 9, places them on fog nodes in clusters. Each cluster is governed by a fog master node, responsible for managing slave fog nodes. An optimal Service Placement (SP) solution ensure resource availability to minimize latency and, most importantly, meet deadlines for time-sensitive applications [158]. The task scheduler and SP module collaboratively ensures a fair share of resources for all requesting applications, as in a real-life scenario, multiple applications might compete for the same set of resources within the same time interval.

1) Problem Formulation for Service Placement of Task at Optimal Destination

This phase computes the overall cost and minimizing the cost function. The overall cost associated during decision making phase of service placement $\mathfrak{C}_{Total}^{d,j}$ can be calculated from Eq. 18, 21 and 24 as follows [145]:

$$\mathfrak{C}_{Total}^{d,j} = \sum_{j=1}^{m}\left(1-\Theta_{d,j}^{2}\right) * \mathfrak{C}_{IoT}^{d,j} + \sum_{j=1}^{m}\frac{\Theta_{d,j}(\Theta_{d,j}-1)}{2} * \mathfrak{C}_{\mathcal{FN}}^{d,j} + \sum_{j=1}^{m}\frac{\Theta_{d,j}(\Theta_{d,j}+1)}{2} * \mathfrak{C}_{\mathcal{DS}}^{d,j} \tag{25}$$

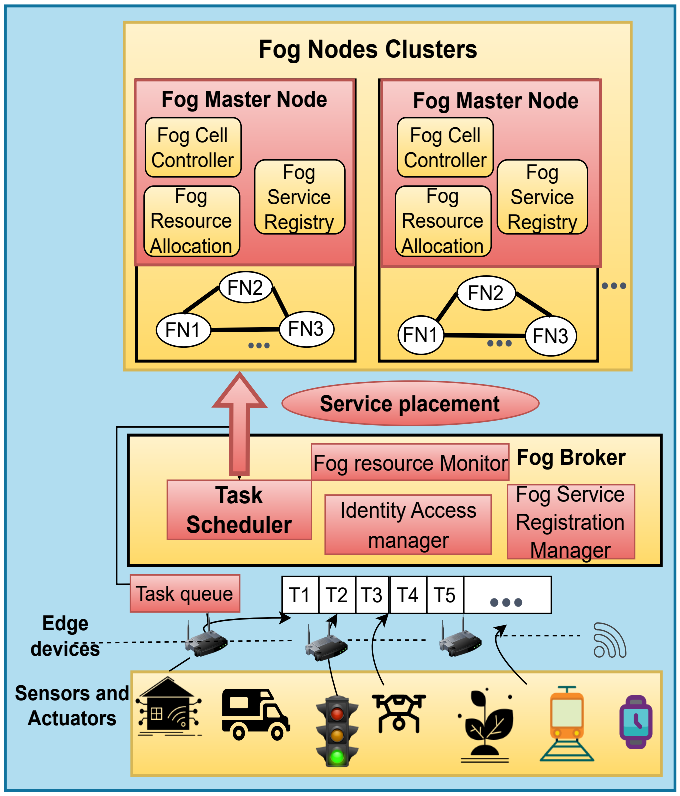

Fig. 9. Service Placement Problem in Fog Computing Environment

In similar manner, the total energy consumed $\mathcal{EC}^{d,j}_{Total}$ from Eq. 17, 20 and 23

$$\mathcal{EC}^{d,j}_{Total} = \sum_{j=1}^{m}(1-\Theta^2_{d,j}) * \mathcal{EC}^{d,j}_{IoT} + \sum_{j=1}^{m}\frac{\Theta_{d,j}(\Theta_{d,j}-1)}{2} * \mathcal{EC}^{d,j}_{\mathcal{FN}} + \sum_{j=1}^{m}\frac{\Theta_{d,j}(\Theta_{d,j}+1)}{2} * \mathcal{EC}^{d,j}_{\mathcal{DS}} \quad (26)$$

Finally, total cost $\vartheta^{d,j}$ can be computed combining Eq. 25 and 26, which is computed as follows:

$$\vartheta^{d,j} = \Psi_1 * \mathfrak{C}^{d,j}_{Total} + \Psi_2 * \mathcal{EC}^{d,j}_{Total} \quad (27)$$

Where, Ψ_1 and $\Psi_2$are weight factors.

The optimization function can be formulated as:

$min\, \vartheta^{d,j}$; subjected to constraints $\Theta_{d,j} \in \{0,\text{-}1,1\}$ (28)

And, the additional constraints are depicted in Eq. 4, 5, 6 and 16.

2) *Service Placement Challenges*

The following are research challenges in service placement:

- **Task Dependencies:** The majority of IoT applications, like augmented reality and image recognition, are modelled as Directed Acyclic Graphs (DAGs) in diverse topologies. The nodes depict the tasks, whereas links delineate the data communication pathway. Consequently, such applications sustain intricate dependencies and constraints when it comes to the decision of service placement. Doubtlessly, SP is a combinational optimization problem [159].
- **Device Heterogeneity:** The device heterogeneity in IoT is result of varying configurations (hardware and software) along with vendor-specific product-related specifications [160]. Moreover, apart from the collection of sensors and actuators, IoT technology thrives for high-end computing tasks such as routing and switching for heavy-duty tasks. Due to the same reason, this aspect is challenging and will be considered during the solution design and implementation phases.

3) *Existing solutions for non AI-based Service Placement*

This section presents some state-of-the-art works based on polynomial and mathematical models. A study conducted by Yu et al. [161] provided a solution to the above-stated problem from a network perspective. In the case of real-time IoT processing, the underlying infrastructure determines the best FN residing on the host, along with the channels where the application data stream will be transmitted. The author presents the provisioning problem in the context of single and multiple applications. A Fully Polynomial-Time Approximation Scheme (FPTAS) is used in the case of single applications, whereas multiple applications can be parallelized amongst various instances. In the case of non-parallel applications, the author proposed a randomized algorithm, where the incoming application is assigned to a specific host only.

4) *Existing solutions for AI-based Service Placement*

Most of the existing work considers centralized DRL agents, that lack generalizability and quick adaptability in heterogeneous and stochastic fog computing environments. DRL agents excel at obtaining optimal policies and long-term rewards with no prior knowledge of the operational environment. However, fog computing operates within a stochastic environment characterized by an extensive state space. In order to learn about the environment, an exploration phase is carried out, which involves trial and error, and correspondingly, the experiences are recorded in the form of a sequence of states, actions and rewards. But this leads to increased exploration costs and time due to the large number of interactions required for optimal agent training. Therefore, centralized DRL is not considered suitable for highly distributed fog computing scenarios. To deal with the same, Goudarzi et al. [111] presents an Experience sharing Distributed-DRL is called the X-DDRL SP technique, in which different sets of experience trajectories are produced in parallel fashion, which further assists the agent in training and learning optimal policies. Multiple agents interact with the environment simultaneously, and the resulting trajectories are regularly sent to the learning system for developing an optimal policy. The actors synchronize their parameters with the learner's during each policy update, and the actors independently conduct their exploration. Hence, this collaborative and distributed learning approach enables experience sharing, which reduces exploration costs and improves the reuse of experience trajectories. In addition, the proposed distributed Actor-Critic-based technique enables the capture of the temporal behavior of incoming data via interconnected layers, and the replay buffer enhances the overall efficiency. The goal of using experience replay is to improve sample efficiency, which refers to how effectively the model learns from the data it collects. By breaking the correlation between experiences, the learning process becomes more efficient and reliable. Finally, RNNs are employed to accurately identify temporal patterns within the data. The efficacy of the proposed model has been demonstrated by extensive simulation and testbed experimentation. The results reveal an 8-16 times faster performance gain in contrast to other DRL-based techniques, along with improved application execution time, cost, and energy consumption. Another work done by Sami et al. [162] provides dynamic solutions inspired by MDP to formulate proactive fog selection and placement solutions. The proposed work utilizes DRL agents to make placement decisions before actual demand occurs.

An energy-efficient dynamic service migration scheme for higher QoE in edge computing is presented by Chen et al. [163]. The study integrates cognitive learning, which encompasses self-learning technologies involving pattern recognition, data mining, and Natural Language Processing (NLP) to simulate human intelligence. Within the suggested Edge Cognitive Computing (ECC) platform, dynamic service migration occurs, guided by the behavioral analysis of mobile user-generated traffic data and network resource conditions. In contrast to the commonly deployed models of cognitive learning, which involve training the machine learning-based models on the cloud and carrying the analytics and inference parts on the edge. The proposed work emphasizes incorporating training and inferencing ML models on the edge itself to further improve latency. ECC not only resolves the problem of computation; furthermore, it aids in knowing “what” to compute and “where” to compute. The author has introduced a framework that comprises both the edge network and edge cognition. The former involves interconnected heterogeneous edge devices, while the latter is divided into two components: (1) the Data Cognitive Engine and (2) the Resource Cognitive Engine. The Data Cognitive Engine is responsible for real-time IoT data analysis and provides network processing capabilities at the network edge. For example, it utilizes Deep Convolutional Networks (DCN) for facial emotion recognition and Hidden Markov Models (HMM) for user mobility prediction. The insights derived from this analysis are then fed to the Resource Cognitive Engine, which reinterprets the data to generate new information. This new information is subsequently utilized by the Data Cognitive Engine. Additionally, the Resource Cognitive Engine receives the analyzed results from the data engine to perform functions such as admission control, resource scheduling, and traffic monitoring. Its caching capability ensures the availability of predicted content at the edge layer in advance, reducing the load and latency on the underlying core network.

TABLE X

COMPARISON OF PERFORMANCE METRICS FOR SERVICE PLACEMENT IN FOG/EDGE COMPUTING

Year & Reference	Dataset	Performance Matrix													
		Cost	Makespan	Flow Time	Running Time	QoS Satisfaction	Delay	Energy Consumption	SLAV Ratio	Fog Utilization	Response Time	Waiting Time	Gross Profit	Service Acceptance Ratio	Resource Utilization
2018 [161]	NA	✓	✗	✗	✓	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗
2018 [164]	NA	✓	✓	✓	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗
2019 [163]	NA	✗	✗	✗	✗	✗	✓	✗	✗	✗	✓	✗	✗	✗	✗
2020 [166]	NA	✗	✗	✗	✗	✓	✗	✗	✓	✗	✗	✓	✓	✗	✗
2021 [167]	NA	✓	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗
2021 [111]	DAGs	✓	✗	✗	✓	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗
2022 [113]	Synthetic	✗	✗	✗	✗	✗	✓	✓	✗	✗	✗	✗	✗	✓	✓
2022 [162]	NASA server logs & Google Cluster Trace	✓	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
2022 [165]	NA	✗	✓	✗	✗	✗	✓	✓	✗	✓	✓	✗	✗	✗	✗

The application of metaheuristic techniques in service allocation has been highlighted by the work of Mishra et al. [164], which implements different nature-inspired algorithms like PSO, Binary PSO (BPSO), and BAT. The problem of assigning an appropriate target VM to an incoming user request is depicted as a bi-objective minimization problem. The author addresses the issue of optimal scheduling of incoming task requests to VMs (execution units in a fog environment). The results were simulated by an in-house simulator and MATLAB, and the above-stated techniques have been compared in terms of energy efficiency and makespan. The study conducted shows that BAT outperforms other Swarm intelligence-based metaheuristics in the mentioned performance matrix. Ghobaei-Arani et al. [113] depicted the SP as an autonomous placement approach for diverse IoT applications onto the fog/edge infrastructure. The author has worked towards ensuring QoS requirements for requesting IoT devices in the form of predicting potential resources for fog using Whale Optimization Algorithms (WOA). This is a stochastic approach inspired by the humpback whale hunting strategy, employed for optimizing computationally challenging problems to achieve efficient resource allocation. Humpback whales belong to a unique category of whales known for using bubble-net feeding techniques in their prey hunting. This technique operates in three distinct phases: encirclement of the target, exploitation, and exploration to cover the entire search space. The effectiveness of this proposed method has been demonstrated through improvements in resource utilization, acceptance ratio, and energy efficiency. The baseline algorithms referred to are Genetic Algorithm (GA), PSO, BAT Algorithm (BAT) and Simulated Annealing (SA)-Fog Service Placement (FSP). The authors have proposed the implications of a hybrid model consisting of the amalgamation of Grey Wolf Optimizer (GWO) and SA for further amelioration of results. Another conceptual framework for IoT service placement in fog environments is proposed by Liu C et al.

TABLE XI

STATE-OF-THE-ART SOLUTIONS FOR SERVICE PLACEMENT IN FOG/EDGE COMPUTING

Year & Reference	Objective	Technique	Experimental Configuration/	Limitations
2018 [161]	To minimize the resolution complexity of IoT applications by optimizing provisioning	Fully polynomial time Approximation	Simulation (C++ based)	A simulation scenario consists of a few nodes, which doesn't depict a real-time scenario.
2018 [164]	To implement sustainable computing in fog servers for Industrial applications	Metaheuristics (PSO, Binary PSO, BAT)	Simulation (MATLAB)	Lacks implementation in areal-world testbed.
2019 [163]	Energy-efficient Dynamic service migration for higher QoE	Cognitive computing	Real Testbed	Highly complex system. Security aspects are not considered, as large amounts of data become prone to breaches.
2020 [166]	Profit-aware placement of IoT applications for integrated fog cloud environments	ILP	Simulation (iFogSim)	Vulnerability to security breaches in fog is not addressed.
2021 [167]	Context-aware decision-making for IIoT applications	Metaheuristics (MGAPSO and EGAPSO, are developed by combining the GA & PSO and Elitism-based GA (EGA & PSO)	Fog TestBed	The proposed approach lacks the capability to serve interdependent IIoT applications.
2021 [111]	Cost-effective and timely decision-making framework for DAG-based IoT applications in heterogeneous fog environments	DRL	Simulation (Python using OpenGym) and TestBed.	Energy efficiency parameter ignored. Mobility amongst fog nodes is not taken into consideration
2022 [113]	Autonomous deployment of IoT applications in fog infrastructure ensure cost and energy effectiveness.	Metaheuristics (WOA)	Simulation	Privacy concerns with IoT devices are not addressed.
2022 [162]	Intelligent Dynamic Fog Service placement for better QoS	DRL MDP	Simulation	The proposed work does not evaluate energy consumption, computational overhead
2022 [165]	A conceptual computing framework using cloud-fog middleware for managing service requests for optimized QoS parameters	Metaheuristic- CSA	Simulation	The proposed work was not implemented in real scenario. The fog node failover has been precluded.

[165], which comprises of Cloud Fog Control Middleware (CFCM). This module is responsible for managing incoming service requests and satisfying constraints. CFCM does all decision-making based upon MADE-k automatic control loops (M-Monitoring, A-Analysis, D-Decision making, E-Execution and k-shared knowledge base). The Fog Service Placement Problem (FSPP) is modelled as a multi-objective dynamic optimization problem based on the Cuckoo Search Algorithm (CSA).

This prominent metaheuristic algorithm works with the behavior of cuckoo birds, which comprises adult cuckoos laying eggs and migrating as per environmental factors in a search for an optimal destination that suits their lives and laying eggs. Considering the implications of this metaheuristic in the context of service placement, the best environment would refer to achieving the global optima of objective function. Nevertheless, apart from improving QoS parameters, the work addressed prioritizing the incoming requests based on task deadlines. Table X depicts the comparison of various QoS parameters used to analyze existing studies in the domain of service placement. The majority of the works have been demonstrated via simulations using a VM-based fog nodes framework in a two-tier fog architecture for optimal service placement in IIoT applications. The authors address the problem of resource provisioning and IoT service placement as a multi-objective optimization problem that aims at minimizing service cost, energy usage, and service time (the sum of processing and communication time) for context-aware decisions in Industry 4.0. A hybrid version of Modified GA and PSO (MGAPSO) has been proposed and implemented in a testbed comprising 20 fog nodes harnessing the master-worker model. The result obtained signifies improvement in terms of the parameters stated above in comparison to First-Fit (FF), Branch & Bound (BB), Double Matching System (DMS), GA, PSO, etc. Finally, The key points of all surveyed work in the domain of service placement have been illustrated in Table XI.

G. Resource Allocation and Load Balancing

The fog computing landscape comprises a load balancer module that manages the distribution of incoming IoT workloads evenly amongst available resources. The dynamism that occurs in fog infrastructure might lead to an increase or decrease in the number of active fog nodes due to variant workloads. Henceforth, the heterogeneity in fog nodes in terms of computational capability is the principal cause for the non-applicability of cloud load balancing solutions in fog computing environments. For a similar reason, from a resource management point of view, load balancing becomes a tough task. A specialized component called a load balancer, after receiving requests from users, runs load-balancing algorithms and then further selects and distributes the requests among VMs available on the underlying fog node. It ascertains that no node is overloaded or underloaded. It is the last step in the resource management lifecycle, which is represented by Figure 10.

1) Problem Formulation for Load Balancing

The Load Balancing Degree (LBD) serves the purpose of distributing workloads among numerous fog nodes to enhance resource efficiency and performance. Its primary objective is to address the challenges associated with resource over-utilization and under-utilization. Assigning workloads to each fog server or cloud data center is crucial, as it is not an ideal solution to have some nodes overloaded while others remain idle. Therefore, it plays a crucial role in distributing and balancing the workload among nodes [168]. It is computed from the makespan time $\mathcal{MST}_{j,x}$, which is as follows:

$$\mathcal{MST}_{j,x}=\sum_{j=1}^{m}\sum_{i=1}^{n+x+1}(\mathcal{T}_{IoT\_Compute}^{d,j}+\mathcal{T}_{\mathcal{FN}\_Compute}^{d,j}+\mathcal{T}_{\mathcal{DS}\_Compute}^{d,j}) \tag{29}$$

$$\mathcal{DLB}=\frac{\sum_{i=1}^{x}\max(\mathcal{MST}_{j,x})-\sum_{i=1}^{x}\min(\mathcal{MST}_{j,x})}{mean\ (\mathcal{MST}_{j,x})} \tag{30}$$

2) Resource Allocation and Load Balancing Challenges

- **Interoperability**: It's important to provide consumers with the choice to migrate from one fog/edge-based platform to another in a customized way with load balancing, keeping in mind factors like cost and functionality [169].
- **Fog/Cloud Dynamism**: Although cloud and fog/edge work in harmony with one another, both dominant technologies rest upon different dynamics. Where the cloud rests on centralized datacenters, in contrast, edge/fog servers are distributed by nature. In addition to this, the capacity of edge devices to dynamically reconfigure themselves for various applications by offloading a variety of tasks introduces an additional layer of dynamism to the edge ecosystem.

3) Existing solutions for non AI-based Resource allocation and Load balancing

This section provides an overview of currently employed approaches for allocating resources. Sthapit et al. [114] have employed a sensor network utilizing a network of queues for implementing computational load balancing in the absence of cloud and fog layers. The scheduling decisions are based on a linear programming model. On executing 100 Monte-Carlo simulations, the results reinforce the fact that the proposed model can compute the incoming jobs efficiently when the total job rate is less than the total computational capability of Node State Information (NSI). Another study by Ahmad et al. [170] describes effective resource utilization in Smart Grids (SGs). The authors have considered a scenario comprising a smart building with multiple apartments acquiring smart gadgets and devices.

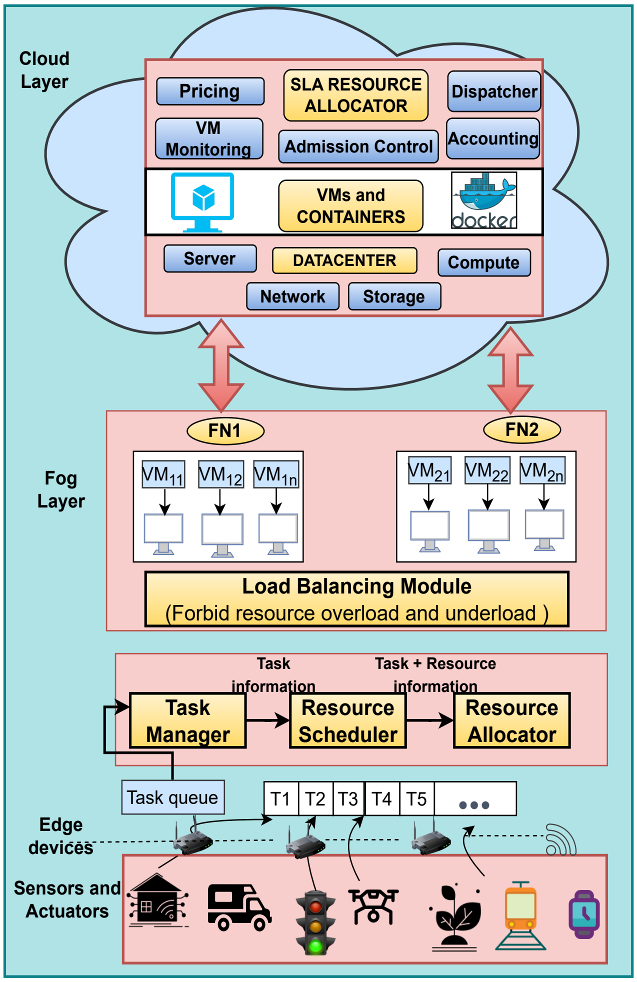

Fig. 10. Load balancing and Resource Allocation in Fog/Edge Computing

The request is fulfilled by the nearest grid, which is further connected to the centralized cloud. The closest datacenter service broker policy is used for selecting the fog nearest to the cluster of apartments. Conventional legacy approaches to allocating resources and load balancing are restricted by their static nature (fixed resource pool), which fail to find the optimal solution in a dynamic and heterogeneous environment. Hence, there is a need to explore dynamic resource allocation approaches that can predict workload changes automatically and adjust in accordance with resource needs. Hence, the subsequent section discusses AI-based solutions to the resource allocation problem.

4) *Existing solutions for AI-based Resource allocation and Load balancing*

As per Forbes 2022 trends, the complexity of devices has been increasing due to the emergence of high-tech gadgets enabling VR, AR, and MR [176]. Extended Reality (XR) and smart applications thrive on the exquisite allocation of distributed resources in the close computing paradigm. This section highlights the existing work on the applicability of AI techniques, including ML, fuzzy logic, and metaheuristics, in this domain.

Singh et al. [177] presented a fuzzy-based load balancer that handles overloading, underloading and disparity in resource utilization. The fuzzy-based three-tier framework is based on software-defined IoT task distribution. The rule-based fuzzy technique enables the handling of diversified incoming traffic patterns among IoT devices. This includes the type of incoming load (video, audio, web, sensor data, etc.), which is generally unstructured in nature, traffic arrival time, etc., in a collaborative cloud-fog environment. The experimentation demonstrates improvements in resource utilization and cost reduction. In addition, the author inferred the superiority of 3-level fuzzy model design in comparison to 5-level and 7-level traffic controllers, supporting the fact that increasing the number of layers results in an overwhelming fuzzy system with overlapping, inconsistent fuzzy rules and nevertheless increases the complexity in terms of the if-then rule base. Talaat et al. [115] presented a resource allocation technique based on effective predictions for ensuring QoS parameters. DRL helps in achieving load balancing amongst fog nodes whereas Probabilistic Neural Networks (PNN) are trained for providing predictions. The authors focus on real-time resource allocation being utilized in the smart healthcare sector to diagnose the probability of heart attack occurrence. It works in three distinct phases: data processing, resource allocation, and making effective predictions. Here, PNN assists in making predictions regarding the occurrence of heart attacks by training the model. To achieve optimized dynamic and real-time allocation of resources, load balancing in fog nodes has been explored using metaheuristic algorithms by Baburao et al. [174]. The study utilizes the concept of containerization in order to create microservices applications, which provide a lightweight solution in comparison to VMs. The Load Balancing (LB) algorithm bifurcates the incoming workload for assigning it to optimal fog node based upon computational resource availability utilizing PSO. Each incoming request from an IoT device is mapped to a particle, and the shortest path is calculated corresponding to the nearest available fog node for mobile IoT users. Lastly, the authors in [175] optimized the resource allocation decision, via autonomic workload prediction of incoming IIoT requests using metaheuristic techniques.

This work predicts the incoming workload using the proposed autoencoder deep learning model. Then, after making predictions about the resources, the incoming job is allocated to an appropriate resource using the Crow Search Algorithm (CSA) by optimizing multiple objectives. As fog nodes are battery-driven, the allocation of incoming applications to a particular node shall consider the same to avoid application execution failover.

TABLE XII
COMPARISON OF PERFORMANCE METRICS FOR ALLOCATING RESOURCES AND LOAD BALANCING IN FOG/EDGE COMPUTING

Year & Reference	Dataset	Performance Matrix														
		Cost	Response Time	Latency	Processing Time	Network Bandwidth Consumption	CPU Availability	Energy Consumption	Load Balancing Degree	Makespan	QoE	Delay	Resource Utilization	Waiting Time	SLAV	Request Rejection Ratio
2018 [170]	NA	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
2018 [114]	Princeton Application Repository for Shared Memory Computers (PARSEC)	✓	✗	✗	✗	✗	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗
2020 [171]	NA	✗	✓	✗	✗	✗	✗	✗	✓	✓	✗	✗	✗	✗	✗	✗
2020 [172]	NA	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗
2020 [173]	NA	✗	✓	✗	✗	✗	✗	✓	✗	✗	✗	✓	✗	✗	✓	✗
2021 [174]	NA	✗	✗	✓	✗	✓	✗	✗	✗	✗	✓	✗	✗	✓	✗	✗
2022 [115]	MOBILE Health (MHEALTH)	✗	✗	✗	✗	✗	✗	✗	✓	✓	✗	✗	✓	✗	✗	✗
2022 [116]	PlanetLab workload	✓	✗	✗	✓	✗	✗	✗	✗	✗	✗	✓	✗	✗	✓	✗
2023 [175]	NA	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓

In such cases, migrating the application to the backup fog node will further increase the response time, failing to satisfy time-sensitive requirements. For the confrontation of the same, Naha et al. [116] present an energy-aware AI-based technique for allocating resources. The fog nodes are fed with information related to CPU utilization and energy usage patterns from datasets, which is later processed using the proposed Multiple Linear Regression (MLR). The proposed approach makes predictions as per constraints such as task deadlines and energy in order to make optimal decisions. The author considers an experimental scenario which comprises independent and dependent variables. Here, the time it takes for the application to complete its execution serves as the primary independent variable, influenced by four other independent variables: CPU usage, node mobility, network communication, and response time. The following MLR equation depicts the calculation of execution completion time:

$$\mathcal{ET}_{\mathrm{A}_x} = \beta_0 + \beta_1\mathcal{CPU}_{\mathcal{U}_{d_r}} + \beta_2\mathcal{DM}_{d_r} + \beta_3\mathcal{NC}_{d_r} + \beta_4\mathcal{RT}_{d_r} + \in \quad (31)$$

Where, $\mathcal{ET}_{\mathrm{A}_x}$: Execution time of Application x

$\mathcal{CPU}_{\mathcal{U}_{d_x}}$: CPU utilization of device r

$\mathcal{DM}_{d_r}$: Device mobility

$\mathcal{NC}_{d_r}$: Network communication of device r

$\mathcal{RT}_{d_r}$: Response time of device r

$\in$: symbolizes the error rate, denoting the variance between the predicted values from the multiple linear regression (MLR) model and the real observations. β_0, $\beta_1, \beta_2, \beta_3$ *and* β_4 are the coefficients that represent the regression line slope. Noteworthy, the purpose of using MLR lies in the fact that the predictor variable depends on multiple quantitatively independent variables; hence, simple regression cannot be used. Hence, based on the energy usage of the fog devices, the best device is selected for energy-aware application processing.

The work has been simulated using the extended CloudSim version along with the Planet Lab workload dataset and compared with the Fog Computing Architecture Network (FOCAN) [178]. The deadline constraint has been accompanied by an energy component using a hybrid approach that minimizes processing delay, time, and SLA violations. Table XII provides a list of various datasets along with QoS comparisons of existing works, which will serve as a benchmark for evaluating new techniques. Li et al. [172] introduced intermediary nodes between the edge and cloud layers.

TABLE XIII
STATE-OF-THE-ART SOLUTIONS FOR RESOURCE ALLOCATION AND LOAD BALANCING IN FOG/EDGE COMPUTING

Year & Reference	Objective	Type	Technique	Experimental Configuration/	Limitations
2018 [170]	To implement effective resource utilization in Smart Grids	Static	Throttled, RR and FF	Simulation (CloudSim)	No real-life based implementation of the proposed study
2018 [114]	To enhance energy efficiency using computational load balancing in edge computing	Static	Linear Programming	Simulation (NS-3)	The performance boost involves a hefty energy cost.
2020 [171]	Towards an efficient Joint Cloud enabled Edge-aware strategy for optimal task deployment using Load balancing	Dynamic	DRL	Simulation (Python on Tensorflow)	Requires multiple experiments to gradually obtain the optimal solution.
2020 [172]	To implement dynamic load balancing through task allocation	Dynamic	Naïve Bayes	Simulation	No real-life based implementation of the proposed study
2020 [173]	Load balancing and managing payloads in cloud and fog zones for better QoS parameters	Dynamic	Fuzzy	Simulation (jperf and fuzzylite api)	Resource waste, Redundant fuzzy rule creation, Routing failures, and network convergence issues have been ignored.
2021 [174]	To enhance the Dynamic Resource Allocation method by balancing load in fog nodes for improved QoE	Dynamic	PSO	Simulation using Docker Desktop and Linpack software	Data security issues are not addressed in context with fog nodes.
2022 [115]	Effective resource allocation using prediction techniques	Dynamic	DL and PNN	Simulation (iFogSim)	Energy efficiency is not considered
2022 [116]	Energy-aware deadline-based Resource Allocation method with a hybrid approach	Dynamic	MLR	Simulation (Extended CloudSim)	No real fog environment implementation
2023 [175]	Optimizing IIoT requests via autonomic workload prediction	Dynamic	Metaheuristic	Simulation (iFogSim)	The security aspect of IIoT data is not taken into consideration. The real implication of the proposed study is missing.

These nodes have the task of gathering real-time data regarding the attributes of fog nodes. This global information monitoring establishes the foundation for classifying edge nodes into categories of light, normal, and heavy workloads. The task assignment module selects the node with the lightest workload, while the other nodes temporarily remain unassigned to tasks, promoting dynamic load balancing. In addition to obtaining the state information of nodes, the intermediary nodes also diminish the pressure on edge nodes. Where most of the works discussed are based upon unilateral computing, implying job deployment either on the edge or on the cloud, Dong et al. [171] proposed a joint Cloud-Edge datacenters approach for resource allocation, which is based on the pruning algorithm and DRL. Firstly, a joint host set is formed by combining the physical hosts from cloud data centers and edge computing centers. In other words, the monitor module receives this “cloud-edge” physical host information along with the task set. Further, the pruning algorithm eliminates fewer promising hosts to create a non-dominated host set, which then serves as the starting point for the DRL process. The work fuses the idea of the Deep Deterministic Policy Gradient (DDPG) of DRL. This technique undergoes behavioral modifications after several iterations of learning from the environment. The main advantage of using pruning as a pre-processing technique is that it reduces the state space, which ultimately reduces the complexity of RL. Hence, making the algorithm more robust during its running phase via effective task deployment and load balancing. The environment is continuously explored for effective computational ability and load balancing. Table XIII depicts the discussed work.

III. REVIEW METHODOLOGY

Now a days, Systematic Literature Reviews (SLR) are becoming paramount, supplementing the minds of researchers with ingenious knowledge by supplying a repository of existing literature in a systematic manner. The SLR encompasses three distinct stages: planning, conducting and reporting the review. In the planning phase, we define the research objectives, pinpoint the research domains, and apply inclusion and exclusion criteria to select the specific research areas of interest.

A. Planning the Review

The process of conducting the review involves identifying primary studies, implementing inclusion and exclusion criteria, and finally generating the results. The electronic databases have been searched extensively, and the respective studies have been reported. In addition to this, some of the leading journals in fog/edge computing that did not appear in electronic searches have been searched manually. The study selection procedure is shown in Figure 11. This survey demystifies the emerging computing paradigms with their architectural framework and integration of various thrust technologies to enhance the QoS parameters in integrated next-generation computing paradigms. This survey discusses the following research questions and the sub-questions identified, including motivation for work.

B. Research Questions

The identification of research questions, as summarized in Table XIV, channels the flow of various processes, hence systematizing the reviewing methodology.

C. Sources of Information

The efficacious content of our work has been collected from several sources. The keywords delved into were Fog/Edge computing, AI, Resource Management, IoT, Software Defined Networking (SDN), Industrial IoT (IIoT), Digital twins, Quantum Computing, Federated Learning, Serverless Computing, 5G and Blockchain. In addition to this, the survey has been emphasized by searching for the role of thrust technology like 5G, blockchain, SDN, Digital Twin, Industry 4.0, IIoT, and Federated Learning (FL). The databases searched are as follows:

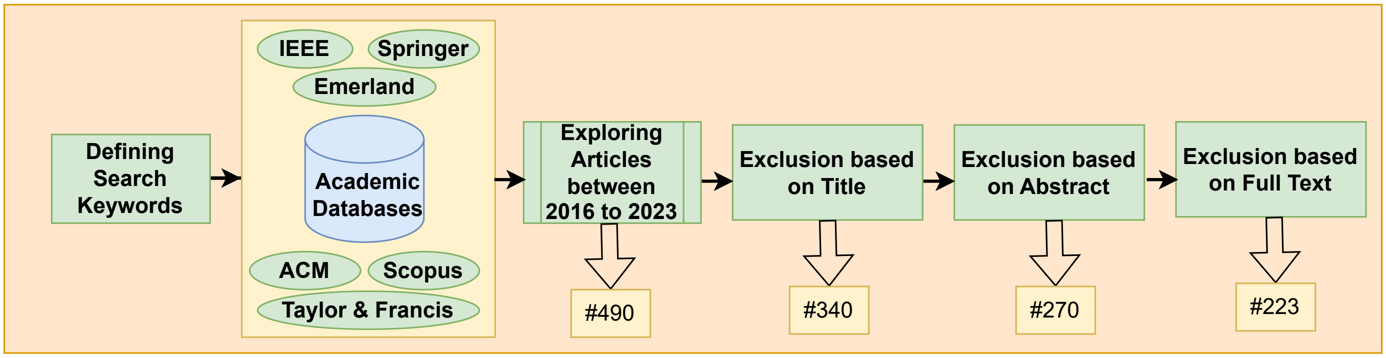

Fig. 11. Selection Criteria used in this SLR

- ACM Digital Library (https://dl.acm.org/journals)
- IEEE Xplore (https://ieeexplore.ieee.org)
- Web of Sciences (https://wos-journal.com/)
- Science Direct (https://www.sciencedirect.com/)
- Taylor and Francis Journal (https://www.tandfonline.com/)
- Elesvier (https://www.elsevier.com)
- Emerland (https://www.emerald.com) and other resources

Additional Sources

- E-Scientific research databases.
- Books and Technical Reports
- National Digital Library

D. Search Criteria

The majority of the searches comprise the keyword "Fog/Edge" and "Resource Management" is included in the abstract. The following strings of words are applied using the Boolean operators AND and OR for combining the keywords, which are as follows:

{("Resource Management" OR "Resource Provisioning" OR "Task Offloading" OR "Resource Scheduling" OR "Task Scheduling" OR "Service Placement" OR "Resource Allocation" OR "Load Balancing") AND ("Artificial Intelligence" OR "Machine Learning") AND ("Fog computing" OR "Edge Computing")}

E. Inclusion and Exclusion Criteria

The implications of AI-based approaches in Fog/Edge computing are relatively new research areas. In addition to this, a major chunk of our referred works lies in the past 6 years, which will assist researchers in enhancing their skills with the latest AI techniques and making an impact in the arena of IoT-assisted fog computing. To refine our work, inclusion and exclusion criteria have been applied to filter out insignificant papers. The above-mentioned search keywords and combinations were framed to narrow down the available academic databases to the most relevant articles. Due to the high potential of Web of Science (WoS) journals, we have taken the research work (journals, transactions, and conferences) indexed in WoS with peer-reviewed methods into consideration for AI-enabled resource management in fog/edge.

This article has been accepted for publication in IEEE Communications Surveys & Tutorials. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/COMST.2023.3338015

Initially, 490+ papers were considered at the start of the process. But to find eminent publications, an extensive screening process was carried out to filter out non-peer-reviewed articles, conferences, and book chapters that were not capable of contributing to our research domain. Henceforth, our subsequent steps shortlisted 223 potentials.

F. *Quality Assessment*

To compile the best available research on this topic, we used a systematic review approach in accordance with the "Centre for Reviews and Dissemination (CRD) guidelines" provided by Kitchenham [173]. Additionally, there are a number of academic articles and conference proceedings on AI for fog/edge computing. After implementing the criteria for exclusion and inclusion, we performed a quality assessment of the articles that fulfilled the standards to determine which were most deserving for further review. We used the criteria set by the CRD to assess the research's overall quality, including its fairness, internal cohesion, and neutrality.

IV. EXPLORING FUTURE RESEARCH DIRECTIONS: DEMYSTIFYING THRUST TECHNOLOGY IN INTEGRATION WITH FOG/EDGE

In this section, we spotlight the integration of emerging paradigms with thrust technology, which is evolving as a potential research trend. This integration is taking the existing functionality of fog computing to the next level by inculcating advanced and sophisticated techniques such as providing a hyper-personalized space for privacy preservation of

TABLE XIV
RESEARCH QUESTIONS AND MOTIVATION

S. No.	Research question	Motivation
RQ1	What are the main issues in the IoT-based paradigm?	This question investigates various open issues in the IoT paradigm, and the need to include fog/edge computing to elevate the resource-constrained nature of IoT devices.
RQ2	What are the various ingredients in the realm of resource management, along with their research challenges?	This research question helps to investigate and identify different subareas in the realm of resource management in Fog/Edge.
RQ3	What is the contemporary status of AI-based solutions in the arena of resource management in fog/edge computing?	The mentioned question is beneficial for determining the recent state-of-the-art works done along with the application of AI-based resource management in Fog/Edge.
RQ4	What is the basis for full and partial offloading of tasks in AI-enabled resource management in fog/edge computing?	The purpose of this question is to assist researchers regarding the decision on task offloading.
RQ5	What is the rationale for deciding where to offload tasks corresponding to an incoming request?	This question helps in understanding the various pre-requisites for task offloading decisions, which include what to offload, where to offload, how to offload and when to offload.
RQ6	Determining the role and need to integrate predictive analysis capability into emerging paradigms via AI, ML, etc.	This section discusses the capability of ML/DL to effectively analyze and quickly extract features from voluminous data generated from IoT devices.
RQ7	What is the basis for the implementation of the upcoming Fog/Edge computing paradigms?	The implementation method: Simulation or testbed has been discussed. Supporting tools and APIs have also been mentioned to assist researchers in the future implementation of their proposed work.
RQ8	What is the status of upcoming thrust technology in emerging computing paradigms?	The motive of this question is to explore numerous thrust technologies, such as 5G, Digital Twin, IIoT, Blockchain, SDN and Federated Learning and to learn how they are being integrated with Fog/Edge computing.
RQ9	What are the most prominent application areas of IoT-enabled Edge/Fog computing?	The survey discusses some real-time applications of IoT-enabled Fog/Edge in the form of case studies.
RQ10	How can the efficiency of AI-enabled Fog/Edge computing be computed and what are the key performance indicators?	The survey presents various parameters in tabular format to gauge the effectiveness of AI-based resource management techniques in Fog/Edge computing.
RQ11	How will thrust technology and AI impact Fog/Edge computing in the future?	It helps to find out the research directions and lessons learned in the field of Fog/Edge.
RQ12	What are the proposed solutions to the research challenges in a collaborative cloud-Fog-IoT environment?	The survey equips its readers with trending techniques and cutting-edge technologies that will assist in imparting business intelligence in IoT-based applications.

IoT data, containerization, high-end telecommunication networking capabilities, and many more.

A. Integration of Fog/Edge with Serverless Computing

The ideology of serverless computing can be considered a pay-as-you-go model in the cloud paradigm, along with leading-edge technologies such as microservices, containerization, event-driven modelling, Function-as-a-Service (FaaS), and Backend-as-a-Service (BaaS) [179]. The Cloud subscribers bear the consequences of paying for the resources allocated, not for the resources utilized, and the scalability drawback, wherein the configuration of auto-scalars is preordained based upon the load profile and application characteristics. Furthermore, the rapid shift from monolithic systems to SOA and then the final paradigm shift to microservices applications, recognized the possibility of running small pieces of code as functions, well-known as FaaS, hence leading to the emergence of serverless computing [180].

Apart from easing out the process of server management, it executes different events that simplify the backend code. For instance, the generation of an IoT event that originated from a home security sensor might invoke a lambda function that further notifies the user on the end device. Another example might include creating a database event that triggers a serverless function and then finally prompting the customer with an email. Other events might include HTTP requests, file uploads, or database analytics. Therefore, it can be called the next promising shift in the cloud era, with the full realization of cloud capabilities comprising disjointed functions called Lambda functions. Although the term was originally coined for the cloud, serverless is locating itself in the fog/edge landscape, with IoT as its prime partner. Seamless integration of serverless computing with fog/edge is complex due to the significantly larger count of fog nodes and their distributed nature [181]. Most of the IoT devices are connected through various communication mediums such as Wi-Fi, Bluetooth, Zigbee [182], and 2/3/4/5G/LTE networks. These devices create incoming tasks using messaging protocols like HTTP and Application Programming Interfaces (APIs), following event-driven principles. However, due to the distributed nature of IoT devices, such as the temperature sensors spread throughout a smart city's manufacturing plant, relying solely on HTTP may not be sufficient [101]. Hence, the Serverless edge enables the capability to push/publish via communication protocols comprising Message Queuing Telemetry Transport (MQTT), MQTT-SN and Data Distribution Service (DSS) [105].

B. Integration of Fog/Edge with 5G

The emergence of 5G has ushered in a wide range of unprecedented applications, offering enhanced mobile broadband, ultra-reliable connections, improved data rates, low latency, and extensive device connectivity. 5G serves as the fundamental pillar for enabling the new AIoT economy. Corporations are looking at the IoT as the next industrial revolution with its capability of imparting intelligence into their operations, building smart factories, hospitals, campuses, cities, businesses etc. These application domains necessitate advanced communication services ensuring data security, real-time communication, and widespread device connectivity. But at the same time, it becomes challenging to manage and serve the enormous number of devices with the existing cellular network architecture. To maintain the QoS parameters, the service providers are working towards expanding their Base Stations (BS). BS provides a consistent number of resources all the time, whereas mobile users utilize services only at discrete intervals, which results in futile resource management.

Resource management has always been a perplexing problem in cellular networks because of the underlying heterogeneous resources and dynamic workload requirements. To help with the same, AI is one of the most dominant technologies and plays a significant role in almost all spheres of industrial domain applications. Concomitantly, it requires extra compute, memory resources, and training time for its decision-making. In order to transfer big data seamlessly across Baseband Units (BBUs) and Remote Radio Head (RRH**)**, edge computing-enabled 5G networks are bringing cloud capabilities in close proximity to the end user. This strategic placement effectively mitigates the inherent challenges associated with high latency and security gaps found in conventional architectures. [184]. This integration will not only decrease the transmission of useless data but also solve bottleneck issues like congestion. The edge devices (IoT gateways, routing switches) will assist the functionality of AI by filtering out useless data beforehand, which will result in a reduction in transmission backhaul. The incorporation of the Edge layer will assist BBU in autonomously configuring its underlying resources under real-time changes in its environment.

However, the swift expansion of tech-savvy devices along with IoT devices is eventually overburdening the existing portable remote sensor networks. Even after the incorporation of AI with 5G & beyond and Industry 4.0, real-life application domains like medical service management and transportation frameworks suffer from some key issues like radio resource management optimization and interference management [185].

C. Integration of Fog/Edge with IIoT

It refers to the utilization of selected IoT sensors and actuators in the industrial arena to enhance manufacturing and industrial processing capabilities without human intervention

[186]. It focuses on digitizing and integrating all essential physical processes across the entire organization [187]. The IIoT, also called the Industrial Internet, is a metamorphic change after the industrial and Internet revolutions that drastically impacts the way industries function. It is a significant paradigm shift from traditional centrally controlled machines to more decentralized functioning capabilities. It utilizes a new software-defined machine framework that virtualizes the machine's functionality in software, enabling seamless monitoring and management of industrial assets via remote access.

With the advent of Industry 4.0, a prodigious amount of time- and delay-sensitive data is being generated by machines. Moreover, the sensors embedded in industrial equipment are resource-constrained and battery-driven. Henceforth, executing all the computational workload on these devices might drain them quickly. To realize the system from a resource perspective, the workload compute capability of IoT devices can be boosted by introducing a fog layer. As discussed by Sengupta et al. [188] the IoT devices transmit raw data to the closest fog node through Wi-Fi access points. For data with strict time sensitivity, the fog node analyzes the incoming request and sends a control command back to the respective device. In contrast, data with lower time constraints is sent to the cloud for extended storage and large-scale data analytics. The integration of the fog layer enhances resource performance by extending the battery life of sensing devices by offloading computation-intensive tasks to the cloud. This integration also diminishes the trust dependency on the cloud by utilizing it for storage and archival purposes. The decentralized nature of FNs imposes several challenges, such as global monitoring and controlling the computing states of FNs. Furthermore, the diversified nature of tasks in fog-enabled IIoT introduces a misalignment between the anticipated computational efficiency and the allocated resources on fog nodes.

The problem has been addressed in the form of service popularity-based smart partitioning of resources for fog-enabled industrial IoT [189]. Still, fog-enabled IIoT suffers from trust establishment problems. To resolve these problems, the same blockchain-based security service architecture is proposed by Hewa et al. [190]. The proposed work facilitates cloud manufacturing equipment authentication, channel privacy protection, and the unlikability of transactional data over blockchain records.

Use Cases: This subsection discusses the practical implications of Fog/Edge in IIoT.

- **Smart Pump**: An IIoT-enabled smart pump equipped with predictive analysis, maintenance, and machine learning is capable of imagining a potential failover. Such systems prepare in advance for the downtime instead of encountering a whole system failover.
- **Smart metering:** This capability can be exploited in the arenas of measuring energy, natural gas, water consumption, and many more.
- **Fleet management:** It holds applications in smart transportation systems where, for instance, the most efficient routes are calculated for waste management collection vehicles. This system is further strengthened by real-time traffic feeds and efficiency algorithms.
- **Jet Engines:** A fleet of locomotives, particularly airplanes equipped with IIoT sensors, can foresee fuel requirements or any other type of failover. This technology works with insight to impart zero unplanned downtime. With its advent, the maintenance course of the plane can be predicted even before the plane lands, thereby preventing unscheduled maintenance events.

D. Integration of Fog/Edge with Blockchain

IoT, in conjunction with the fast-propelling Industry 4.0, requires the gathering, analysis and sharing of raw data from which valuable information is extracted. Nevertheless, IoT security and data privacy remain major issues as the accumulated data is exposed to vulnerabilities. Any malicious user can track devices and continuously listen to conversations between IoT devices, which ultimately leads to a breach of privacy. To add to that, upcoming researchers have proposed the integration of blockchain in order to revolutionize the business process, which anticipates greater data integrity in fog-enabled IoT networks [191]. The integration of blockchain in fog-enabled IoT networks is revolutionizing broad spectrum of fields such as industries, retail, finance, the public sector, and above all, technological aspects, as depicted in Figure 12.

Blockchain is prominent for trusted transactions based on the concept of a distributed, shared, replicated and permissioned ledger, where the transactions are provably endorsed by relevant participants. It constitutes independent computers called nodes, which enable the sharing and synchronization of transactions in their corresponding electronic ledgers instead of maintaining a centralized ledger. The shared ledger contains: (1) immutable blocks containing a set of transactions that are chained together in append-only mode. The distributed ledger acts as the building block of the "Internet of Value," where value is transferred from peer to peer. Here, value could be any identity, health information, personal data, and many more. (2) World State: stores the current state of assets, which includes an ordinary database (key/value store).

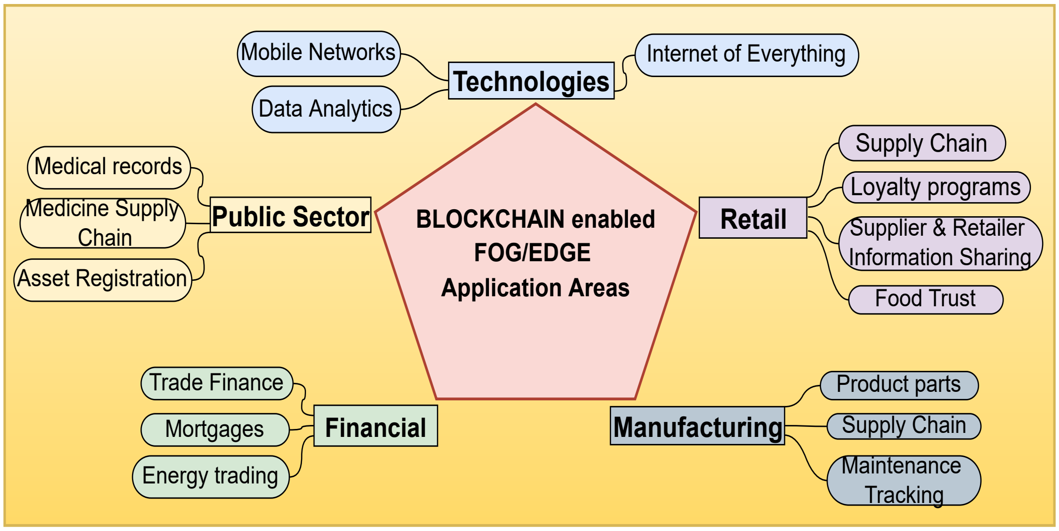

Fig. 12. Applications of Blockchain-enabled Fog/Edge computing

The architecture flow of integration of blockchain with IIoT, Industry 4.0 in fog, and edge-enabled IoT networks is demonstrated in Figure 13, which depicts how digital trust can be implemented in the healthcare domain via blockchain. Initially, all the active actors, including healthcare workers (doctors, nurses) and patients, are registered with the main healthcare service provider. The parameters (like ID, name, age and gender) are stored as Ethereum addresses. The patients authenticate themselves in the proposed framework and are authorized to consult their own data at any time. Granting the right permissions is a vital ingredient for the efficient implementation of the proposed system. Afterwards, data processing is done via specific sensors relating to certain diseases, for instance, electrocardiogram (ECG) and blood-oxygen saturation (SpO2) sensors for heart patients, sugar levels for diabetic patients and so on. This phase involves data acquisition by various sensors and then storing the same in a blockchain. Storing such sensitive information directly on the cloud is not recommended as it can be deleted or tampered with by malicious actions which can be life-threatening for the patient. Noteworthy, such data storage requires resources on the cloud and blockchain; hence, periodic data storage like acquiring temperature sensor readings every 1 or 2 minutes is not a good choice. Hence, only vital values exceeding the threshold value are stored in the blockchain for further analysis by the application of AI algorithms. Afterwards, data is transferred via gateway to the edge/fog devices. This is where data processing takes place for real-time data, such as storing data in blockchain format and recovering patients' parameters from blockchain. Blockchain-enabled fog paradigms are cooperative rather than competitive in nature. At this stage, smart contracts perform the verification of credentials at the fog layer. Finally, processed results can be accessed via a web application or mobile application. The real-time parameters are accessed, and the reports are uploaded in Interplanetary File System (IPFS) format to the blockchain, which can further be stored on the cloud for future monitoring of the patient's parameters [192], [193].

E. Integration of Fog/Edge with Digital Twin

With the inception of Industry 4.0, the journey towards automating traditional industrial practices emerged. Gartner estimates that by 2027, over 40 percent of the major industrial companies will utilize Digital Twins, in order to improve their revenue and operational effectiveness [223]. With a digital twin as its counterpart, which aims at replicating elements, processes, functions, and dynamics of the physical world come into a digital counterpart. The integration of the digital twin eases out processes such as monitoring, testing, and evaluation, along with predictive analytics of complex, which otherwise would have been out of question using traditional simulations [193]. The first pragmatic model was built in the form of a virtual spacecraft by NASA in 2012, which received real-time inputs from sensors [194]. With the latest developments in ML, AI, VR, AR, next-generation mobile communications (beyond 5G), Transfer learning and many more along with emerging computing paradigms, the digital twin has been reshaped with its enhanced capabilities, covering a wide range of domains including logistics, smart cities, smart manufacturing, healthcare, and robotics under its umbrella.

Talking about its legitimate applicability, Digital Twins technology is incorporated into IoT devices that are confined to a particular region. For such low-powered devices, the cloud alone cannot assure optimal QoS services for latency and real-time devices.

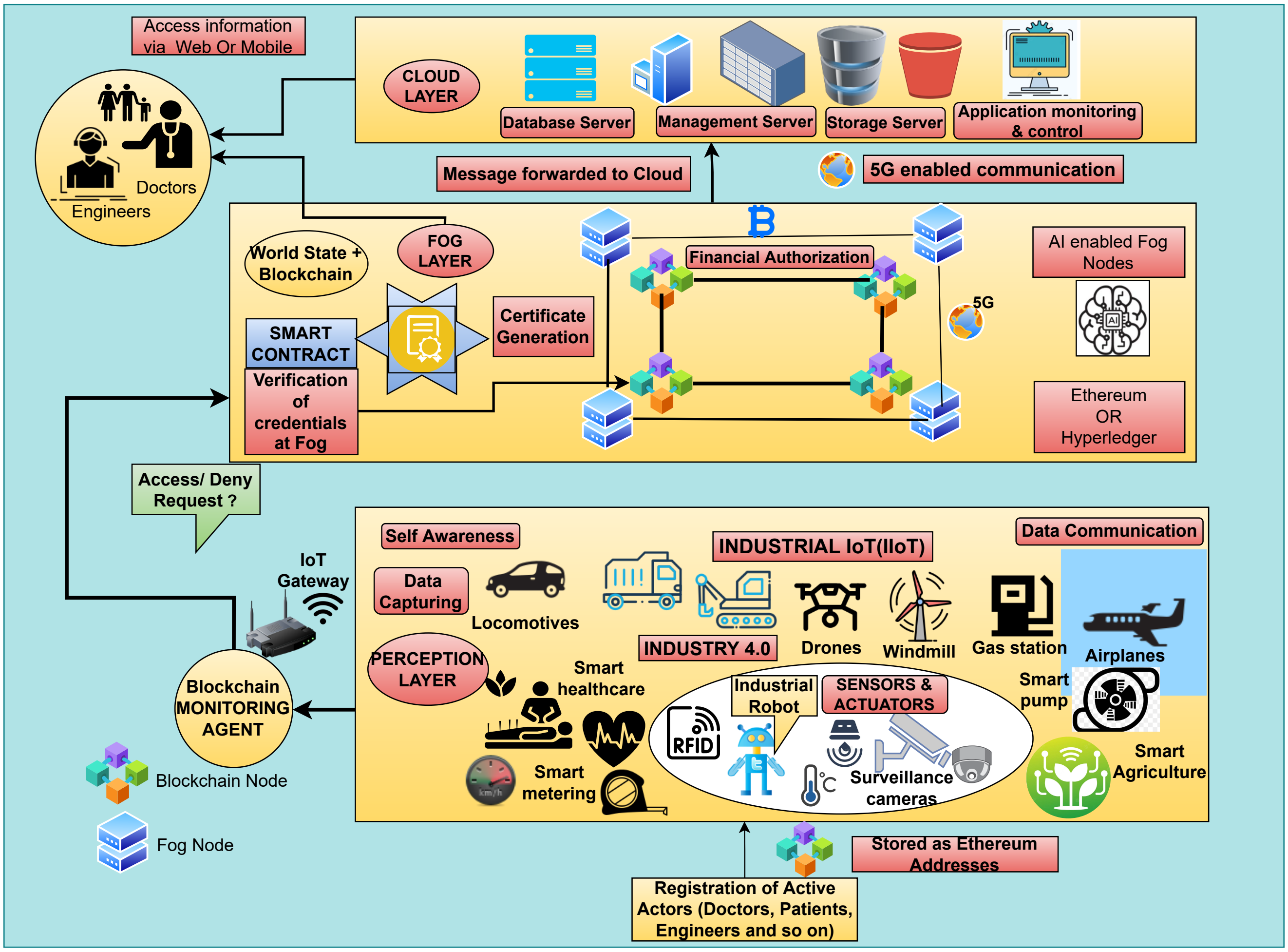

Fig. 13. Integration of IIoT, Industry 4.0, Blockchain and 5G with Fog/Edge computing

Hence, Fog/Edge becomes indispensable for the inculcation of the framework of the Digital Twin by reducing connectivity and latency issues in networks, thereby empowering the system with robustness.

F. Integration of Fog/Edge with Quantum Computing

The idea emerged from transistor computing when Richard Feynman, a Nobel Prize-winning physicist, realized that atoms comprising transistors can exist in both high and low states simultaneously. In contrast to the classical bit, which holds only two values, zero and one, a quantum bit, or qubit, holds a complex coefficient value. The term envisioned for this behavior was coined Quantum Superposition State. Thus, quantum bits, or qubits (which hold a complex coefficient value describing a particular state), came into the picture, which are the fundamental building blocks of quantum computers [195]. Quantum Computing as a technology is still in its infancy, but it can be utilized to calculate intractable things around us. The fidelity of this computing paradigm is going to touch many applications and industry verticals. Currently, serverless computing-enabled fog/edge frameworks, driving function as a service, necessitate quantum computing for processing massive computations for dynamic provisioning and load balancing of underlying resources [196].

G. Integration of Fog/Edge with Federated Learning

Assuring optimal resource management with known execution times in a classical edge computing landscape is a tedious task [197]. It becomes practically infeasible to estimate the execution time due to the complex framework of the edge server. Furthermore, processing colossal amount of data aggregated by cameras, GPS, sonar, and IMU within the existing framework of fog/edge is challenging without the integration of intelligent paradigms [198]. With AI almost influencing all aspects of our lives, the traditional AI methods involve training the models on data aggregated from several IoT/edge devices on a centralized cloud server. For instance, consider the smart city scenario where AI and ML techniques consolidate the results for better predictions and enhance the user experience. Nevertheless, data is stored at a centralized location, and the AI techniques

This article has been accepted for publication in IEEE Communications Surveys & Tutorials. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/COMST.2023.3338015

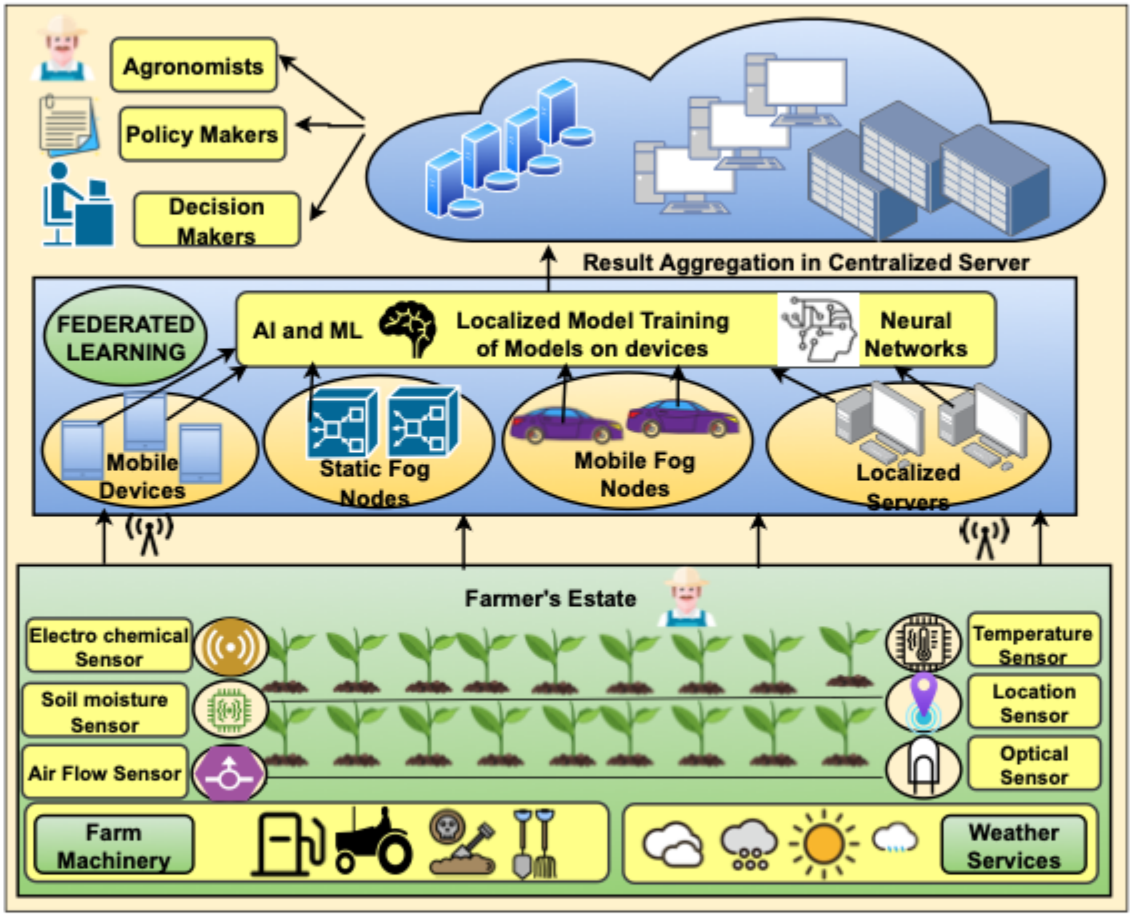

Fig. 14. Fog-Assisted Federated Learning Framework for Agriculture

rely on this training data in order to forecast trends and patterns [199]. Nonetheless, this method encounters various challenges including privacy concerns, data security, regulatory compliance etc. The solution to this problem lies in training the model on the device itself instead of centralized server.

To serve the same, Federated Learning (FL) comes into play, which provides hyper-personalized space, low cloud infrastructure overhead, and prominent privacy preservation while minimizing latency. FL can be treated as a decentralized form of machine learning, which creates a shared model in place of a central data model. The new models are being trained collaboratively on the edge, where the data never leaves the personalized device. Although the devices and machines train several models at distributed locations in parallel and send their collaborative results to a centralized server to create a machine learning model [200]. Therefore, FL leverages both the distribution of data and computational resources while safeguarding data privacy [199].

For instance, Saha et al. [201] illustrated the implications of FL via an irrigation scheduling application where the deployed IoT sensors such as humidity, temperature, moisture, air flow, and so on forward, the parameters via edge devices. The demographic versatility across fields enables each edge device to update its local model utilizing on-device local data. Global aggregation of data on centralized servers results in inefficient training models that are even more susceptible to malicious attacks. Hence, fog nodes act as local aggregators, which send the global aggregators to centralized cloud servers. The implications of an integrated framework are shown in Figure 14.

H. Integration of Fog/Edge with Software Defined Network (SDN)

SDN is a new architecture wherein the control plane is separated from the data plane and consists of two primary components: the SDN switch and the controller. Each SDN switch comprises a flow table, which defines the actions to be applied to the packets that enter it. A match criterion for each entry in the flow table is defined over the IP source, IP destination, and protocol fields. The actions are implicated once the match criteria are satisfied.

TABLE XV
DEMYSTIFYING THRUST TECHNOLOGY WITH FOG/EDGE COMPUTING

Year & Reference	Objective	Thrust Technology integrated with Fog/Edge								Implementation		QoS Addressed						
		Serverless	IIoT	Industry 4.0	SDN	Blockchain	Federated Learning	5G	AI	Simulation	Testbed	Security	Latency	Fault tolerance	Response time	Energy consumption	Throughput	Cost
2018 [189]	Smart resource partitioning based on service priority in fog-enabled IIoT	✗	✓	✗	✗	✗	✗	✗	✗	✓	✗	✗	✓	✓	✓	✗	✗	✓
2019 [191]	To minimize the communication delay between IoT and ensure uniform resource distribution in an IoT-enabled network	✗	✗	✗	✓	✓	✗	✗	✗	-	-	✗	✗	✗	✗	✗	✗	✗
2020 [202]	Secure and energy efficient framework for IoT Networks	✗	✗	✗	✓	✓	✗	✗	✗	✓	✗	✓	✓	✗	✓	✓	✓	✓
2021 [201]	Enabling distributed learning on resource-constrained IoT devices	✗	✗	✗	✗	✗	✓	✗	✓	✓	✗	✗	✓	✗	✗	✓	✗	✗
2021 [185]	AI-enabled novel architecture for smart healthcare for better resource management	✗	✗	✓	✗	✗	✗	✓	✓	✓	✗	✗	✓	✗	✗	✓	✓	✗
2021 [188]	Enhancing security via fog-based architecture and towards judicial usage of resources in IIoT	✗	✓	✓	✗	✗	✗	✗	✗	✓	✓	✓	✓	✗	✗	✗	✗	✗
2022 [190]	Secure blockchain-enabled fog computing model for manufacturing equipment clusters	✗	✓	✓	✗	✓	✗	✓	✗	✓	✗	✓	✓	✗	✗	✗	✗	✗
2022 [192]	Secure and authorized data sharing boosts QoS requirements in terms of cost, security and reliability in the healthcare Sector	✗	✗	✗	✗	✓	✗	✗	✗	✓	✗	✓	✗	✗	✗	✓	✗	✓
2022 [205]	To optimize Resource usage via effective vehicle selection in Vehicle Edge computing	✗	✗	✗	✗	✗	✓	✗	✓	✓	✗	✗	✗	✗	✗	✓	✗	✓
2022 [206]	To implement Energy-aware Task Offloading in massive IoT Edge Networks	✗	✗	✗	✓	✓	✗	✓	✓	✓	✗	✗	✓	✗	✗	✗	✓	✗

Whereas the SDN controller brings forth full control over the network, communication flow, enabling remote control, elevated flexibility, and programming capability [202].

The integration of SDN with IoT boosts network performance by providing the management of ephemeral network states in a centralized control model. Besides, the SDN controller acts as a centralized controller for IoT networks, assisting in the monitoring and management of heterogeneous IoT devices [203]. The research is trending towards the usage of SDN for optimized IoT management in alliance with blockchain to improve the security aspect. Finally, Table XV presents an overview of various studies that delve into the incorporation of thrust technology in fog/edge computing.

V. RESULTS AND ANALYSIS

Our work emphasizes the systematic assessment and discussion of various articles based on the prevailing status of resource management issues in fog/edge computing, along with the identification of its collaborative thrust technology. Our study includes numerous driving forces that are leaving a remarkable impact on emerging computing paradigms, presented as open challenges and from a future research perspective.

Throughout, 490+ articles were collected, out of which 223 have been shortlisted after extensive selection. The articles highlight the existing state-of-the-art work carried out in the resource management domain based upon non-AI and AI-based technologies. Most of the selected articles cover the period from 2016 to 2023. As illustrated in Figure 15, a major chunk of our referred articles are from the past 6 years. The organization and methodology of the article are motivated by the systematic literature review procedure [204]. It can be ascertained that the formulated research questions serve as a principal solution for elucidating various RM-related issues, hence heading the flow of the review methodology.

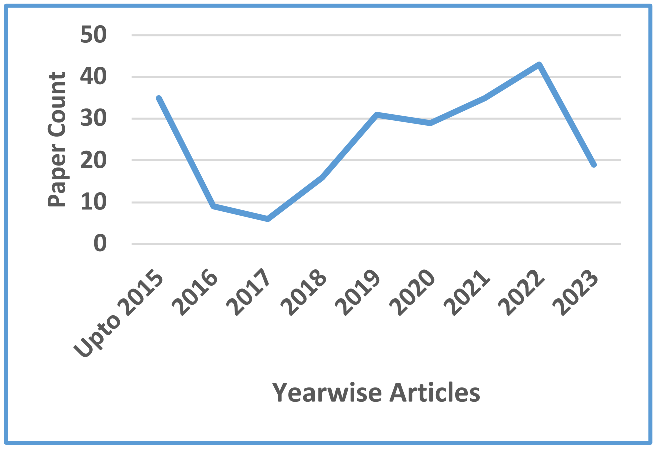

Fig. 15. Year-wise Publications based on Resource Management in Fog/Edge Computing

Furthermore, we meticulously reviewed each article and bifurcated it into five fields: review, SLR, Implementation platform (Real/Testbed or Simulation) and book chapters, as represented in Figure 16. In addition to this, Figure 17 states comparison statistics in the context of evaluation tools, filtering out the papers based on real implementation. This indicates that a significant portion of the papers examined did not specify the simulator tool they used and, as a result, have been categorized as part of the "other" group. Besides, the majority of them utilized iFogSim, CloudSim, and Python-based implementations with 17%, 12%, and 8%, respectively. It has been observed that few infrastructure platforms exist for pursuing real-time fog computing research work. Hence, there is a need to expedite research towards developing realistic Fog testbeds for evaluating the results of deployed AI models.

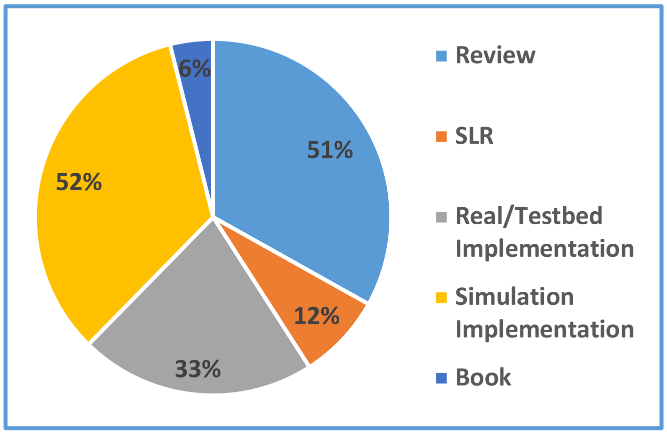

Fig. 16. Type of Studies in the Selected Papers

The search criteria play a significant role in the review methodology. In reference to the same context, Figure 18 represents a bifurcation of selected papers based upon prominent publishers comprising IEEE, Elsevier, ACM, Springer, and other sources such as Web of Sciences, Science Direct, Taylor and Francis Journal, Scopus, Google Scholar, Research Gate, Springer Link, Emerland, and other resources like scientific electronic research databases. It is concluded that the majority of the selected articles are published by IEEE journals, transactions, and conferences, in comparison to other publishers. Lastly, we have also categorized reviewed papers based on parameters including IoT, QoS parameters, energy efficiency, application-based (healthcare, vehicular network etc.) and papers integrating thrust technology, as shown in Figure 19. A major chunk of our surveyed papers is based on resource-related aspects.

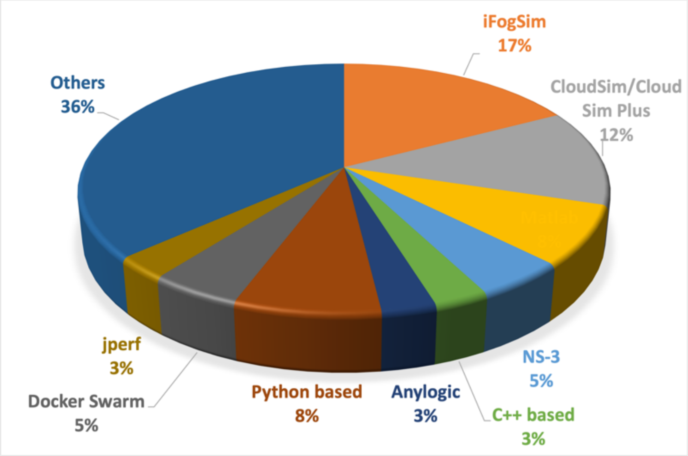

Fig. 17. Comparison of Performance Evaluation Tools for Resource Management in Fog/Edge Computing

VI. Open Issues and Research Directions

The Fog/Edge-Cloud computing model heralds unprecedented expansions in the building of IoT solutions. However, the real implications of this model for multi-layer computing pose a huge number of challenges. The previous section discussed futuristic trends in the form of key-enabler thrust technology, which are making room for their integration with the aforementioned model. In this section, we highlight some long-standing challenges that lay ahead, along with research perspectives, which are discussed as follows:

A. Security and Privacy preservation in Public FNs

The IoT-enabled fog/edge framework strives to improve users' experiences and the resilience of services in case of failovers. To achieve the same, retaining the security, authorization, integrity, and confidentiality of the application

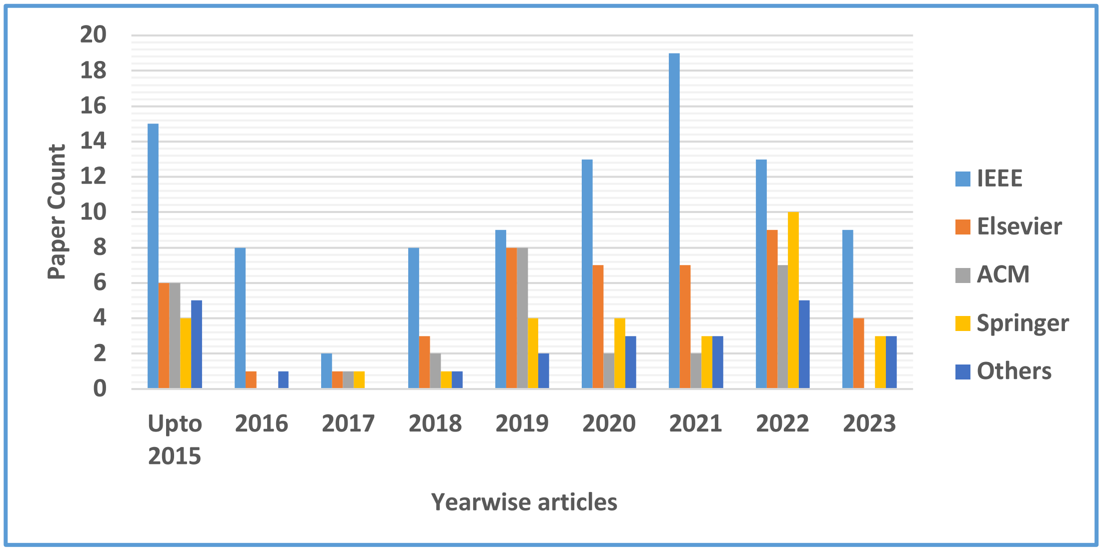

Fig. 18. Bifurcation of Selected publications based on different Publishers

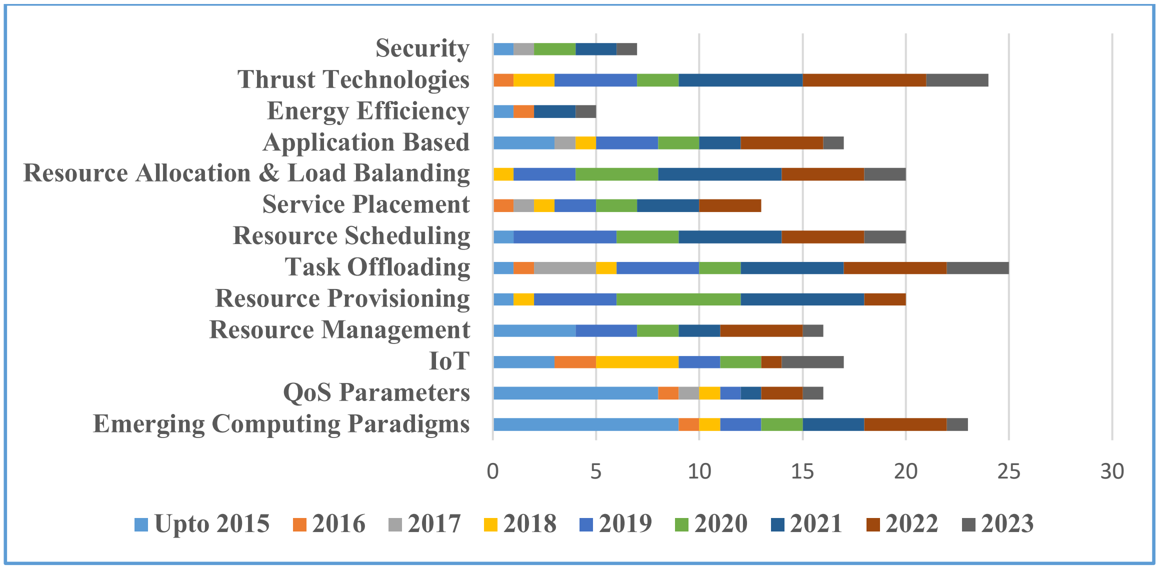

Fig. 19. Categorization of Publications based on Resource Oriented Factors relating Fog/Edge Computing Paradigms

along with the underlying network becomes the prime factor that must be addressed. The IoT-sensing data is exposed to various risks, such as unauthorized access, the risk of intrusion and a wide range of security attacks [174] [175]. The authors had discussed a wide umbrella of attacks under the Network layer of the IoT architecture, which includes DoS, Spoofing, sinkholes, wormholes, man-in-the-middle attacks, and Sybil attacks. As most IoT devices are connected via wireless communication links, most of the security challenges in IoT are related to the wireless network [209]. Also, ensuring the secure and successful execution of applications on resource-constrained edge devices necessitates strong and robust light-weight encryption methods, security mechanisms and advanced and efficient cryptographic schemes [210].

B. Real-time Analytics for Smart Applications

Nowadays, many researchers are working towards creating efficient algorithms to train ML models over the network edge. For the same reason, sharing and communication of computational results are important for the deployment of this distributed computing paradigm. New computation-aware network models are gaining insight for developing distributed data-sharing systems. For instance, in smart cities, the deployment of large-scale sensing networks, anomalous and hazardous event identification, and enabling real time responses are pre-requisites. The fog computing paradigm must be equipped with learning algorithms to analyze real-time data for smart pipeline monitoring to timely detect any event threatening pipeline safety [211]. This opens avenues for researchers to incorporate learning algorithms in the real-time application area, as the majority of the state-of-the-art covers only theoretical aspects, where real implementation is still missing.

C. Self-Adaptive Scheduling in FNs

Most of the scheduling algorithms lack the learning capability of self-adaptiveness, which makes resource scheduling a challenging task in Fog/Edge nodes [221]. Although research is trending towards the deployment of self-adaptive scheduling, all these works are considered at the simulation level only [222]. Therefore, there is a dire need for resource scheduling techniques to equip themselves with the capability to generate optimal task schedules in a dynamic workload environment.

D. Management of geographically distributed resources

Cloud virtualization solution would not be completely suitable for fog/edge. The varied hardware and OS configurations of Fog architecture call for the need for infrastructure virtualization. It calls for Container Orchestration tools, which ease out the scaling up and down of fog infrastructure nodes, henceforth meeting the requirements of real-time IoT applications and the constraints imposed on FNs [212]. Some of the prominent Container Orchestration tools include Kubernetes, Docker Swarm, and Apache Mesos-Marathon.

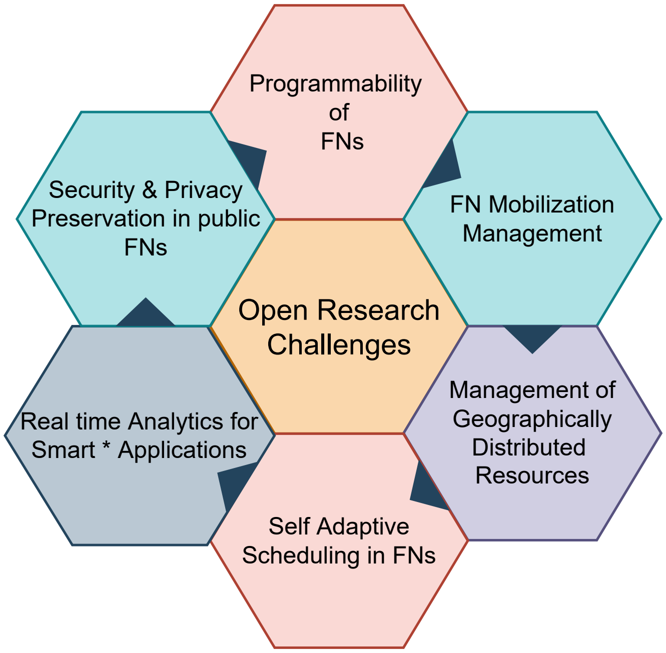
Programmability of FNs
Security & Privacy Preservation in public FNs
FN Mobilization Management
Open Research Challenges
Real time Analytics for Smart * Applications
Management of Geographically Distributed Resources
Self Adaptive Scheduling in FNs

Fig. 20. Open Research Challenges in Fog/Edge Computing

E. FN Mobilization Management

In a VFC environment, Unmanned Aerial Vehicles (UAVs), smartwatches and phones are configured to fully utilize the computational resources of mobile vehicles. However, the management of resources in such a scenario becomes challenging in terms of task offloading due to the varying distance between the FN and the end user [221]. In contrast to static FNs, the decision about task offloading and service placement is not straightforward in the case of mobile FNs. It occurs due to the short and intermittent connection between service provider and user, as well as because the multi-hop forwarding of tasks amongst vehicles is time-consuming and susceptible to packet loss [213]. This ultimately raises a new challenge for task offloading in VFC.

F. Programmability of FNs

The cloud computing paradigm allows users to deploy their code with zero or no knowledge of the underlying platform where the code is being executed. However, the situation is different in Fog/Edge computing, which comprises heterogeneous FNs. The programmer faces huge difficulty in writing an application to run on the fog platform because of the varying runtime environments of FNs. Hence, it has been observed that few works have addressed the issue of the programmability of edge computing. One of the works by Shi et al. [214] proposed the concept of a computing stream: a software-defined computing flow of data, where computing can occur anywhere along with a propagation path in a distributed manner. It ensures data computation in closest proximity to the data source, which optimizes the energy, cost, Total Cost of Ownership (TCO), and latency parameters of applications [214]. All of these challenges have been as depicted in Figure 20.

Proposed Solutions

The implementation and management of smart applications thrive on real-time-based solutions, which can be accomplished by exploiting cutting-edge technologies, including the prediction and decision-making capabilities of AI. The fast-networking capabilities of 5G, the enhanced the security of blockchain etc. The emergence of 5G boosted a broad spectrum of unprecedented applications, along with enhanced mobile broadband with ultra-reliability, improved data rates, low latency, massive device connectivity, and support for a diverse range of IoT and mobile applications. The approaching era of hyper-connectivity with 5G as the fundamental pillar enabling the new AIoT economy, that will eventually bring more intelligence into operational work. Hence, catering to smart manufacturing, e-healthcare, smart campuses, smart stadiums, and smart businesses. 5G supports various types of communications services; ranging from high-speed LAN, WAN guarantee data security, real-time operations, and wide devices, connectivity, thereby enabling effective communication amongst geographically distributed resources in a collaborative environment [215].

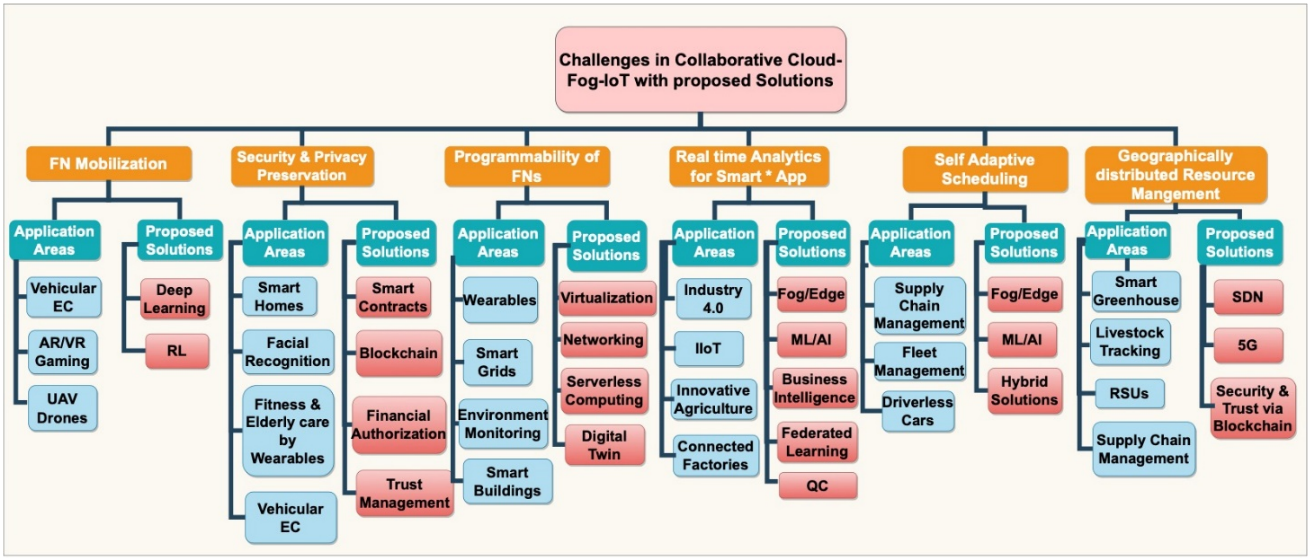

Fig. 21. A Taxonomy of Proposed Solutions for Challenges in a Collaborative Cloud-Fog-IoT Environment

In addition, the computational distribution of the incoming IoT workload by MEC can be boosted by utilizing beyond 5G networks [216]. Figure 21 illustrates the taxonomical representation of proposed solutions corresponding to challenges in a collaborative cloud-fog-IoT environment.

Apart from this, a wide spectrum of critical applications, such as Autonomous vehicles, video surveillance, and AR/VR gaming possess challenges such as mobility amongst vehicles, UAVs, which cannot be handled by a single technique. Hence, hybrid approaches are needed to address these issues. For instance, the drawbacks of existing deep learning techniques, such as slow learning rates, large training data requirements and slow adaptability to dynamic IoT environments have been addressed by proposing a hybridized approach incorporating meta-learning capabilities into existing DRL solutions [217]. This approach improvises the resource management model to learn faster and quickly adapt to rapidly changing environments. Meta learning is characterized by "*learning to learn*" and possessing the capability to adapt quickly, requiring only a few training examples [218].

Despite the significant benefits of collaborative cloud-fog computing frameworks, the data produced by IoT-enabled applications remains a prime target for attackers, presenting potential privacy and security risks. Therefore, safeguarding the security of the extensive data generated at the IoT layer is of utmost importance, necessitating the integration of suitable security measures. Hence, the latest work proposes incorporating blockchain technology as a potential solution. For illustration, consider the healthcare domain, which contains patients' data being streamed from various sensors, smart watches etc. Such sensitive information cannot be sent to the cloud [219]. Consider a heart patient who needs continuous monitoring of their heart rate for the potential risk of a heart attack. In such a situation, the patients and medical staff are registered, followed by the blockchain monitoring agent granting access to registered devices. Then, smart contracts validated by protocols are responsible for proofing a transaction, which, if authenticated, is added as a block to a blockchain-enabled fog server [162]. Nevertheless, the authors envisions addressing the issue of workload prediction, which can facilitate better management of resources by providing a better knowledge of fluctuating incoming workloads [145].

VII. Conclusion

The last decade has witnessed a massive drift in the form of emerging computing paradigms with the widespread prominence of IoT devices. A collaborative cloud-fog/edge paradigm is becoming immensely popular because of its capacity to facilitate real-time IoT applications, ensuring low latency and instant responsiveness. But management of underlying resources becomes more complex and demanding because of its large-scale geographical distribution, heterogeneous and resource-constrained nature, and, above all, the workload is too divergent in fog/edge computational nodes. Hence, our work efficiently presents a comprehensive literature review covering last the 6 year (2018-2023) which includes all aspects of resource management covering existing work to date on AI, non-AI based solutions and hybrid approaches to effectively manage resources in collaborative cloud-fog/edge-IoT environment. Throughout, 490+ articles were collected, out of which 223 have been shortlisted after extensive selection. The articles highlight the existing state-of-the-art work carried out in the resource management domain. Our study outlines various AI-based techniques under a wide umbrella of resource management which covers the provision of computing resources, offloading IoT-based tasks to the cloud, resource scheduling, placement of incoming IoT services, allocating resources and load balancing. Lots of significant efforts have been made to utilize advanced AI-empowered techniques such as metaheuristics, ANNs, Cognitive learning and DRL-based algorithms for optimizing QoS parameters. Moreover, the authors have formulated the mathematical model for each aspect of resource management with the mentioned objective functions like latency, service costs and energy consumption. We have efficiently recognized different challenges which arise at each and every phase of resource management. Our paper highlights the social and ethical impacts of the implications of AI in IoT-driven application areas. We also observed that the future research perspective resides in integrating thrust technology such as Serverless computing, 5G, IIoT, SDN and Federated Learning with edge/fog computing. This comprehensive review will be useful for practitioners, researchers, and academicians in digging into thrust technologies and how its potential key features can be exploited by integrating it with fog/edge. The work endows different proposed solutions in taxonomical form by illustrating IoT-based applications and correspondingly incorporating AI/thrust technology to overcome various challenges. The efforts expended on categorization within the field of resource management will assist researchers in recognizing and choosing the most suitable AI-based techniques for effective resource management in dynamic settings. The future perspective of our work aims to explore and analyze the capabilities and potential of Explainable-AI (XAI) to address the intricacies of existing AI techniques in real-world IoT applications.

ACKNOWLEDGMENTS

We would like to thank the Editor-in-Chief, area editor and anonymous reviewers for their valuable comments, useful suggestions to improve the quality of the paper.

REFERENCES

[1] A. Al-Shafei, H. Zareipour, and Y. Cao, "A Review of High-Performance Computing and Parallel Techniques Applied to Power Systems Optimization," Jul. 2022, Accessed: Oct. 23, 2023. [Online]. Available: https://arxiv.org/abs/2207.02388v1

[2] S. Zhang *et al.*, "State of the Art: High-Performance and High-Throughput Computing for Remote Sensing Big Data," *IEEE Geosci Remote Sens Mag*, vol. 10, no. 4, pp. 125–149, Dec. 2022, doi: 10.1109/MGRS.2022.3204590.

[3] K. Fizza *et al.*, "A Survey on Evaluating the Quality of Autonomic Internet of Things Applications," *IEEE Communications Surveys and Tutorials*, 2022, doi: 10.1109/COMST.2022.3205377.

[4] M. Chernyshev, Z. Baig, O. Bello, and S. Zeadally, "Internet of things (IoT): Research, simulators, and testbeds," *IEEE Internet Things J*, vol. 5, no. 3, pp. 1637–1647, Jun. 2018, doi: 10.1109/JIOT.2017.2786639.

[5] Y. K. Teoh, S. S. Gill, and A. K. Parlikad, "IoT and Fog-Computing-Based Predictive Maintenance Model for Effective Asset Management in Industry 4.0 Using Machine Learning," *IEEE Internet Things J*, vol. 10, no. 3, pp. 2087–2094, Feb. 2023, doi: 10.1109/JIOT.2021.3050441.

[6] W. M. Kang, S. Y. Moon, and J. H. Park, "An enhanced security framework for home appliances in smart home," *Human-centric Computing and Information Sciences*, vol. 7, no. 1, pp. 1–12, Dec. 2017, doi: 10.1186/S13673-017-0087-4/TABLES/5.

[7] M. Pham and K. Xiong, "A survey on security attacks and defense techniques for connected and autonomous vehicles," *Comput Secur*, vol. 109, p. 102269, Oct. 2021, doi: 10.1016/J.COSE.2021.102269.

[8] W. Mao, Z. Zhao, Z. Chang, G. Min, and W. Gao, "Energy-Efficient Industrial Internet of Things: Overview and Open Issues," *IEEE Trans Industr Inform*, vol. 17, no. 11, pp. 7225–7237, Nov. 2021, doi: 10.1109/TII.2021.3067026.

[9] Y. Li, X. Cheng, Y. Cao, D. Wang, and L. Yang, "Smart choice for the smart grid: Narrowband internet of things (NB-IoT)," *IEEE Internet Things J*, vol. 5, no. 3, pp. 1505–1515, Jun. 2018, doi: 10.1109/JIOT.2017.2781251.

[10] H. Wu, K. Wolter, P. Jiao, Y. Deng, Y. Zhao, and M. Xu, "EEDTO: An Energy-Efficient Dynamic Task Offloading Algorithm for Blockchain-Enabled IoT-Edge-Cloud Orchestrated Computing," *IEEE Internet Things J*, vol. 8, no. 4, pp. 2163–2176, Feb. 2021, doi: 10.1109/JIOT.2020.3033521.

[11] C. Jeong and H. Son, "Cooperative Transmission of Energy-Constrained IoT Devices in Wireless-Powered Communication Networks," *IEEE Internet Things J*, vol. 8, no. 5, pp. 3972–3982, Mar. 2021, doi: 10.1109/JIOT.2020.3027101.

[12] A. H. Sodhro *et al.*, "Quality of Service Optimization in an IoT-Driven Intelligent Transportation System," *IEEE Wirel Commun*, vol. 26, no. 6, pp. 10–17, Dec. 2019, doi: 10.1109/MWC.001.1900085.

[13] A. Yousefpour *et al.*, "All one needs to know about fog computing and related edge computing paradigms: A complete survey," *Journal of Systems Architecture*, vol. 98, pp. 289–330, Sep. 2019, doi: 10.1016/J.SYSARC.2019.02.009.

[14] F. Javed, M. K. Afzal, M. Sharif, and B. S. Kim, "Internet of Things (IoT) operating systems support, networking technologies, applications, and challenges: A comparative review," *IEEE Communications Surveys and Tutorials*, vol. 20, no. 3, pp. 2062–2100, Jul. 2018, doi: 10.1109/COMST.2018.2817685.

[15] M. D. Assunção, R. N. Calheiros, S. Bianchi, M. A. S. Netto, and R. Buyya, "Big Data computing and clouds: Trends and future directions," *J Parallel Distrib Comput*, vol. 79–80, pp. 3–15, May 2015, doi: 10.1016/J.JPDC.2014.08.003.

[16] M. Armbrust *et al.*, "A view of cloud computing," *Commun ACM*, vol. 53, no. 4, pp. 50–58, Apr. 2010, doi: 10.1145/1721654.1721672.

[17] R. Buyya, C. S. Yeo, and S. Venugopal, "Market-Oriented Cloud Computing: Vision, Hype, and Reality for Delivering IT Services as Computing Utilities," in *2008 10th IEEE International Conference on High Performance Computing and Communications*, IEEE, Sep. 2008, pp. 5–13. doi: 10.1109/HPCC.2008.172.

[18] F. Alhaddadin, W. Liu, and J. A. Gutierrez, "A User Profile-Aware Policy-Based Management Framework for Greening the Cloud," in *2014 IEEE Fourth International Conference on Big Data and Cloud Computing*, IEEE, Dec. 2014, pp. 682–687. doi: 10.1109/BDCloud.2014.116.

[19] M. Aazam, I. Khan, A. A. Alsaffar, and E. N. Huh, "Cloud of Things: Integrating Internet of Things and cloud computing and the issues involved," *Proceedings of 2014 11th International Bhurban Conference on Applied Sciences and Technology, IBCAST 2014*, pp. 414–419, 2014, doi: 10.1109/IBCAST.2014.6778179.

[20] M. Aazam, P. P. Hung, and E. N. Huh, "Smart gateway based communication for cloud of things," *IEEE ISSNIP 2014 - 2014 IEEE 9th International Conference on Intelligent Sensors, Sensor Networks and Information Processing, Conference Proceedings*, 2014, doi: 10.1109/ISSNIP.2014.6827673.

[21] Lu. Yan, "The Internet of things : from RFID to the next-generation pervasive networked systems," p. 318, 2008, Accessed: Oct. 20, 2022. [Online]. Available: https://books.google.com/books/about/The_Internet_of_Things.html?id=_ZS_g_IHhD0C

[22] C. H. Hsu, K. D. Slagter, S. C. Chen, and Y. C. Chung, "Optimizing Energy Consumption with Task Consolidation in Clouds," *Inf Sci (N Y)*, vol. 258, pp. 452–462, Feb. 2014, doi: 10.1016/J.INS.2012.10.041.

[23] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," *IEEE Communications Surveys and Tutorials*, vol. 17, no. 4, pp. 2347–2376, Oct. 2015, doi: 10.1109/COMST.2015.2444095.

[24] C. Pahl, "Containerization and the PaaS Cloud," *IEEE Cloud Computing*, vol. 2, no. 3, pp. 24–31, May 2015, doi: 10.1109/MCC.2015.51.

[25] S. Kadri, A. Sboner, A. Sigaras, and S. Roy, “Containers in Bioinformatics: Applications, Practical Considerations, and Best Practices in Molecular Pathology,” *The Journal of Molecular Diagnostics*, vol. 24, no. 5, pp. 442–454, May 2022, doi: 10.1016/J.JMOLDX.2022.01.006.

[26] Q. Zhang, L. Liu, C. Pu, Q. Dou, L. Wu, and W. Zhou, “A Comparative Study of Containers and Virtual Machines in Big Data Environment,” *IEEE International Conference on Cloud Computing, CLOUD*, vol. 2018-July, pp. 178–185, Sep. 2018, doi: 10.1109/CLOUD.2018.00030.

[27] Nisha Angeline C. V. and R. Lavanya, “Fog Computing and Its Role in the Internet of Things,” in *https://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/978-1-5225-7149-0.ch003*, IGI Global, 2019, pp. 63–71. doi: 10.4018/978-1-5225-7149-0.ch003.

[28] C. Perera, Y. Qin, J. C. Estrella, S. Reiff-Marganiec, and A. V. Vasilakos, “Fog Computing for Sustainable Smart Cities,” *ACM Computing Surveys (CSUR)*, vol. 50, no. 3, Jun. 2017, doi: 10.1145/3057266.

[29] B. Jennings and R. Stadler, “Resource Management in Clouds: Survey and Research Challenges,” *Journal of Network and Systems Management 2014 23:3*, vol. 23, no. 3, pp. 567–619, Mar. 2014, doi: 10.1007/S10922-014-9307-7.

[30] K. Chakrabarti, “Deep learning based offloading for mobile augmented reality application in 6G,” *Computers and Electrical Engineering*, vol. 95, p. 107381, Oct. 2021, doi: 10.1016/J.COMPELECENG.2021.107381.

[31] R. Mahmud, S. N. Srirama, K. Ramamohanarao, and R. Buyya, “Quality of Experience (QoE)-aware placement of applications in Fog computing environments,” *J Parallel Distrib Comput*, vol. 132, pp. 190–203, Oct. 2019, doi: 10.1016/J.JPDC.2018.03.004.

[32] 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, Oct. 2016, doi: 10.1109/JIOT.2016.2579198.

[33] “(1) (PDF) Above the Clouds: A Berkeley View of Cloud Computing.” Accessed: Oct. 20, 2022. [Online]. Available: https://www.researchgate.net/publication/200045935_Above_the_Clouds_A_Berkeley_View_of_Cloud_Computing

[34] D. Grewe, M. Wagner, M. Arumaithurai, D. Kutscher, and I. Psaras, “Information-Centric Mobile Edge Computing for Connected Vehicle Environments: Challenges and Research Directions,” *Proceedings of the Workshop on Mobile Edge Communications*, vol. 6, 2017, doi: 10.1145/3098208.

[35] B. D. Noble, M. Satyanarayanan, D. Narayanan, J. Eric Tilton, J. Flinn, and K. R. Walker, “Agile Application-Aware Adaptation for Mobility,” *Proceedings of the sixteenth ACM symposium on Operating systems principles - SOSP ’97*, doi: 10.1145/268998.

[36] P. Patil, A. Hakiri, and A. Gokhale, “Cyber Foraging and Offloading Framework for Internet of Things,” *Proceedings - International Computer Software and Applications Conference*, vol. 1, pp. 359–368, Aug. 2016, doi: 10.1109/COMPSAC.2016.88.

[37] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, “Fog computing and its role in the internet of things,” *MCC’12 - Proceedings of the 1st ACM Mobile Cloud Computing Workshop*, pp. 13–15, 2012, doi: 10.1145/2342509.2342513.

[38] X. Zhao and C. Huang, “Microservice Based Computational Offloading Framework and Cost Efficient Task Scheduling Algorithm in Heterogeneous Fog Cloud Network,” *IEEE Access*, vol. 8, pp. 56680–56694, 2020, doi: 10.1109/ACCESS.2020.2981860.

[39] M. Mukherjee, V. Kumar, Q. Zhang, C. X. Mavromoustakis, and R. Matam, “Optimal Pricing for Offloaded Hard- and Soft-Deadline Tasks in Edge Computing,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 9829–9839, Jul. 2022, doi: 10.1109/TITS.2021.3117973.

[40] P. Habibi, M. Farhoudi, S. Kazemian, S. Khorsandi, and A. Leon-Garcia, “Fog Computing: A Comprehensive Architectural Survey,” *IEEE Access*, vol. 8, pp. 69105–69133, 2020, doi: 10.1109/ACCESS.2020.2983253.

[41] M. S. Aslanpour *et al.*, “Serverless Edge Computing: Vision and Challenges,” *ACM International Conference Proceeding Series*, 2021, doi: 10.1145/3437378.3444367.

[42] S. S. Gill *et al.*, “AI for next generation computing: Emerging trends and future directions,” *Internet of Things*, vol. 19, p. 100514, Aug. 2022, doi: 10.1016/J.IOT.2022.100514.

[43] M. Ghobaei-Arani, A. Souri, and A. A. Rahmanian, “Resource Management Approaches in Fog Computing: a Comprehensive Review,” *Journal of Grid Computing 2019 18:1*, vol. 18, no. 1, pp. 1–42, Sep. 2019, doi: 10.1007/S10723-019-09491-1.

[44] K. H. Abdulkareem *et al.*, “A Review of Fog Computing and Machine Learning: Concepts, Applications, Challenges, and Open Issues,” *IEEE Access*, vol. 7, pp. 153123–153140, 2019, doi: 10.1109/ACCESS.2019.2947542.

[45] C. H. Hong and B. Varghese, “Resource Management in Fog/Edge Computing,” *ACM Computing Surveys (CSUR)*, vol. 52, no. 5, Sep. 2019, doi: 10.1145/3326066.

[46] S. Deng, H. Zhao, W. Fang, J. Yin, S. Dustdar, and A. Y. Zomaya, “Edge Intelligence: The Confluence of Edge Computing and Artificial Intelligence,” *IEEE Internet Things J*, vol. 7, no. 8, pp. 7457–7469, Aug. 2020, doi: 10.1109/JIOT.2020.2984887.

[47] I. Martinez, A. S. Hafid, and A. Jarray, “Design, Resource Management, and Evaluation of Fog Computing Systems: A Survey,” *IEEE Internet Things J*, vol. 8, no. 4, pp. 2494–2516, Feb. 2021, doi: 10.1109/JIOT.2020.3022699.

[48] Z. M. Nayeri, T. Ghafarian, and B. Javadi, “Application placement in Fog computing with AI approach: Taxonomy and a state of the art survey,” *Journal of Network and Computer Applications*, vol. 185, p. 103078, Jul. 2021, doi: 10.1016/J.JNCA.2021.103078.

[49] A. Shakarami, H. Shakarami, M. Ghobaei-Arani, E. Nikougoftar, and M. Faraji-Mehmandar, “Resource provisioning in edge/fog computing: A Comprehensive and Systematic Review,” *Journal of Systems Architecture*, vol. 122, p. 102362, Jan. 2022, doi: 10.1016/J.SYSARC.2021.102362.

[50] JamilBushra, IjazHumaira, ShojafarMohammad, MunirKashif, and BuyyaRajkumar, “Resource Allocation and Task Scheduling in Fog Computing and Internet of Everything Environments: A Taxonomy, Review, and Future Directions,” *ACM Computing Surveys (CSUR)*, vol. 54, no. 11s, pp. 1–38, Sep. 2022, doi: 10.1145/3513002.

[51] J. Zhang and D. Tao, “Empowering Things with Intelligence: A Survey of the Progress, Challenges, and Opportunities in Artificial Intelligence of Things,” *IEEE Internet Things J*, vol. 8, no. 10, pp. 7789–7817, Nov. 2020, doi: 10.1109/JIOT.2020.3039359.

[52] M. Lin and Y. Zhao, “Artificial intelligence-empowered resource management for future wireless communications: A survey,” *China Communications*, vol. 17, no. 3, pp. 58–77, Mar. 2020, doi: 10.23919/JCC.2020.03.006.

[53] S. Tuli *et al.*, “AI Augmented Edge and Fog Computing: Trends and Challenges,” Aug. 2022, doi: 10.1016/j.jnca.2023.103648.

[54] W. Su, L. Li, F. Liu, M. He, and X. Liang, “AI on the edge: a comprehensive review,” *Artif Intell Rev*, vol. 55, no. 8, pp. 6125–6183, Dec. 2022, doi: 10.1007/S10462-022-10141-4.

[55] Z. Chang, S. Liu, X. Xiong, Z. Cai, and G. Tu, “A Survey of Recent Advances in Edge-Computing-Powered Artificial Intelligence of Things,” *IEEE Internet Things J*, vol. 8, no. 18, pp. 13849–13875, Sep. 2021, doi: 10.1109/JIOT.2021.3088875.

[56] Z. Chang, S. Liu, X. Xiong, Z. Cai, and G. Tu, “A Survey of Recent Advances in Edge-Computing-Powered Artificial Intelligence of Things,” *IEEE Internet Things J*, vol. 8, no. 18, pp. 13849–13875, Sep. 2021, doi: 10.1109/JIOT.2021.3088875.

[57] S. Iftikhar *et al.*, “AI-based fog and edge computing: A systematic review, taxonomy and future directions,” *Internet of Things*, vol. 21, p. 100674, Apr. 2023, doi: 10.1016/J.IOT.2022.100674.

[58] E. Manavalan and K. Jayakrishna, “A review of Internet of Things (IoT) embedded sustainable supply chain for industry 4.0 requirements,” *Comput Ind Eng*, vol. 127, pp. 925–953, Jan. 2019, doi: 10.1016/J.CIE.2018.11.030.

[59] F. Allhoff and A. Henschke, “The Internet of Things: Foundational ethical issues,” *Internet of Things*, vol. 1–2, pp. 55–66, Sep. 2018, doi: 10.1016/J.IOT.2018.08.005.

[60] Y. Agarwal and A. K. Dey, “Toward Building a Safe, Secure, and Easy-to-Use Internet of Things Infrastructure,” *Computer (Long Beach Calif)*, vol. 49, no. 4, pp. 88–91, Apr. 2016, doi: 10.1109/MC.2016.111.

[61] A. George, C. Allen, and W. Wallach, “Permalink Téléchargé de Scholars Portal Books sur 2020-02-11,” *Robot Ethics: The ethical and social implications of robotics*, pp. 2–8, 2012.

[62] S. U. Amin, M. S. Hossain, G. Muhammad, M. Alhussein, and M. A. Rahman, “Cognitive Smart Healthcare for Pathology Detection and Monitoring,” *IEEE Access*, vol. 7, pp. 10745–10753, 2019, doi: 10.1109/ACCESS.2019.2891390.

[63] Q. Wang, Y. Guo, L. Yu, and P. Li, "Earthquake Prediction Based on Spatio-Temporal Data Mining: An LSTM Network Approach," *IEEE Trans Emerg Top Comput*, vol. 8, no. 1, pp. 148–158, Jan. 2020, doi: 10.1109/TETC.2017.2699169.

[64] F. Liang, W. Yu, X. Liu, D. Griffith, and N. Golmie, "Toward Edge-Based Deep Learning in Industrial Internet of Things," *IEEE Internet Things J*, vol. 7, no. 5, pp. 4329–4341, May 2020, doi: 10.1109/JIOT.2019.2963635.

[65] R. M. Singh, L. K. Awasthi, and G. Sikka, "Towards Metaheuristic Scheduling Techniques in Cloud and Fog: An Extensive Taxonomic Review," *ACM Computing Surveys (CSUR)*, vol. 55, no. 3, pp. 1–43, Feb. 2022, doi: 10.1145/3494520.

[66] B. Jennings and R. Stadler, "Resource Management in Clouds: Survey and Research Challenges," *Journal of Network and Systems Management 2014 23:3*, vol. 23, no. 3, pp. 567–619, 2014, doi: 10.1007/S10922-014-9307-7.

[67] B. Jennings and R. Stadler, "Resource Management in Clouds: Survey and Research Challenges," *Journal of Network and Systems Management 2014 23:3*, vol. 23, no. 3, pp. 567–619, Mar. 2014, doi: 10.1007/S10922-014-9307-7.

[68] R. Chard, K. Chard, K. Bubendorfer, L. Lacinski, R. Madduri, and I. Foster, "Cost-Aware Elastic Cloud Provisioning for Scientific Workloads," *Proceedings - 2015 IEEE 8th International Conference on Cloud Computing, CLOUD 2015*, pp. 971–974, 2015, doi: 10.1109/CLOUD.2015.130.

[69] R. Chard, K. Chard, K. Bubendorfer, L. Lacinski, R. Madduri, and I. Foster, "Cost-Aware Elastic Cloud Provisioning for Scientific Workloads," *Proceedings - 2015 IEEE 8th International Conference on Cloud Computing, CLOUD 2015*, pp. 971–974, Aug. 2015, doi: 10.1109/CLOUD.2015.130.

[70] D. Roca, J. V Quiroga, M. Valero, and M. Nemirovsky, "Fog Function Virtualization: A flexible solution for IoT applications," *2017 2nd International Conference on Fog and Mobile Edge Computing, FMEC 2017*, pp. 74–80, 2017, doi: 10.1109/FMEC.2017.7946411.

[71] M. Aazam, S. Zeadally, and K. A. Harras, "Offloading in fog computing for IoT: Review, enabling technologies, and research opportunities," *Future Generation Computer Systems*, vol. 87, pp. 278–289, Oct. 2018, doi: 10.1016/J.FUTURE.2018.04.057.

[72] M. Taneja and A. Davy, "Resource Aware Placement of Data Analytics Platform in Fog Computing," *Procedia Comput Sci*, vol. 97, pp. 153–156, Jan. 2016, doi: 10.1016/J.PROCS.2016.08.295.

[73] L. Pioli, C. F. Dorneles, D. D. J. de Macedo, and M. A. R. Dantas, "An overview of data reduction solutions at the edge of IoT systems: a systematic mapping of the literature," *Computing*, vol. 104, no. 8, pp. 1867–1889, Aug. 2022, doi: 10.1007/S00607-022-01073-6/FIGURES/10.

[74] M. Saqlain, M. Piao, Y. Shim, and J. Y. Lee, "Framework of an IoT-based Industrial Data Management for Smart Manufacturing," *Journal of Sensor and Actuator Networks 2019, Vol. 8, Page 25*, vol. 8, no. 2, p. 25, Apr. 2019, doi: 10.3390/JSAN8020025.

[75] Y. Yan, Y. Qian, H. Sharif, and D. Tipper, "A survey on cyber security for smart grid communications," *IEEE Communications Surveys and Tutorials*, vol. 14, no. 4, pp. 998–1010, 2012, doi: 10.1109/SURV.2012.010912.00035.

[76] K. Tange, M. De Donno, X. Fafoutis, and N. Dragoni, "A Systematic Survey of Industrial Internet of Things Security: Requirements and Fog Computing Opportunities," *IEEE Communications Surveys and Tutorials*, vol. 22, no. 4, pp. 2489–2520, Oct. 2020, doi: 10.1109/COMST.2020.3011208.

[77] N. Moustafa, N. Koroniotis, M. Keshk, A. Y. Zomaya, and Z. Tari, "Explainable Intrusion Detection for Cyber Defences in the Internet of Things: Opportunities and Solutions," *IEEE Communications Surveys and Tutorials*, vol. 25, no. 3, pp. 1775–1807, 2023, doi: 10.1109/COMST.2023.3280465.

[78] F. Hussain, R. Hussain, S. A. Hassan, and E. Hossain, "Machine Learning in IoT Security: Current Solutions and Future Challenges," *IEEE Communications Surveys and Tutorials*, vol. 22, no. 3, pp. 1686–1721, Jul. 2020, doi: 10.1109/COMST.2020.2986444.

[79] A. Shakarami, H. Shakarami, M. Ghobaei-Arani, E. Nikougoftar, and M. Faraji-Mehmandar, "Resource provisioning in edge/fog computing: A Comprehensive and Systematic Review," *Journal of Systems Architecture*, vol. 122, p. 102362, Jan. 2022, doi: 10.1016/J.SYSARC.2021.102362.

[80] M. Fahimullah, S. Ahvar, and M. Trocan, "A Review of Resource Management in Fog Computing: Machine Learning Perspective," Sep. 2022, Accessed: Jul. 03, 2023. [Online]. Available: https://arxiv.org/abs/2209.03066v1

[81] R. O. Aburukba, T. Landolsi, and D. Omer, "A heuristic scheduling approach for fog-cloud computing environment with stationary IoT devices," *Journal of Network and Computer Applications*, vol. 180, p. 102994, Apr. 2021, doi: 10.1016/J.JNCA.2021.102994.

[82] M. Ghobaei-Arani, "A workload clustering based resource provisioning mechanism using Biogeography based optimization technique in the cloud based systems," *Soft comput*, vol. 25, no. 5, pp. 3813–3830, Mar. 2021, doi: 10.1007/S00500-020-05409-2/FIGURES/11.

[83] M. Abdullah, W. Iqbal, A. Mahmood, F. Bukhari, and A. Erradi, "Predictive Autoscaling of Microservices Hosted in Fog Microdata Center," *IEEE Syst J*, vol. 15, no. 1, pp. 1275–1286, 2021, doi: 10.1109/JSYST.2020.2997518.

[84] R. Naha, S. Garg, S. K. Battula, M. B. Amin, and D. Georgakopoulos, "Multiple linear regression-based energy-aware resource allocation in the Fog computing environment," *Computer Networks*, vol. 216, p. 109240, 2022, doi: 10.1016/J.COMNET.2022.109240.

[85] G. Li, Y. Yao, J. Wu, X. Liu, X. Sheng, and Q. Lin, "A new load balancing strategy by task allocation in edge computing based on intermediary nodes," *EURASIP J Wirel Commun Netw*, vol. 2020, no. 1, pp. 1–10, 2020, doi: 10.1186/S13638-019-1624-9/FIGURES/5.

[86] S. Dehnavi, H. R. Faragardi, M. Kargahi, and T. Fahringer, "A reliability-aware resource provisioning scheme for real-time industrial applications in a Fog-integrated smart factory," *Microprocess Microsyst*, vol. 70, pp. 1–14, Oct. 2019, doi: 10.1016/J.MICPRO.2019.05.011.

[87] V. Jain and B. Kumar, "QoS-Aware Task Offloading in Fog Environment Using Multi-agent Deep Reinforcement Learning," *Journal of Network and Systems Management*, vol. 31, no. 1, pp. 1–32, Mar. 2023, doi: 10.1007/S10922-022-09696-Y/FIGURES/13.

[88] M. Kumar, S. C. Sharma, A. Goel, and S. P. Singh, "A comprehensive survey for scheduling techniques in cloud computing," *Journal of Network and Computer Applications*, vol. 143, pp. 1–33, Oct. 2019, doi: 10.1016/J.JNCA.2019.06.006.

[89] Q. T. Nguyen, N. Quang-Hung, N. H. Tuong, V. H. Tran, and N. Thoai, "Virtual machine allocation in cloud computing for minimizing total execution time on each machine," *2013 International Conference on Computing, Management and Telecommunications, ComManTel 2013*, pp. 241–245, 2013, doi: 10.1109/COMMANTEL.2013.6482398.

[90] J. Yao and N. Ansari, "Fog Resource Provisioning in Reliability-Aware IoT Networks," *IEEE Internet Things J*, vol. 6, no. 5, pp. 8262–8269, Oct. 2019, doi: 10.1109/JIOT.2019.2922585.

[91] M. Salem, A. Adinoyi, M. Rahman, H. Yanikomeroglu, D. Falconer, and Y. D. Kim, "Fairness-aware radio resource management in downlink OFDMA cellular relay networks," *IEEE Trans Wirel Commun*, vol. 9, no. 5, pp. 1628–1639, May 2010, doi: 10.1109/TWC.2010.05.081548.

[92] H. J. Hong, J. C. Chuang, and C. H. Hsu, "Animation rendering on multimedia fog computing platforms," *Proceedings of the International Conference on Cloud Computing Technology and Science, CloudCom*, vol. 0, pp. 336–343, 2016, doi: 10.1109/CLOUDCOM.2016.0060.

[93] S. Dehnavi, H. R. Faragardi, M. Kargahi, and T. Fahringer, "A reliability-aware resource provisioning scheme for real-time industrial applications in a Fog-integrated smart factory," *Microprocess Microsyst*, vol. 70, pp. 1–14, 2019, doi: 10.1016/J.MICPRO.2019.05.011.

[94] R. Balakrishnan, M. Akdeniz, S. Dhakal, and N. Himayat, "Resource Management and Fairness for Federated Learning over Wireless Edge Networks," *IEEE Workshop on Signal Processing Advances in Wireless Communications, SPAWC*, vol. 2020-May, May 2020, doi: 10.1109/SPAWC48557.2020.9154285.

[95] M. Etemadi, M. Ghobaei-Arani, and A. Shahidinejad, "Resource provisioning for IoT services in the fog computing environment: An autonomic approach," *Comput Commun*, vol. 161, pp. 109–131, Sep. 2020, doi: 10.1016/J.COMCOM.2020.07.028.

[96] M. Etemadi, M. Ghobaei-Arani, and A. Shahidinejad, "A learning-based resource provisioning approach in the fog computing environment," *https://doi.org/10.1080/0952813X.2020.1818294*, vol. 33, no. 6, pp. 1033–1056, 2020, doi: 10.1080/0952813X.2020.1818294.

[97] M. Ghobaei-Arani, "A workload clustering based resource provisioning mechanism using Biogeography based optimization technique in the

cloud based systems," *Soft comput*, vol. 25, no. 5, pp. 3813–3830, Nov. 2020, doi: 10.1007/S00500-020-05409-2.

[98] C. Li, J. Bai, Y. Ge, and Y. Luo, "Heterogeneity-aware elastic provisioning in cloud-assisted edge computing systems," *Future Generation Computer Systems*, vol. 112, pp. 1106–1121, Nov. 2020, doi: 10.1016/J.FUTURE.2020.06.022.

[99] N. Madan, A. W. Malik, A. U. Rahman, and S. D. Ravana, "On-demand resource provisioning for vehicular networks using flying fog," *Vehicular Communications*, vol. 25, p. 100252, Oct. 2020, doi: 10.1016/J.VEHCOM.2020.100252.

[100] H. Sami, H. Otrok, J. Bentahar, and A. Mourad, "AI-Based Resource Provisioning of IoE Services in 6G: A Deep Reinforcement Learning Approach," *IEEE Transactions on Network and Service Management*, vol. 18, no. 3, pp. 3527–3540, Sep. 2021, doi: 10.1109/TNSM.2021.3066625.

[101] D. I. Hatti and A. V. Sutagundar, "Swarm intelligence based MSMOPSO for optimization of resource provisioning in Internet of Things," *Recent Trends in Computational Intelligence Enabled Research: Theoretical Foundations and Applications*, pp. 61–82, Jan. 2021, doi: 10.1016/B978-0-12-822844-9.00028-1.

[102] M. Abdullah, W. Iqbal, A. Mahmood, F. Bukhari, and A. Erradi, "Predictive Autoscaling of Microservices Hosted in Fog Microdata Center," *IEEE Syst J*, vol. 15, no. 1, pp. 1275–1286, Mar. 2021, doi: 10.1109/JSYST.2020.2997518.

[103] E. E. Sham and D. P. Vidyarthi, "Admission control and resource provisioning in fog-integrated cloud using modified fuzzy inference system," *Journal of Supercomputing*, vol. 78, no. 13, pp. 15463–15503, Sep. 2022, doi: 10.1007/S11227-022-04483-7/FIGURES/21.

[104] N. Madan, A. W. Malik, A. U. Rahman, and S. D. Ravana, "On-demand resource provisioning for vehicular networks using flying fog," *Vehicular Communications*, vol. 25, p. 100252, Oct. 2020, doi: 10.1016/J.VEHCOM.2020.100252.

[105] S. Shen, V. Van Beek, and A. Iosup, "Statistical characterization of business-critical workloads hosted in cloud datacenters," *Proceedings - 2015 IEEE/ACM 15th International Symposium on Cluster, Cloud, and Grid Computing, CCGrid 2015*, pp. 465–474, Jul. 2015, doi: 10.1109/CCGRID.2015.60.

[106] M. Adhikari and S. N. Srirama, "Multi-objective accelerated particle swarm optimization with a container-based scheduling for Internet-of-Things in cloud environment," *Journal of Network and Computer Applications*, vol. 137, pp. 35–61, Jul. 2019, doi: 10.1016/J.JNCA.2019.04.003.

[107] S. Vemireddy and R. R. Rout, "Fuzzy Reinforcement Learning for energy efficient task offloading in Vehicular Fog Computing," *Computer Networks*, vol. 199, p. 108463, Nov. 2021, doi: 10.1016/J.COMNET.2021.108463.

[108] M. Khabazian and M. K. M. Ali, "A performance modeling of connectivity in vehicular Ad Hoc networks," *IEEE Trans Veh Technol*, vol. 57, no. 4, pp. 2440–2450, Jul. 2008, doi: 10.1109/TVT.2007.912161.

[109] K. Fizza, N. Auluck, and A. Azim, "Improving the Schedulability of Real-Time Tasks Using Fog Computing," *IEEE Trans Serv Comput*, vol. 15, no. 1, pp. 372–385, 2022, doi: 10.1109/TSC.2019.2944360.

[110] G. Vijayasekaran and M. Duraipandian, "An Efficient Clustering and Deep Learning Based Resource Scheduling for Edge Computing to Integrate Cloud-IoT," *Wirel Pers Commun*, vol. 124, no. 3, pp. 2029–2044, Jun. 2022, doi: 10.1007/S11277-021-09442-8.

[111] M. Goudarzi, M. S. Palaniswami, and R. Buyya, "A Distributed Deep Reinforcement Learning Technique for Application Placement in Edge and Fog Computing Environments," *IEEE Trans Mob Comput*, 2021, doi: 10.1109/TMC.2021.3123165.

[112] H. Arabnejad and J. G. Barbosa, "List scheduling algorithm for heterogeneous systems by an optimistic cost table," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 3, pp. 682–694, Mar. 2014, doi: 10.1109/TPDS.2013.57.

[113] M. Ghobaei-Arani and A. Shahidinejad, "A cost-efficient IoT service placement approach using whale optimization algorithm in fog computing environment," *Expert Syst Appl*, vol. 200, p. 117012, Aug. 2022, doi: 10.1016/J.ESWA.2022.117012.

[114] S. Sthapit, J. Thompson, N. M. Robertson, and J. R. Hopgood, "Computational Load Balancing on the Edge in Absence of Cloud and Fog," *IEEE Trans Mob Comput*, vol. 18, no. 7, pp. 1499–1512, Jul. 2019, doi: 10.1109/TMC.2018.2863301.

[115] F. M. Talaat, "Effective prediction and resource allocation method (EPRAM) in fog computing environment for smart healthcare system," *Multimed Tools Appl*, vol. 81, no. 6, pp. 8235–8258, Mar. 2022, doi: 10.1007/S11042-022-12223-5/FIGURES/13.

[116] R. Naha, S. Garg, S. K. Battula, M. B. Amin, and D. Georgakopoulos, "Multiple linear regression-based energy-aware resource allocation in the Fog computing environment," *Computer Networks*, vol. 216, p. 109240, Oct. 2022, doi: 10.1016/J.COMNET.2022.109240.

[117] A. G. Gad, "Particle Swarm Optimization Algorithm and Its Applications: A Systematic Review," *Archives of Computational Methods in Engineering 2022 29:5*, vol. 29, no. 5, pp. 2531–2561, Apr. 2022, doi: 10.1007/S11831-021-09694-4.

[118] Z. Geng *et al.*, "A model-free Bayesian classifier," *Inf Sci (N Y)*, vol. 482, pp. 171–188, May 2019, doi: 10.1016/J.INS.2019.01.026.

[119] H. Sami, H. Otrok, J. Bentahar, and A. Mourad, "AI-Based Resource Provisioning of IoE Services in 6G: A Deep Reinforcement Learning Approach," *IEEE Transactions on Network and Service Management*, vol. 18, no. 3, pp. 3527–3540, 2021, doi: 10.1109/TNSM.2021.3066625.

[120] R. N. Calheiros, E. Masoumi, R. Ranjan, and R. Buyya, "Workload prediction using ARIMA model and its impact on cloud applications' QoS," *IEEE Transactions on Cloud Computing*, vol. 3, no. 4, pp. 449–458, 2015, doi: 10.1109/TCC.2014.2350475.

[121] H. Ye, L. Yang, and X. Liu, "Optimizing weight and threshold of BP neural network using SFLA: Applications to nonlinear function fitting," *Proceedings - 4th International Conference on Emerging Intelligent Data and Web Technologies, EIDWT 2013*, pp. 211–214, 2013, doi: 10.1109/EIDWT.2013.41.

[122] F. Jalali, K. Hinton, R. Ayre, T. Alpcan, and R. S. Tucker, "Fog computing may help to save energy in cloud computing," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 5, pp. 1728–1739, May 2016, doi: 10.1109/JSAC.2016.2545559.

[123] N. Kumari, A. Yadav, and P. K. Jana, "Task offloading in fog computing: A survey of algorithms and optimization techniques," *Computer Networks*, vol. 214, p. 109137, Sep. 2022, doi: 10.1016/J.COMNET.2022.109137.

[124] H. Xiang, M. Zhang, and C. Jian, "Federated deep reinforcement learning-based online task offloading and resource allocation in harsh mobile edge computing environment," *Cluster Comput*, pp. 1–17, Oct. 2023, doi: 10.1007/S10586-023-04143-2/FIGURES/9.

[125] M. Kumar, G. K. Walia, H. Shingare, S. Singh, and S. S. Gill, "AI-Based Sustainable and Intelligent Offloading Framework for IIoT in Collaborative Cloud-Fog Environments," *IEEE Transactions on Consumer Electronics*, 2023, doi: 10.1109/TCE.2023.3320673.

[126] J. Lin, S. Huang, H. Zhang, X. Yang, and P. Zhao, "A Deep Reinforcement Learning based Computation Offloading with Mobile Vehicles in Vehicular Edge Computing," *IEEE Internet Things J*, 2023, doi: 10.1109/JIOT.2023.3264281.

[127] M. A. Ebrahim, G. A. Ebrahim, H. K. Mohamed, and S. O. Abdellatif, "A Deep Learning Approach for Task Offloading in Multi-UAV Aided Mobile Edge Computing," *IEEE Access*, vol. 10, pp. 101716–101731, 2022, doi: 10.1109/ACCESS.2022.3208584.

[128] F. Saeik *et al.*, "Task offloading in Edge and Cloud Computing: A survey on mathematical, artificial intelligence and control theory solutions," *Computer Networks*, vol. 195, p. 108177, Aug. 2021, doi: 10.1016/J.COMNET.2021.108177.

[129] M. A. Mirza *et al.*, "DRL-assisted delay optimized task offloading in automotive-industry 5.0 based VECNs," *Journal of King Saud University - Computer and Information Sciences*, vol. 35, no. 6, p. 101512, Jun. 2023, doi: 10.1016/J.JKSUCI.2023.02.013.

[130] D. Sha and R. Zhao, "DRL-based task offloading and resource allocation in multi-UAV-MEC network with SDN," *2021 IEEE/CIC International Conference on Communications in China, ICCC 2021*, pp. 595–600, Jul. 2021, doi: 10.1109/ICCC52777.2021.9580253.

[131] J. Shi, J. Du, J. Wang, and J. Yuan, "Deep reinforcement learning-based V2V partial computation offloading in vehicular fog computing," *IEEE Wireless Communications and Networking Conference, WCNC*, vol. 2021-March, 2021, doi: 10.1109/WCNC49053.2021.9417450.

[132] D. Chen, Y. C. Liu, B. G. Kim, J. Xie, C. S. Hong, and Z. Han, "Edge Computing Resources Reservation in Vehicular Networks: A Meta-Learning Approach," *IEEE Trans Veh Technol*, vol. 69, no. 5, pp. 5634–5646, May 2020, doi: 10.1109/TVT.2020.2983445.

[133] P. Dai, Y. Huang, K. Hu, X. Wu, H. Xing, and Z. Yu, "Meta Reinforcement Learning for Multi-task Offloading in Vehicular Edge

Computing," *IEEE Trans Mob Comput*, 2023, doi: 10.1109/TMC.2023.3247579.

[134] J. Gerup, C. B. Soerensen, and P. Dieckmann, "Augmented reality and mixed reality for healthcare education beyond surgery: an integrative review," *Int J Med Educ*, vol. 11, p. 1, Jan. 2020, doi: 10.5116/IJME.5E01.EB1A.

[135] L. Hu, Y. Tian, J. Yang, T. Taleb, L. Xiang, and Y. Hao, "Ready Player One: UAV-Clustering-Based Multi-Task Offloading for Vehicular VR/AR Gaming," *IEEE Netw*, vol. 33, no. 3, pp. 42–48, May 2019, doi: 10.1109/MNET.2019.1800357.

[136] O. Muñoz, A. Pascual-Iserte, and J. Vidal, "Optimization of Radio and Computational Resources for Energy Efficiency in Latency-Constrained Application Offloading," *IEEE Trans Veh Technol*, vol. 64, no. 10, pp. 4738–4755, Oct. 2015, doi: 10.1109/TVT.2014.2372852.

[137] Z. Han, H. Tan, X. Y. Li, S. H. C. Jiang, Y. Li, and F. C. M. Lau, "OnDisc: Online Latency-Sensitive Job Dispatching and Scheduling in Heterogeneous Edge-Clouds," *IEEE/ACM Transactions on Networking*, vol. 27, no. 6, pp. 2472–2485, Dec. 2019, doi: 10.1109/TNET.2019.2953806.

[138] D. Rahbari and M. Nickray, "Task offloading in mobile fog computing by classification and regression tree," *Peer-to-Peer Networking and Applications 2019 13:1*, vol. 13, no. 1, pp. 104–122, 2019, doi: 10.1007/S12083-019-00721-7.

[139] A. A. Alli and M. M. Alam, "SecOFF-FCIoT: Machine learning based secure offloading in Fog-Cloud of things for smart city applications," *Internet of Things*, vol. 7, p. 100070, Sep. 2019, doi: 10.1016/J.IOT.2019.100070.

[140] Z. Zhu, T. Liu, Y. Yang, and X. Luo, "BLOT: Bandit Learning-Based Offloading of Tasks in Fog-Enabled Networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 30, no. 12, pp. 2636–2649, Dec. 2019, doi: 10.1109/TPDS.2019.2927978.

[141] M. K. Hussein and M. H. Mousa, "Efficient task offloading for IoT-Based applications in fog computing using ant colony optimization," *IEEE Access*, vol. 8, pp. 37191–37201, 2020, doi: 10.1109/ACCESS.2020.2975741.

[142] S. Vemireddy and R. R. Rout, "Fuzzy Reinforcement Learning for energy efficient task offloading in Vehicular Fog Computing," *Computer Networks*, vol. 199, p. 108463, Nov. 2021, doi: 10.1016/J.COMNET.2021.108463.

[143] Y. Sharma, Z. Das, and S. Moulik, "SPORTS: A Semi-partitioned Real-Time Scheduler for Heterogeneous Multicore Platforms," *Communications in Computer and Information Science*, vol. 1362, pp. 405–417, 2021, doi: 10.1007/978-981-16-0010-4_35/COVER.

[144] M. Ghobaei-Arani, R. Khorsand, and M. Ramezanpour, "An autonomous resource provisioning framework for massively multiplayer online games in cloud environment," *Journal of Network and Computer Applications*, vol. 142, pp. 76–97, Sep. 2019, doi: 10.1016/J.JNCA.2019.06.002.

[145] M. Kumar, A. Kishor, J. K. Samariya, and A. Y. Zomaya, "An Autonomic Workload Prediction and Resource Allocation Framework for Fog enabled Industrial IoT," *IEEE Internet Things J*, pp. 1–1, Jan. 2023, doi: 10.1109/JIOT.2023.3235107.

[146] F. Cheng, Y. Huang, B. Tanpure, P. Sawalani, L. Cheng, and C. Liu, "Cost-aware job scheduling for cloud instances using deep reinforcement learning," *Cluster Comput*, vol. 25, no. 1, pp. 619–631, Feb. 2022, doi: 10.1007/S10586-021-03436-8/FIGURES/5.

[147] X. Li, J. Wan, H. N. Dai, M. Imran, M. Xia, and A. Celesti, "A Hybrid Computing Solution and Resource Scheduling Strategy for Edge Computing in Smart Manufacturing," *IEEE Trans Industr Inform*, vol. 15, no. 7, pp. 4225–4234, Jul. 2019, doi: 10.1109/TII.2019.2899679.

[148] H. R. Boveiri, R. Khayami, M. Elhoseny, and M. Gunasekaran, "An efficient Swarm-Intelligence approach for task scheduling in cloud-based internet of things applications," *J Ambient Intell Humaniz Comput*, vol. 10, no. 9, pp. 3469–3479, Sep. 2019, doi: 10.1007/S12652-018-1071-1/FIGURES/8.

[149] I. H. Sarker, "Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions," *SN Comput Sci*, vol. 2, no. 6, pp. 1–20, Nov. 2021, doi: 10.1007/S42979-021-00815-1/FIGURES/6.

[150] S. H. S. Basha, S. R. Dubey, V. Pulabaigari, and S. Mukherjee, "Impact of fully connected layers on performance of convolutional neural networks for image classification," *Neurocomputing*, vol. 378, pp. 112–119, Feb. 2020, doi: 10.1016/J.NEUCOM.2019.10.008.

[151] P. Kuppusamy and V. C. Bharathi, "Human abnormal behavior detection using CNNs in crowded and uncrowded surveillance – A survey," *Measurement: Sensors*, vol. 24, p. 100510, Dec. 2022, doi: 10.1016/J.MEASEN.2022.100510.

[152] S. Tuli, S. Ilager, K. Ramamohanarao, and R. Buyya, "Dynamic Scheduling for Stochastic Edge-Cloud Computing Environments Using A3C Learning and Residual Recurrent Neural Networks," *IEEE Trans Mob Comput*, vol. 21, no. 3, pp. 940–954, Mar. 2022, doi: 10.1109/TMC.2020.3017079.

[153] S. Shadroo, A. M. Rahmani, and A. Rezaee, "The two-phase scheduling based on deep learning in the Internet of Things," *Computer Networks*, vol. 185, p. 107684, Feb. 2021, doi: 10.1016/J.COMNET.2020.107684.

[154] P. Hosseinioun, M. Kheirabadi, S. R. Kamel Tabbakh, and R. Ghaemi, "A new energy-aware tasks scheduling approach in fog computing using hybrid meta-heuristic algorithm," *J Parallel Distrib Comput*, vol. 143, pp. 88–96, Sep. 2020, doi: 10.1016/J.JPDC.2020.04.008.

[155] M. Abdel-Basset, R. Mohamed, M. Elhoseny, A. K. Bashir, A. Jolfaei, and N. Kumar, "Energy-Aware Marine Predators Algorithm for Task Scheduling in IoT-Based Fog Computing Applications," *IEEE Trans Industr Inform*, vol. 17, no. 7, pp. 5068–5076, Jul. 2021, doi: 10.1109/TII.2020.3001067.

[156] S. Javanmardi, M. Shojafar, V. Persico, and A. Pescapè, "FPFTS: A joint fuzzy particle swarm optimization mobility-aware approach to fog task scheduling algorithm for Internet of Things devices," *Softw Pract Exp*, vol. 51, no. 12, pp. 2519–2539, Dec. 2021, doi: 10.1002/SPE.2867.

[157] M. Hosseinzadeh *et al.*, "Improved Butterfly Optimization Algorithm for Data Placement and Scheduling in Edge Computing Environments," *Journal of Grid Computing 2021 19:2*, vol. 19, no. 2, pp. 1–27, Mar. 2021, doi: 10.1007/S10723-021-09556-0.

[158] E. Badidi, "QoS-Aware Placement of Tasks on a Fog Cluster in an Edge Computing Environment," *Journal of Ubiquitous Systems & Pervasive Networks*, vol. 13, no. 1, pp. 11–19, Oct. 2020, doi: 10.5383/JUSPN.13.01.002.

[159] S. Pallewatta, V. Kostakos, and R. Buyya, "QoS-aware placement of microservices-based IoT applications in Fog computing environments," *Future Generation Computer Systems*, vol. 131, pp. 121–136, Jun. 2022, doi: 10.1016/J.FUTURE.2022.01.012.

[160] M. A. Razzaque, M. Milojevic-Jevric, A. Palade, and S. Cla, "Middleware for internet of things: A survey," *IEEE Internet Things J*, vol. 3, no. 1, pp. 70–95, Feb. 2016, doi: 10.1109/JIOT.2015.2498900.

[161] R. Yu, G. Xue, and X. Zhang, "Application Provisioning in FOG Computing-enabled Internet-of-Things: A Network Perspective," *Proceedings - IEEE INFOCOM*, vol. 2018-April, pp. 783–791, Oct. 2018, doi: 10.1109/INFOCOM.2018.8486269.

[162] H. Sami, A. Mourad, H. Otrok, and J. Bentahar, "Demand-Driven Deep Reinforcement Learning for Scalable Fog and Service Placement," *IEEE Trans Serv Comput*, vol. 15, no. 5, pp. 2671–2684, 2022, doi: 10.1109/TSC.2021.3075988.

[163] M. Chen, W. Li, G. Fortino, Y. Hao, L. Hu, and I. Humar, "A Dynamic Service Migration Mechanism in Edge Cognitive Computing," *ACM Transactions on Internet Technology (TOIT)*, vol. 19, no. 2, Apr. 2019, doi: 10.1145/3239565.

[164] S. K. Mishra, D. Puthal, J. J. P. C. Rodrigues, B. Sahoo, and E. Dutkiewicz, "Sustainable Service Allocation Using a Metaheuristic Technique in a Fog Server for Industrial Applications," *IEEE Trans Industr Inform*, vol. 14, no. 10, pp. 4497–4506, Oct. 2018, doi: 10.1109/TII.2018.2791619.

[165] C. Liu, J. Wang, L. Zhou, and A. Rezaeipanah, "Solving the Multi-Objective Problem of IoT Service Placement in Fog Computing Using Cuckoo Search Algorithm," *Neural Processing Letters 2021 54:3*, vol. 54, no. 3, pp. 1823–1854, Jan. 2022, doi: 10.1007/S11063-021-10708-2.

[166] R. Mahmud, S. N. Srirama, K. Ramamohanarao, and R. Buyya, "Profit-aware application placement for integrated Fog–Cloud computing environments," *J Parallel Distrib Comput*, vol. 135, pp. 177–190, Jan. 2020, doi: 10.1016/J.JPDC.2019.10.001.

[167] B. V. Natesha and R. M. R. Guddeti, "Adopting elitism-based Genetic Algorithm for minimizing multi-objective problems of IoT service placement in fog computing environment," *Journal of Network and Computer Applications*, vol. 178, p. 102972, Mar. 2021, doi: 10.1016/J.JNCA.2020.102972.

[168] K. Dubey, S. C. Sharma, and M. Kumar, "A Secure IoT Applications Allocation Framework for Integrated Fog-Cloud Environment," *J Grid Comput*, vol. 20, no. 1, pp. 1–23, Mar. 2022, doi: 10.1007/S10723-021-09591-X/METRICS.

[169] R. Mahmud, F. L. Koch, and R. Buyya, "Cloud-fog interoperability in IoT-enabled healthcare solutions," *ACM International Conference Proceeding Series*, Jan. 2018, doi: 10.1145/3154273.3154347.

[170] N. Ahmad, N. Javaid, M. Mehmood, M. Hayat, A. Ullah, and H. A. Khan, "Fog-Cloud Based Platform for Utilization of Resources Using Load Balancing Technique," *Lecture Notes on Data Engineering and Communications Technologies*, vol. 22, pp. 554–567, 2019, doi: 10.1007/978-3-319-98530-5_48/COVER.

[171] Y. Dong, G. Xu, M. Zhang, and X. Meng, "A high-efficient joint 'Cloud-Edge' aware strategy for task deployment and load balancing," *IEEE Access*, vol. 9, pp. 12791–12802, 2021, doi: 10.1109/ACCESS.2021.3051672.

[172] G. Li, Y. Yao, J. Wu, X. Liu, X. Sheng, and Q. Lin, "A new load balancing strategy by task allocation in edge computing based on intermediary nodes," *EURASIP J Wirel Commun Netw*, vol. 2020, no. 1, pp. 1–10, Dec. 2020, doi: 10.1186/S13638-019-1624-9/FIGURES/5.

[173] S. P. Singh, A. Sharma, and R. Kumar, "Design and exploration of load balancers for fog computing using fuzzy logic," *Simul Model Pract Theory*, vol. 101, p. 102017, May 2020, doi: 10.1016/J.SIMPAT.2019.102017.

[174] D. Baburao, T. Pavankumar, and C. S. R. Prabhu, "Load balancing in the fog nodes using particle swarm optimization-based enhanced dynamic resource allocation method," *Applied Nanoscience 2021*, pp. 1–10, Jul. 2021, doi: 10.1007/S13204-021-01970-W.

[175] M. Kumar, A. Kishor, J. K. Samariya, and A. Y. Zomaya, "An Autonomic Workload Prediction and Resource Allocation Framework for Fog enabled Industrial IoT," *IEEE Internet Things J*, pp. 1–1, Jan. 2023, doi: 10.1109/JIOT.2023.3235107.

[176] Z. Allam, A. Sharifi, S. E. Bibri, D. S. Jones, and J. Krogstie, "The Metaverse as a Virtual Form of Smart Cities: Opportunities and Challenges for Environmental, Economic, and Social Sustainability in Urban Futures," *Smart Cities 2022, Vol. 5, Pages 771-801*, vol. 5, no. 3, pp. 771–801, Jul. 2022, doi: 10.3390/SMARTCITIES5030040.

[177] S. P. Singh, A. Sharma, and R. Kumar, "Design and exploration of load balancers for fog computing using fuzzy logic," *Simul Model Pract Theory*, vol. 101, p. 102017, May 2020, doi: 10.1016/J.SIMPAT.2019.102017.

[178] P. G. V. Naranjo, Z. Pooranian, M. Shojafar, M. Conti, and R. Buyya, "FOCAN: A Fog-supported smart city network architecture for management of applications in the Internet of Everything environments," *J Parallel Distrib Comput*, vol. 132, pp. 274–283, Oct. 2019, doi: 10.1016/J.JPDC.2018.07.003.

[179] M. S. Aslanpour *et al.*, "Serverless Edge Computing: Vision and Challenges," *ACM International Conference Proceeding Series*, Feb. 2021, doi: 10.1145/3437378.3444367.

[180] P. Castro, V. Ishakian, V. Muthusamy, and A. Slominski, "The rise of serverless computing," *Commun ACM*, vol. 62, no. 12, pp. 44–54, Nov. 2019, doi: 10.1145/3368454.

[181] R. Mahmud, K. Ramamohanarao, and R. Buyya, "Application Management in Fog Computing Environments," *ACM Computing Surveys (CSUR)*, vol. 53, no. 4, Jul. 2020, doi: 10.1145/3403955.

[182] E. D. N. Ndih and S. Cherkaoui, "On Enhancing Technology Coexistence in the IoT Era: ZigBee and 802.11 Case," *IEEE Access*, vol. 4, pp. 1835–1844, 2016, doi: 10.1109/ACCESS.2016.2553150.

[183] 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 Integration," *ACM Computing Surveys (CSUR)*, vol. 51, no. 6, Jan. 2019, doi: 10.1145/3292674.

[184] N. Hassan, K. L. A. Yau, and C. Wu, "Edge computing in 5G: A review," *IEEE Access*, vol. 7, pp. 127276–127289, 2019, doi: 10.1109/ACCESS.2019.2938534.

[185] B. Alhayani *et al.*, "5G standards for the Industry 4.0 enabled communication systems using artificial intelligence: perspective of smart healthcare system," *Applied Nanoscience (Switzerland)*, vol. 1, pp. 1–11, Jan. 2022, doi: 10.1007/S13204-021-02152-4/FIGURES/6.

[186] H. Boyes, B. Hallaq, J. Cunningham, and T. Watson, "The industrial internet of things (IIoT): An analysis framework," *Comput Ind*, vol. 101, pp. 1–12, Oct. 2018, doi: 10.1016/J.COMPIND.2018.04.015.

[187] J. Sengupta, S. Ruj, and S. Das Bit, "A Comprehensive Survey on Attacks, Security Issues and Blockchain Solutions for IoT and IIoT," *Journal of Network and Computer Applications*, vol. 149, p. 102481, Jan. 2020, doi: 10.1016/J.JNCA.2019.102481.

[188] J. Sengupta, S. Ruj, and S. Das Bit, "A Secure Fog-Based Architecture for Industrial Internet of Things and Industry 4.0," *IEEE Trans Industr Inform*, vol. 17, no. 4, pp. 2316–2324, Apr. 2021, doi: 10.1109/TII.2020.2998105.

[189] G. Li, J. Wu, J. Li, K. Wang, and T. Ye, "Service Popularity-Based Smart Resources Partitioning for Fog Computing-Enabled Industrial Internet of Things," *IEEE Trans Industr Inform*, vol. 14, no. 10, pp. 4702–4711, Oct. 2018, doi: 10.1109/TII.2018.2845844.

[190] T. Hewa, A. Braeken, M. Liyanage, and M. Ylianttila, "Fog Computing and Blockchain-Based Security Service Architecture for 5G Industrial IoT-Enabled Cloud Manufacturing," *IEEE Trans Industr Inform*, vol. 18, no. 10, pp. 7174–7185, Oct. 2022, doi: 10.1109/TII.2022.3140792.

[191] S. El Kafhali, C. Chahir, M. Hanini, and K. Salah, "Architecture to manage internet of things data using blockchain and fog computing," *ACM International Conference Proceeding Series*, Oct. 2019, doi: 10.1145/3372938.3372970.

[192] B. Dammak, M. Turki, S. Cheikhrouhou, M. Baklouti, R. Mars, and A. Dhahbi, "LoRaChainCare: An IoT Architecture Integrating Blockchain and LoRa Network for Personal Health Care Data Monitoring," *Sensors 2022, Vol. 22, Page 1497*, vol. 22, no. 4, p. 1497, Feb. 2022, doi: 10.3390/S22041497.

[193] S. Mihai *et al.*, "Digital Twins: A Survey on Enabling Technologies, Challenges, Trends and Future Prospects," *IEEE Communications Surveys and Tutorials*, 2022, doi: 10.1109/COMST.2022.3208773.

[194] H. Yin and L. Wang, "Application and Development Prospect of Digital Twin Technology in Aerospace," *IFAC-PapersOnLine*, vol. 53, no. 5, pp. 732–737, Jan. 2020, doi: 10.1016/J.IFACOL.2021.04.165.

[195] Z. Yang, M. Zolanvari, and R. Jain, "A Survey of Important Issues in Quantum Computing and Communications," *IEEE Communications Surveys and Tutorials*, vol. 25, no. 2, pp. 1059–1094, 2023, doi: 10.1109/COMST.2023.3254481.

[196] S. S. Gill, "Quantum and blockchain based Serverless edge computing: A vision, model, new trends and future directions," *Internet Technology Letters*, p. e275, Feb. 2021, doi: 10.1002/ITL2.275.

[197] F. Sun, Z. Zhang, S. Zeadally, G. Han, and S. Tong, "Edge Computing-Enabled Internet of Vehicles: Towards Federated Learning Empowered Scheduling," *IEEE Trans Veh Technol*, vol. 71, no. 9, pp. 10088–10103, Sep. 2022, doi: 10.1109/TVT.2022.3182782.

[198] C. Chen, J. Hu, T. Qiu, M. Atiquzzaman, and Z. Ren, "CVCG: Cooperative V2V-Aided Transmission Scheme Based on Coalitional Game for Popular Content Distribution in Vehicular Ad-Hoc Networks," *IEEE Trans Mob Comput*, vol. 18, no. 12, pp. 2811–2828, Dec. 2019, doi: 10.1109/TMC.2018.2883312.

[199] S. Pandya *et al.*, "Federated learning for smart cities: A comprehensive survey," *Sustainable Energy Technologies and Assessments*, vol. 55, p. 102987, Feb. 2023, doi: 10.1016/J.SETA.2022.102987.

[200] L. U. Khan, W. Saad, Z. Han, E. Hossain, and C. S. Hong, "Federated learning for internet of things: Recent advances, taxonomy, and open challenges," *IEEE Communications Surveys and Tutorials*, vol. 23, no. 3, pp. 1759–1799, Jul. 2021, doi: 10.1109/COMST.2021.3090430.

[201] R. Saha, S. Misra, and P. K. Deb, "FogFL: Fog-Assisted Federated Learning for Resource-Constrained IoT Devices," *IEEE Internet Things J*, vol. 8, no. 10, pp. 8456–8463, May 2021, doi: 10.1109/JIOT.2020.3046509.

[202] A. Yazdinejad, R. M. Parizi, A. Dehghantanha, Q. Zhang, and K. K. R. Choo, "An Energy-Efficient SDN Controller Architecture for IoT Networks with Blockchain-Based Security," *IEEE Trans Serv Comput*, vol. 13, no. 4, pp. 625–638, Jul. 2020, doi: 10.1109/TSC.2020.2966970.

[203] M. Ojo, D. Adami, and S. Giordano, "A SDN-IoT architecture with NFV implementation," *2016 IEEE Globecom Workshops, GC Wkshps 2016 - Proceedings*, 2016, doi: 10.1109/GLOCOMW.2016.7848825.

[204] B. Kitchenham, "Procedures for performing systematic reviews," *Keele, UK, Keele University*, vol. 33, no. 2004, pp. 1–26, 2004.

[205] H. Xiao, J. Zhao, Q. Pei, J. Feng, L. Liu, and W. Shi, "Vehicle Selection and Resource Optimization for Federated Learning in Vehicular Edge Computing," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 11073–11087, Aug. 2022, doi: 10.1109/TITS.2021.3099597.

[206] B. Sellami, A. Hakiri, and S. Ben Yahia, “Deep Reinforcement Learning for energy-aware task offloading in join SDN-Blockchain 5G massive IoT edge network,” *Future Generation Computer Systems*, vol. 137, pp. 363–379, Dec. 2022, doi: 10.1016/J.FUTURE.2022.07.024.

[207] T. Wang, G. Zhang, A. Liu, M. Z. A. Bhuiyan, and Q. Jin, “A secure IoT service architecture with an efficient balance dynamics based on cloud and edge computing,” *IEEE Internet Things J*, vol. 6, no. 3, pp. 4831–4843, Jun. 2019, doi: 10.1109/JIOT.2018.2870288.

[208] Q. Li, S. Meng, S. Zhang, J. Hou, and L. Qi, “Complex attack linkage decision-making in edge computing networks,” *IEEE Access*, vol. 7, pp. 12058–12072, 2019, doi: 10.1109/ACCESS.2019.2891505.

[209] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao, “A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications,” *IEEE Internet Things J*, vol. 4, no. 5, pp. 1125–1142, Oct. 2017, doi: 10.1109/JIOT.2017.2683200.

[210] W. Z. Khan, E. Ahmed, S. Hakak, I. Yaqoob, and A. Ahmed, “Edge computing: A survey,” *Future Generation Computer Systems*, vol. 97, pp. 219–235, Aug. 2019, doi: 10.1016/J.FUTURE.2019.02.050.

[211] B. Tang, Z. Chen, G. Hefferman, T. Wei, H. He, and Q. Yang, “A hierarchical distributed fog computing architecture for big data analysis in smart cities,” *ACM International Conference Proceeding Series*, vol. 07-09-Ocobert-2015, Oct. 2015, doi: 10.1145/2818869.2818898.

[212] S. Hoque, M. S. De Brito, A. Willner, O. Keil, and T. Magedanz, “Towards Container Orchestration in Fog Computing Infrastructures,” *Proceedings - International Computer Software and Applications Conference*, vol. 2, pp. 294–299, Sep. 2017, doi: 10.1109/COMPSAC.2017.248.

[213] C. Tang, X. Wei, C. Zhu, Y. Wang, and W. Jia, “Mobile Vehicles as Fog Nodes for Latency Optimization in Smart Cities,” *IEEE Trans Veh Technol*, vol. 69, no. 9, pp. 9364–9375, Sep. 2020, doi: 10.1109/TVT.2020.2970763.

[214] 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, Oct. 2016, doi: 10.1109/JIOT.2016.2579198.

[215] S. Aggarwal and N. Kumar, “Fog Computing for 5G-Enabled Tactile Internet: Research Issues, Challenges, and Future Research Directions,” *Mobile Networks and Applications*, pp. 1–28, Nov. 2019, doi: 10.1007/S11036-019-01430-4/TABLES/9.

[216] I. Al Ridhawi, M. Aloqaily, Y. Kotb, Y. Al Ridhawi, and Y. Jararweh, “A collaborative mobile edge computing and user solution for service composition in 5G systems,” *Transactions on Emerging Telecommunications Technologies*, vol. 29, no. 11, p. e3446, Nov. 2018, doi: 10.1002/ETT.3446.

[217] G. Qu, H. Wu, R. Li, and P. Jiao, “DMRO: A Deep Meta Reinforcement Learning-Based Task Offloading Framework for Edge-Cloud Computing,” *IEEE Transactions on Network and Service Management*, vol. 18, no. 3, pp. 3448–3459, Sep. 2021, doi: 10.1109/TNSM.2021.3087258.

[218] L. Huang, L. Zhang, S. Yang, L. P. Qian, and Y. Wu, “Meta-Learning Based Dynamic Computation Task Offloading for Mobile Edge Computing Networks,” *IEEE Communications Letters*, vol. 25, no. 5, pp. 1568–1572, May 2021, doi: 10.1109/LCOMM.2020.3048075.

[219] 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, Jan. 2022, doi: 10.1016/J.COMCOM.2021.11.005.

[220] B. Dammak, M. Turki, S. Cheikhrouhou, M. Baklouti, R. Mars, and A. Dhahbi, “LoRaChainCare: An IoT Architecture Integrating Blockchain and LoRa Network for Personal Health Care Data Monitoring,” *Sensors 2022, Vol. 22, Page 1497*, Feb. 2022, doi: 10.3390/S22041497.

[221] R. Singh and S. S. Gill, "Edge AI: a survey." *Internet of Things and Cyber-Physical Systems* 2023, Vol. 3, Page 71-92, doi: 10.1016/j.iotcps.2023.02.004.

[222] A. R. Nandhakumar, et al. "EdgeAISim: A Toolkit for Simulation and Modelling of AI Models in Edge Computing Environments." *Measurement: Sensors*, 2023.

[223] M. Attaran and B. G. Celik, “Digital Twin: Benefits, use cases, challenges, and opportunities,” *Decision Analytics Journal*, vol. 6, p. 100165, Mar. 2023, doi: 10.1016/J.DAJOUR.2023.100165.

Guneet Kaur Walia is a Ph.D scholar at the Department of Information Technology, Dr. B. R. Ambedkar National Institute of Technology, Jalandhar, Punjab, India. She successfully completed her Masters in Computer Science Engineering at Punjab Agricultural University, Ludhiana, Punjab, in 2016. Her research interests includes Cloud Computing, Edge Computing, Internet of Things (IoT), Resource Management in Edge Computing, and Artificial Intelligence (AI).

Mohit Kumar is Assistant Professor in the Department of Information Technology at Dr. B R Ambedkar National Institute of Technology, Jalandhar, India. He received his Ph.D. degree from Indian Institute of Technology Roorkee in the field of Cloud Computing, 2018, and M. Tech degree in Computer Science and Engineering from ABV-Indian Institute of Information Technology Gwalior, India in 2013. His research topics cover the areas of Cloud computing, Fog/ Edge Computing, Internet of Things, federated learning, Blockchain, and Artificial Intelligence. He has published more than 55 research articles in reputed journals, IEEE Transactions and international conferences. He has been Session chair and keynotes Speaker of many International conferences, webinars, FDP, STC in India. He has guided six M. Tech Thesis and guiding 5 Ph.D. Scholar. He is an active reviewer of several reputed journals and international conferences. He is a member of the IEEE.

Sukhpal Singh Gill (FHEA) is an Assistant Professor of Cloud Computing at the School of Electronic Engineering and Computer Science, Queen Mary University of London, UK. Prior to his present stint, Dr. Gill has held positions as a Research Associate at the Lancaster University, UK and also as a Postdoctoral Research Fellow at CLOUDS Laboratory, The University of Melbourne, Australia. Dr. Gill is serving as an Associate Editor in IEEE IoT, Wiley SPE, Elsevier IoT, Wiley ETT and IET Networks Journal. He has co-authored 150+ peer-reviewed papers (with 6800+ citations and H-index 40) and has published in prominent international journals and conferences such as ACM CSUR, IEEE TCC, IEEE TSC, IEEE TII, IEEE TSUSC, IEEE TNSM, IEEE IoT Journal, Elsevier JSS/FGCS, IEEE/ACM UCC and IEEE CCGRID. His research interests include Cloud Computing, Edge Computing, IoT and Energy Efficiency. For further information, please visit: http://www.ssgill.me