IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 9 January 2024, accepted 2 March 2024, date of publication 13 March 2024, date of current version 19 March 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3375368

SURVEY

Fog Offloading and Task Management in IoT-Fog-Cloud Environment: Review of Algorithms, Networks, and SDN Application

MOHAMMAD REZA REZAEE[1,2], NOR ASILAH WATI ABDUL HAMID[1,3], (Senior Member, IEEE), MASNIDA HUSSIN[1], (Senior Member, IEEE), AND ZURIATI AHMAD ZUKARNAIN[1], (Member, IEEE)

[1]Department of Communication Technology and Network, Faculty of Computer Science and Information Technology, University Putra Malaysia, Serdang 43400, Malaysia
[2]Department of Information Technology, Faculty of Computer Science, Kabul Education University, Kabul 1001, Afghanistan
[3]Institute for Mathematical Research, University Putra Malaysia, Serdang 43400, Malaysia

Corresponding author: Nor Asilah Wati Abdul Hamid (asila@upm.edu.my)

This material is based upon work supported by the Air Force Office of Scientific Research under award number FA2386-22-1-4018.

ABSTRACT The proliferation of Internet of Things (IoT) devices and other IT forms in almost every area of human existence has resulted in an enormous influx of data that must be managed and stored. One viable solution to this issue is to store and handle massive amounts of data in cloud environments. Real-time data analysis has always been critical. However, it becomes even more crucial as technology and the IoT develop, and new applications emerge, such as autonomous cars, smart cities, and IoT devices for healthcare, agriculture, and other industries. Given the massive volume of data, moving to a remote cloud is time-consuming and produces severe network congestion, rendering cloud administration and rapid data processing difficult. Fog computing provides close-to-device processing at the network's periphery, and fog computing can analyze data in near real-time. However, the increased amount of IoT gadgets and data they produce is a formidable challenge for fog nodes. Task offloading may enhance fog computing by offloading the excess data to other nodes for processing due to the restricted resources in the fog. Management of tasks and resources must be optimized in fog devices. This review article overviews related works on task offloading in IoT-Fog-Cloud Environment. In addition, we discuss about fog networks and Software-defined network (SDN) applications and challenges in fog offloading.

INDEX TERMS Fog computing, cloud computing, task offloading, task management, fog offloading, software defined network (SDN).

I. INTRODUCTION

The vast development of intelligent devices and the rise of cloud computing have led to exponential growth in the quantity of data generated by IoT gadgets. Furthermore, cutting-edge IoT applications need ultra-low latency for data transmission and processing. These include augmented reality (AR), virtual reality (VR), autonomous driving, and intelligent manufacturing. A new computer paradigm, "fog computing," is designed to fulfil these needs. Rather than transmitting large amounts of data to a centralized cloud server, delay-sensitive apps like augmented reality (AR), virtual reality (VR), and online games may do the necessary processing at fog nodes. It would significantly lessen the

The associate editor coordinating the review of this manuscript and approving it for publication was Massimo Cafaro.

latency of information processing and reduce the quantity of data sent inside the primary networks. In fog computing, fog nodes are located close to the IoT devices so that they may act as a bridge between them and the public cloud. Depending on the structure, a fog node might be a cloudlet, a micro-data center, or an IoT gateway. The benefits of fog computing have yet to be realized entirely despite many practical and educational efforts [1], [2]. Since the IoT generates copious amounts of data, cloud computing may be used to manage and analyze this deluge of information. However, real-time processing is essential for many IoT devices. Fog computing provides a great answer to this problem. Offloading tasks from one fog to another node is crucial for continuing processing operations without interruption due to the limited resources in fogs for heavy-duty processing, particularly for procedures that need to be handled in real-time or quickly.

Fog offloading and SDN have been widely used across several industries, showcasing their capacity to improve operational efficiency and decision-making. Fog computing is employed in smart cities to decrease latency and power usage, which is essential for IoT applications such as traffic management and public safety systems. Additionally, wireless fog networks enabled by SDN offer reduced latency and effective load balancing [3], [4], [5]. Within the healthcare sector, it facilitates expedited processing of data at the periphery, hence enhancing the promptness and dependability of patient treatment [6], [7] Fog computing and SDN are advantageous for industrial automation since they provide predictive maintenance and process optimization, particularly in industrial Internet of Things (IIoT) systems, ensuring real-time performance and high reliability [8], [9].

Vehicular ad-hoc networks (VANET) [10] and vehicular networks [11] integrate SDN and fog computing, while [12] employs a fusion of fog computing, SDN, and blockchain to enhance IoT sensors in agriculture. Additionally, within the domain of Intelligent Transportation Systems (ITS), [13] highlights further innovative deployments and [14] focus on prediction of forest fires via distributed machine learning on a fog network. These examples highlight the extensive influence of the technology across several industries.

An evaluation of existing research shows that several frameworks and methods have been used to offload computation from the fog onto other nodes in the computing infrastructure. While algorithms provide many options for dealing with this problem, including offloading to the cloud, the most practical approach is to upload work to another fog due to the high expense of cloud offloading. The challenge now is picking the right node from all the available ones on the network—one that has enough computing power, a fast enough connection, enough bandwidth, and is not too far from the original place.

Literature evaluation and research of various methods and methodologies show that centralized and distributed architectures have benefits and drawbacks. The fogs assemble their data and the rest of the network's nodes and linkages. In addition to placing a heavy computational burden on the fog and communication lines, this effort also needs to improve the efficiency of the underlying network and computing fog. The Centralized architecture has performance concerns since it relies on a single device, making it vulnerable to outages in case of a communication breakdown between the nodes.

In light of the numerous complexities surrounding offloading in fog computing, with particular emphasis on the intricate nature of network structures, this review paper delves into a comprehensive analysis of the prevailing challenges and their corresponding solutions. Our objective is to scrutinize the diverse aspects of network utilization in fog offloading and explore the applications, benefits, and obstacles associated with SDN technology within this domain.

The following are the primary contributions of this survey:

1. A thorough analysis of the fog offloading issue classifies the most recent solutions and identifies the various aspects impacting the offloading decision.

2. We discuss and explore diverse types of offloading networks by focusing on SDN applications in this field.

3. We present a classification for the optimization methodologies to tackle the service placement issue for IoT applications deployed across fog nodes.

4. Finally, we explore future research directions in fog-based systems and identify great difficulties.

We will go deeply into each issue in the following parts of this review article. In the second part, we will review the significant components, ideas, and aspects of fog computing. In the third section, the related surveys are discussed. We will then perform a detailed assessment and evaluation of fog offloading related approaches and analyzing recent studies in the fourth part. In the fifth part, we will assess and evaluate the approaches and categorization of fog computing architectures and SDN. In the sixth part, we will discuss the results, emphasizing both the advantages and problems connected with SDN in fog computing. Finally, in the last part, we will make conclusions that will pave the way for future research areas in fog computing.

II. OVERVIEW OF FUNDAMENTAL CONCEPTS

A. IoT

The term ''Internet of Things'' (IoT) refers to a network of physical things, such as buildings, cars, and other objects, that collect and distribute data using software, sensors, electronics, and network connections. IoT devices have confronted storage, communication, energy, and computing limits. Due to these inefficiencies, IoT and cloud computing technologies are combined [15].

Many IoT systems and information science addresses associated with IoT devices are connected to the internet to provide end users and businesses with regular services and tasks. Applications for the IoT have exploded in recent years, and predictions indicate that this trend will continue. More than seven billion IoT devices were linked to the internet in 2018; fourteen billion were anticipated to do so in 2019. Among the IoT applications with the highest number of deployed devices is the home sector, with 663 million devices in use in 2017. Some of the purposes for intelligent homes include locks, freezers, stoves, refrigerators, and lighting. Expanded IoT usage, or the ''smart city,'' has been proposed for many European nations. Automation, pricing potency, and exactness have contributed most to the present trend in industrial, agricultural, and health applications. Patient monitoring, energy-saving, and imaging-related technologies are a few examples of healthcare gadgets (X-ray machines) [16].

B. CLOUD COMPUTING

''IoT'' and ''cloud computing'' are the foundation of a swift and autonomous transformation. IoT technology's processing, storage, and communications constraints may initially be made up for by the almost limitless cloud computing resources and capabilities. Additionally, cloud computing may be advantageous in many real-world situations by allowing new services. IoT technology has extended its reach to solve real-world issues in a more distributed and dynamic fashion [17].

Connecting self-aware, intelligent nodes, or ''things,'' in a dynamic, worldwide network architecture is the core concept of the IoT. It supports ubiquitous and pervasive computing potential and is one of the most problematic technologies. Some define the IoT as a phrase for small, universal, global, and responsible items with limited processing and storage power and privacy, security, and accountability concerns. On the other hand, cloud computing is a highly developed technology that offers almost endless processing and storage capacity and has addressed or substantially resolved most IoT issues. As such, it is anticipated that a single IT paradigm—in which Cloud and IoT are complementary technologies—will disrupt the current and future internet [18]. The necessity for an interface between cloud and fog infrastructures was seen from the beginning of the fog concept [19]. Additionally, [18] addresses the interaction between such entities; the standardization of this interface is highlighted as one of the crucial challenges [20].

C. EDGE COMPUTING

In order to improve reaction times and save bandwidth, edge computing and fog computing—two essential elements of distributed computing—seek to move processing and data storage closer to the point of need. Even if their goals are identical, their distinct qualities may either support or contradict one another. Whereas fog computing makes use of an intermediate layer, such as routers or gateway devices, edge computing processes data on devices at the edge of the network, such as smartphones and IoT sensors. Through a layered computing architecture, the synergy of these two enables a more effective allocation of computational activities, boosting efficiency, performance, scalability, and system dependability. Edge computing requires the construction of tiny, virtualized infrastructures between base stations, radio network controllers, and other aggregation sites at the network edge. Although edge computing uses an architecture and an operating system distinct from fog computing (one operator maintains the infrastructure, excluding consumer devices), it targets applications comparable to fog computing. Edge computing also leverages telecom infrastructures to provide mobile edge services, including location, information measure management, and radio network information [21].

To prolong the battery life of edge devices, the applications running on edge devices may be efficiently migrated to fog devices that possess sufficient resources and battery capacity. This approach has been demonstrated in research [22].

Nevertheless there are additional difficulties in combining edge and fog computing. It is difficult to manage resources between fog nodes and edge devices, and the increasing data processing at the edge raises security and privacy issues. Interoperability problems may arise from a lack of standardisation, and in certain situations, the interdependency on network connection presents difficulties. Furthermore, the deployment and maintenance of these integrated systems are very complicated, and the energy consumption and sustainability of such a dispersed network need cautious management. All things considered, edge computing enhances fog computing by improving processing power, but it also brings new challenges that must be resolved in order to maximise dispersed computing systems.

D. FOG COMPUTING

Fog computing has emerged as a feasible supplement to cloud computing with the advent of real-time IoT devices, bringing cloud computing to the edge of the network to fulfill the stringent latency constraints and intense processing needs of these applications [22], [23]. A typical fog compute node comprises several geographically distributed fog nodes positioned at the network's edge with different resource provisioning, such as storage, processing power, and communication bandwidth [24]. Fog computing bridges edge devices and the cloud, offering low-latency connections and a dynamic environment. It is similar to the edge in architecture and closer to the cloud regarding resources, capable of performing computations and storing data fields [25].

A cloud extension known as fog computing occasionally reduces latency and supports time-sensitive applications, including online gaming, healthcare, and autonomous driving. Local devices execute simple activities, whereas large cloud installations handle sophisticated procedures. Fog computing collects virtual resources from several devices scattered around the environment, each managed by a different entity [21]. The IoT enables connections between objects and networks. Fog computing, an innovative concept, provides computer, storage, and networking services to IoT devices (such as sensors or embedded devices) at the network's edge. Fog computing benefits mobile computing swiftly analyses massive amounts of data and manages billions of internet-connected devices.

Fog property, quality of service, interface and programming model, machine offloading, scheduling, accounting and tracking, resource and repair management, and information privacy are among the fundamental problems in fog computing. Fog computing eliminates interruptions with an economical cloud architecture. Positioning fog nodes adjacent to the information source provides cloud support options. Instead of transferring IoT data to the cloud, fog evaluates and stores it locally at IoT devices, enabling fog to offer higher-quality services with quicker response times. Fog is the ideal choice for allowing the IoT to provide a range of IoT applications in an IoT system with practical and dependable services [26].

The closeness of end users, regionally targeted allocation, and quality allowance will distinguish fog from the cloud. Fog substantially minimizes network information measure utilization when compared to the cloud. In cloud based IoT data processing, each unique piece of IoT data must be sent to the cloud data center. Transferring data becomes increasingly costly if the number of data to be reviewed rises fast as is the situation with today's IoT. Because of the fog, information is kept and processed locally, reducing the utility of sending such information. If processed data is required to be retained for different analytical and historical reasons, it is uploaded to the cloud. Consumers would benefit from cheaper operational expenses as a consequence. Furthermore, since the data is handled so close to the source, the end-user application becomes very fast, which is crucial for maintaining the quality of service for period and Mobile to Mobile operations. As a result, fog management services improve customer satisfaction while IoT information services become more reliable, efficient, and consistent [26].

Even though fog networking seems to be a realistic solution for overcoming the drawbacks of cloud computing and current networks, some concerns must be addressed in the future. The most critical need is a distributed intelligent platform for managing computing, networking, and storage resources at the edge. Given the wide variety of processing power capabilities of nodes, the uncertainty surrounding task demands, resources available at fog nodes, and distribution choices, it is challenging to make an acceptable distribution decision in fog networks [27]. Furthermore, communication delays between nodes should be considered when selecting a distribution, since this may lead to more extended processing times [28]. Consequently, the Fog computing paradigm confronts several challenges [29].

A fog node's response period is used to evaluate whether it should handle the whole of a service request that has been received, process just a part of it, or offload it to another fog.

Each fog's reaction time will be determined regularly utilizing the fog's available capacity (i.e., waiting line volume) and the service's demand travel time (lower latency is always desirable) [30].

Each fog in the network tries to gather data about other fogs and nodes nearby. Whenever a fog has to offload its responsibilities to other nodes, it stays connected to the surrounding nodes, changes the value of the computational function and the state of its connection connections, and decides about which fog to offload based on an algorithm that analyses the different fogs.

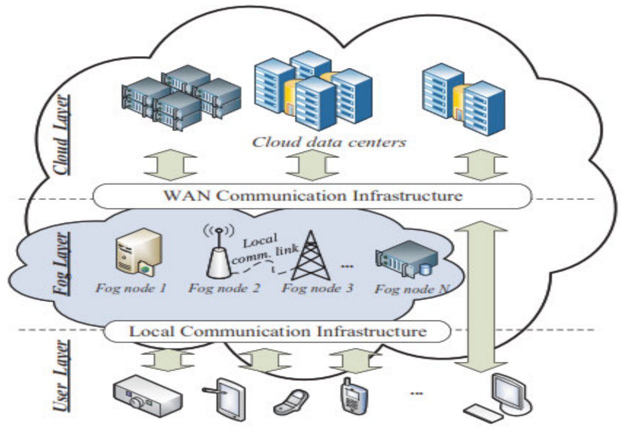

FIGURE 1. A three-layer fog computing architecture [22].

In contrast, the centralized architecture relies on a server or other centralized device to gather data from distributed fogs, analyze computing resources, and map out connections between distributed fogs throughout a network. Send the communication connections from the nodes constantly to the device in the center. A node will communicate with this centralized machine when it wants to submit its duties. Central devices pick the target fog for offloading and introduce the source node based on nodes and network connections following the decision process for which the choice is made. A three-layer fog computing architecture is shown in Figure 1.

E. TASK OFFLOADING

IoT and mobile devices are not suited for running programs needing significant resources since they have limited memory, computing power, battery life, and measurement of communication data. Mobile cloud computing was developed to solve the current problem and eliminate such challenges. It is a paradigm for shifting heavy-lifting mobile device tasks to the cloud or compute offloading, which frees up mobile devices' resource constraints. Several constraints, including a sizeable average access latency between users and remote clouds, high battery consumption, and a deficiency of local user data, make offloading activities to the cloud possible only for specific mobile applications. Novel approaches are being presented to address these problems, including offloading mobile network traffic to complementary networks like Wi-Fi and satellite-terrestrial networks and edge cloud computing [31].

Consider the case when a fog node first receives and processes a request for data processing from a thing before responding. When a fog node is overloaded with other requests, it may only process a portion of the payload before sending it to other fog nodes. There are two ways to simulate fog node interactions: With the first approach, a central node keeps up the fog node offload contact. Every fog node follows a protocol to provide its most current status information to

FIGURE 2. Task offloading scopes related to fog computing.

its neighbors. Second, every fog node maintains a frequently updated list of the best nodes to manage the offloaded workloads [30].

Offloading must handle many challenges in fog computing, including: How to construct the best offloading scheme, how to partition an application for offloading, and what sort of data is needed for offloading decisions are the first three questions [24].

Offloading must handle many challenges in fog computing, including: How to construct the best offloading scheme, how to partition an application for offloading, and what sort of data is needed for offloading decisions are the first three questions [13], [32], [33].

According to [23], resource allocation and dynamic offloading in fog computing present many vital challenges. These challenges include the following:

1) The Power-Latency Tradeoff and System Dynamics: A fog system often consists of several layers, intricate dynamic interactions between fog tiers and the cloud, and dynamic internal dynamics that are always changing.

2) Effective Decision-Making Process: Because of overheads, decision-making processes should be computationally effective. The challenges are often brought on by the unpredictable nature of the traffic data, the ongoing nature of task delivery, and the intrinsic complexity of the issue.

3) Understanding the Benefits of Predictive Offloading: A crucial aspect of online decision-making is using predictive offloading to reduce delays and improve the quality of service. Although fog computing prediction offloading has various applications, its primary constraints remain a mystery.

The key topic that will be explored is resource management in terms of computing resources and data storage amongst different fogs in the network. IoT devices may offload tasks to fog nodes fog nodes to other fog nodes, and fog nodes to clouds.

The tasks are re-offloaded to the surrounding fog nodes or cloud due to the fog node's low processing resources [34], [35], [36], [37], that resulted in more transmission and processing costs [37], [38]. Figure 2 depicts the scopes of task offloading in fog computing.

Here, we will present and discuss related fog processing and offloading efforts. Due to its relevance and significance, the fog processing debate has garnered the attention of several scholars.

F. SOFTWARE DEFINED NETWORK (SDN)

SDN is an advanced networking concept that divides the control plane from the data plane [39]. This leads to increased flexibility and agility as well as more straightforward network design and management. SDN's main idea is to let a logically centralized software-based controller, or control plane, take care of network intelligence and decision-making, with the data plane handling tasks like traffic forwarding. SDN has many advantages, including network programmability, which promotes network automation, and network management, which lowers operating costs by streamlining administration activities. And virtualization of networks [40].

Most device power consumption in IoT applications happens during packet forwarding on the network side [13], [41]. Device life and offloading quality will suffer from poor-quality offloading brought on by limited bandwidth and excessive latency. SDN has evolved as an auxiliary technology on the network side in offloading operations to alleviate this issue [13], [42].

Centralized control, programmability, load balancing, and management are just some of the problems that plague traditional Fog networks [43], [44], [45]. Fog nodes can efficiently operate together using SDN [2], [45], [46]. By enhancing the rules in its centralized SDN controller with a network-wide perspective [47], SDN allows for more wiggle room in the network's programming. By gathering information from each network device, adjusting the load-balancing plan, and keeping an eye on network traffic, an SDN controller dynamically manages the network [45], [48], [49].

Gathering network data, such as utilization of resources, device positioning and mobility, load metrics, and network information, is the primary duty of the SDN controller in order to ensure the field of SDN-enabled networks provides a high level of service [13], [50], [51], [52], [53]. The offloading service with SDN can detect changes in the network since it uses the SDN controller according to the compute demands and network conditions, it may select the best offloading node for an overloaded fog node.

III. RELATED SURVEYS

As pointed out earlier, merely transferring and offloading data across Fogs won't provide the perfect setup without considering task management and resource availability. Nevertheless, the majority of prior review papers have addressed algorithms from a broad perspective, failing to take into account the specifics of the algorithm's capacity for task and resource management. This work makes a substantial addition to the area of fog computing by analyzing fog analysis algorithms based on the three primary capabilities of task management, resource management, and task offloading.

This study diverges from recent surveys focused on fog and Edge computing, mobile Edge computing, or machine learning applications. Instead, it offers an in-depth analysis of network concerns and SDN applications, specifically within the framework of fog offloading. This extensive examination of network and SDN issues fills a gap in previous research, which either ignored or briefly addressed these aspects. As a result, it makes a substantial addition to the subject. This approach fills an essential need in existing literature, establishing our study as a crucial asset in comprehending and progressing network management tactics in these developing computing paradigms.

Surveys [39], [40], and [44] mainly examined Edge computing, mobile Edge computing, and general SDN usage. However, they either neglected or just briefly addressed network concerns. On the other hand, this study provides a detailed analysis of these topics, addressing a significant deficiency in the existing body of knowledge and offering valuable perspectives for enhancing network management tactics in Fog computing.

In [54], task offloading in edge computing and the IoT is investigated. The study considers the benefits, factors that impact the decision of offloading, and different offloading techniques and algorithms. The algorithms are classified into two categories, machine learning and non-machine learning, and offloading strategies, such as total and partial offloading, are discussed. Authors of [55] analyzed the role and challenges of Artificial Intelligence and Machine Learning algorithms for resource management in fog/edge computing environments.

References [56] examined significant journal publications on mobile computing and how the research route, challenges, and methodologies have evolved. Also, the architecture and essential components of edge computing are described. Reference [57] overviews offloading strategies in a fog environment. The contributions they cover are articles published on or before November 2020 and do not contain the most recent research in fog computing.

A general overview of the goals, optimization methods, algorithms relevant to task offloading in fog computing, and formulation of its mathematical issue is covered in [58]. The article [35] examines the factors influencing whether to utilize the cloud or the fog for offloading while discussing various offloading techniques in the cloud-IoT ecosystem. The authors also discuss other offloading-supporting technologies, such as wireless connection, virtualization, and AI. Reference [59] reviews how Reinforcement Learning (RL) and Deep Reinforcement Learning (DRL) algorithms deal with offloading problems in fog computing. Value-based, policy-based, and hybrid-based algorithms are the three classes into which it divides the fog computing offloading methods.

In contrast to prior review studies, we focused on task management from different perspectives in this study, including network problems and SDN technologies. Table 1 compares this study with other relevant studies in this field.

This review article focuses on the importance of networks and SDN in fog computing, emphasizing how they improve resource management, scalability, and compute efficiency in centralized and distributed computing environments. As a cloud computing extension, fog computing moves storage and processing closer to the network's edge, requiring reliable and adaptable network management systems. In this context, SDN's function is essential since it offers dynamic network design and optimization, which can adjust to fog computing environments' different and sometimes erratic requirements.

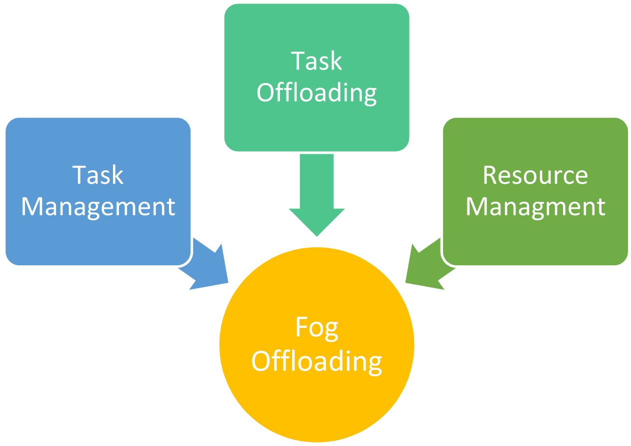

FIGURE 3. The system of interconnected components that constitutes fog offloading.

The detailed examination of these issues by this study not only offers a complete knowledge of the problems and solutions in this field today but also opens the door for future advancements in SDN application and network management, both essential for developing fog computing technologies.

IV. FOG OFFLOADING RELATED APPROACHES

The paradigm for fog computing has extremely low latency. Due to fog computing scalability and improved reactivity, it has caught the interest of numerous academics [14], [35], [46], [55], [59], [60]. Several publications in the fog computing sector were investigated in the literature review. These articles are classified and compared into the following areas of fog offloading based on the kind of algorithms and system architectures.

a) Resource management
b) Task management
c) Task Offloading

The interconnected components comprising the fog computing architecture are shown in Figure 3, and relevant articles will be discussed in the following sections.

Before proceeding to the articles connected to the debate, we evaluate the available surveys on this issue and emphasize the significance of this study in this part.

A. RESOURCE MANAGEMENT

The fog network consists of various heterogeneous devices that may perform computations, store data, and exchange messages. Due to the ever-changing nature of fog networks and the mobility of their constituent devices, resource management is a significant challenge. Considering the importance of effectively managing resources, this might enhance fog network performance and make the most of fog's processing capability. Since then, various attempts to improve fog networks' resource management have been assessed.

In [30], a fog-to-fog cooperation paradigm encourages fog nodes to offload incoming requests based on their load and processing capacity. The mathematical model of fog-to-fog collaboration that achieves near-optimal task distribution

TABLE 1. Related works.

Survey	Year	Area			Focus									Remarks
		Fog	Edge	Cloud	Task Offloading	Task management	Resource management	Metrics	Algorithm types	AI Discussion	Network Architecture	SDN Details	Simulation Tools	
[54]	2023		✓		✓		✓	✓	✓	✓			✓	Focus on Edge computing
[56]	2023		✓		✓			✓	✓	✓		✓	✓	Focus on mobile Edge computing
[55]	2023	✓	✓	✓	✓	✓	✓		✓	✓				Did not discuss network issues.
[57]	2022	✓			✓			✓	✓					Did not discuss network issues.
[58]	2022	✓			✓			✓	✓	✓	✓	✓		Just short refer to network architecture and SDN application in fog offloading
[59]	2023	✓			✓			✓		✓		✓	✓	Focus on Machine learning usage and just generally mentioned which articles use SDN or not.
[35]	2018	✓	✓	✓	✓			✓		✓				Didn't discuss network issues.
This survey	2023	✓			✓	✓	✓	✓	✓	✓	✓	✓	✓	Network and SDN discussion.

across cooperating fog nodes. The Fog Resource Management Scheme (FRAMES) encourages load balancing to address the latency issue with service requests received from objects. It is built on the fog as a service concept, which implies that each fog node is self-contained in processing, networking, and storage. The study uses a formal mathematical model that underlies fog node load balancing.

The DLAEC algorithm, introduced in reference [61] aims to enhance service quality through the utilization of deep learning, taking into account the availability of human resources and network capacity for each edge system. The DLAEC method maximizes resource utilization at the edge while guaranteeing the concurrent execution of the maximum number of deep learning tasks via a three-step decision procedure. DLAEC independently evaluates edge access and calculates the correct number of deep learning layers for identification at both the edge and cloud nodes, in contrast to previous models that allocate a set number of deep learning layers to edge systems. By assigning the majority of deep learning tasks to edge nodes, this method lessens the requirement for cloud transfers in an Edge Computing-based intelligent city environment, reducing network congestion and the load on the cloud. Nevertheless, instead of minimizing processing time delays, its primary objective is to maximize the capacity of the edge nodes. Consequently, the maximization may result in delays, and additional tasks may be accumulated on a single node before being redistributed to other nodes.

Blaster is a federated structure for routing packets within a dispersed edge network to improve application performance and data-intensive application scalability [62]. It also includes a revolutionary path selection algorithm that predicts the best route using Long Short-Term Memory (LSTM). This method employs a Federated Learning (FL) model to increase communication across SDN controllers while retaining data flow capacity. Choose the optimal route during system construction by using the results of an LSTM model, which is a kind of recurrent neural network that uses regression as a tunable technique. It is common practice to use the LSTM approach to deep learning-based time series forecasting problems. It provides a traffic matrix and a network graph to the LSTM. In a peak load prediction, the SDN controller or application may modify route selection based on the output, which is the estimated future load on the input connections. As a consequence, the state of the network influences route selection.

Reference [63] deal with implementing SDN to fog networks, namely P4/P4Runtime. According to the industry, P4 as a data plan programming language and P4Runtime as a control-to-data plan interface may assist address the demands of next-generation networks. It presents a unique technique for deploying SDN control plans capable of handling fog network SDN data plans and SDN data plans integrated near the main network. In this manner, SDN controllers may handle cross-layer SDN data plans, enabling certain operations to be offloaded to the Edge.

Reference [64] creates Android applications that serve as intermediaries between IoT devices and Edge/Fog/Cloud Computing ecosystems is made more. These modules facilitate connectivity with numerous devices functioning as data sources, and they can seamlessly integrate with various Fog/Cloud frameworks, making them readily adaptable for diverse applications. This concept is confined to one Android application and face limitation to be implemented in any system.

A mechanism for data allocation in IoT systems was proposed in [65] using the Blockchain. A data controller based on fuzzy logic aids in data allocation decisions. FogBus used blockchain-based cloud and fog technologies to improve a healthcare case study. The latency of data transfers, network utilization, energy consumption, and blockchain storage have all decreased, but security mechanism delays persist.

In [66], a paradigm based on Edge Affinity was proposed and developed to manage applications in processioning fog infrastructures. This paradigm organizes applications so that its resource-aware approach employs a larger number of fog processes to run applications with heavy data loads and rigorous resource needs. It addresses difficulties with application data flow and bandwidth. The drawback of this strategy is that it is dependent on a static fog management server to perform.

In [67], a fog computing model be utilized to produce mobility assistance advice for visually impaired people. This device combines a mobile phone with a low-cost, low-power embedded board and a neural computing stick, giving rise to the acronym PEN (Phone + Embedded Board + Neural Computing Stick). These three devices work together to provide numerous distributed capabilities by combining fog computing with cloud computing. The model's shortcoming is that varied hardware devices impact the system's mobility and adaptability.

Reference [68] present SOSW in Mobile Edge Computing (MEC) by leveraging the network traffic matrix. To determine the ideal placements for fog nodes, the SOSW model combines column-pivoting linear algebra techniques with singular-value decomposition (SVD) and QR factoring. With the aid of ant colony optimization (ACO), SOSW provides the constraint-based shortest path algorithm (CSPA), a heuristic-based traffic engineering solution that optimizes route calculation for task offload. The architecture of fog and its grid to its location within the network are the main topics of this research. The placement of fog to IoT devices in offloading data to fog in less time is also useful in decreasing energy consumption. Selects the route by which the IoT device may upload its data to the fog by collecting data through SDN and the recommended method for transmission.

Research [69] measuring connection quality over time as the average failure, downtime, or repair time. To evaluate the reliability of connections, use a k-nearest (k-NNR) ML approach based on regression trained on a 5-month real-life network data set. The k-NNR considers connection parameters such as the rate of link utilization, link failure, and the frequency of data set failures. Furthermore, calculate the connection delay as the sum of the transfer delay package size.

Research [70] discusses the internet of vehicular in MEC by utilizing SDN to propose a resource allocation and communication enhancement technique. It leverages q-learning to enhance the allocation of communication and computation resources by anticipating offload choices. This design's shortcoming is that it ignores a number of environmental factors that are necessary for the proper functioning of the Internet of Vehicles system. In order to determine the best offloading choice, communication, computing resource allocation, and privacy protection design, it applies a deep reinforcement learning method in SDN networks with an emphasis on edge computing-based computing offloading and resource allocation. Real-time business from terminal vehicles is sent straight into MEC's processing equipment, and MEC and SDN are integrated to provide centralized resource management and flexible network control.

Study [9] has proposed an adaptive computing optimization for resource management in industrial IoT using SDN and edge computing. It describes a strategy for effectively calculating the resources of neighboring devices and edge nodes for task processing: priority-based transmission with the least energy.

In [71] a user revocation system with an efficient architecture for a multi-user cloud environment that maintains data integrity is provided. It is efficient at reducing overhead when compared to current models in the cloud environment, but as user numbers rise, it increases the duration it takes to create signatures and proofs.

Using CCTV cameras for surveillance and event monitoring or on-board cameras for traffic monitoring, an application may produce enormous volumes of data, especially when video stream processing is involved field [72]. The primary contribution of [20] is the development of dynamic workload placement methods for latency-constrained stream processing requests. The primary concentration is on the arrangement of operators and the allocation of resources in a distributed environment that utilizes cloud and fog computing. When there is a network capacity shortage, provisioning of computing infrastructure uses the interaction between fog and cloud. The algorithms' goal is to optimize the proportion of successfully handled requests while adhering to application delay requirements by efficiently using the resources at hand.

To balance quality of service and energy consumption in large-scale networks, [73] introduces the Set-based Differential Evolutionary algorithm for energy-efficient resource management in IoT-based smart cities, employing SDN and fog computing.

In [74], each fog node in the resource allocation approach for fog computing based on SDN is defined by its processing capability and bandwidth (communication capacity). The method begins with picking an admin node and then mapping the nodes in the task graph to the fog node depending on the capacity of the fog node. Following this mapping, the allotted resources are taken from the fog node's capacity, which is maintained by a management module. However, this approach begins by virtualizing all of the nodes and scheduling them to one fog node. It then optimizes the schedule by relocating the nodes to other fog nodes.

B. TASK MANAGEMENT

By leveraging fog in processing information related to IoT devices in many sectors, such as health, agriculture, transportation, media, and other sections of intelligent cities, various tasks are assigned to the fog for analysis and processing. Several tasks are available, ranging from little data gathered by sensors to extensive data and films taken by cameras, as well as location and navigation for analysis and processing. Most of these data and devices need immediate reactions and real-time data processing via fog. Consequently, effectively managing diverse tasks is crucial for tasks that must be completed before a deadline. Several studies have developed approaches and algorithms for task management in fog networks, some of which are addressed here.

In order to improve quality in hybrid cloud settings for data-intensive applications running in a centralized shared file system, Tuli et al. suggested making use of dynamic resources and task scheduling. For data-intensive applications, it assesses data file type, data transfer time, network speed, and data proximity while planning and dynamically supplying public cloud resources. By lowering the quantity of shared file transfers across nodes, this technique may improve service quality and decrease task placement in a hybrid cloud environment. This approach was designed for cloud computing and is currently incompatible with edge and fog computing [75].

A lightweight distributed solution to self-organizing surveillance and monitoring fog networks is offered by [76]. FogMon analyses the hardware resources (CPU, RAM, HDD) of the access nodes, the end-to-end network quality of service (latency and bandwidth) and detects available IoT devices. It does not take storage or bandwidth into account when selecting leader nodes.

Reference [77] propose a privacy-aware task scheduling approach for a Blockchain-based fog network. The Ant Colony Optimization (ACO) approach optimizes work scheduling. It keeps end-user devices anonymous and reduces work processing time on the cloud's VRs.

To reduce application latency, consider demands from edge scheduling [78]. Network latency and service time for live video streaming services are significantly decreased by a score-based edge service scheduling approach that examines the network, computing, and reliability aspects of edge nodes. The flaw is that it restricts the centralized optimization process.

A fuzzy evolutionary organizer is provided for the multi-target resource allocation in a fog environment [79]. Since task processing time, information about inter-task communication, and task deadlines are all represented as fuzzy values; the technique achieves both the agreement index and robustness aims. The proposed approach has a significant temporal complexity.

Machine learning is employed in the mobile edge processing environment by [80] or distributed task scheduling algorithms and distributed device coordination. This model uses the Stackelberg game theory and the distributed ADMM optimization approach. This approach can achieve quick and ideal convergence regarding intelligent edge processing. The design has a significant issue in that it only impacts one MEC server, and the scheduling strategy may change when there are several MEC servers.

In [81], a paradigm for safe data storage and processing at the edge and in the cloud was presented. For privacy, this approach encrypts data during transit. In this architecture, tasks are broken into tiny portions and transmitted between edge nodes for execution; if a node does not have adequate processing capacity, the task migrates to its side nodes, and so on, until a suitable node is identified and wholly performed. In this system, the node is only dispatched for processing based on its proximity, independent of its resources or capacity. It may fail and be compelled to move to other nodes. This data transfer and undefined movement might cause substantial data latency issues when system demands increase.

C. TASK OFFLOADING

Since there is a variety of fog in the network, it is vital to identify the destination fog for offloading when fog is overloaded, or IoT devices demand an external processor. So, the destination selection choice influences the amount of processing time, data delay, and deadline. Conversely, the destination fog's reliability in completing the requisite processing with no errors is another priority offloading operation. Factors like having adequate processing power in the destination fog, enough memory, and a sufficient and acceptable connection to send data may all be utilized to establish the fog's competence. The right strategy for gathering fog information in the network and selecting the optimal fog is vital at this stage. Several approaches have been developed to acquire fog information in the network, which may be categorized into two categories: centralized and distributed. Most centralized algorithms have incorporated SDN technology because of the advantages of SDN networks.

1) DECENTRALIZE TASK OFFLOADING

As discussed in last section, some fog offloading methods are based on distributed architecture or created without SDN technologies. The matching theory-based distributed computing offloading framework MATO, utilized for heterogeneous fog nodes to minimize task execution latency, was introduced in the research field [82]. Reference [14] enhanced the framework for distributed processing on a fog network based on Akka. The Akka toolkit is one of the most comprehensive and well-liked actor modeling solutions for Java Virtual Machine (JVM). In the Fog IoV network, [37] provide a task offloading strategy employing the roadside parked cars as a computing offload location.

The struggle between IoT devices for fog nodes is treated as a game in Field [83]. The Weighted Potential Game's finite improvement characteristic demonstrates the Nash equilibrium.

DecChain is a secure edge computing solution based on blockchain that removes the requirement for a trusted third party on the network [84]. Concerns such as eliminating a trusted network failure point when losing access to a third party have been addressed. It faces challenges, such as integrating blockchain into the processing environment at the edge. Edge servers or service providers manage tasks in the network. As can be seen, service providers handle tasks based on a predefined structure, and the capacity of edge nodes to split tasks is not considered.

Reference [85] concentrate on multi-hop vehicular systems to have optimum options for performing activities locally or remotely. For compute offloading, efficient route selection and fog node assignment may reduce average service latency and energy consumption. Its performance is inferior to other frameworks when there are less than 50 automobiles, but by increasing the numbers, performance is more outstanding than others.

In 5G network technologies based on edge computing, research [86] combines optimization techniques for task offloading and resource allocation by controlling the energy consumption of the system. The plan may provide even better latency performance, which boosts the functionality of 5G mobile communication networks and enhances end-user satisfaction. The suggested approach ignores the order in which computing tasks are completed and only considers the overall processing time delay of all tasks inside the system.

An Optimal Stopping Theory (OST) inspired paradigm for data quality-aware task offloading in mobile edge computing is presented in [87]. The OST-based method is acceptable when mobile nodes make small, independent choices inside the MEC environment and do not need many resources. Each offloading mobile node will run the models in a single configuration without considering the context of other mobile nodes.

The structure and task offloading based on neural network service are illustrated using a GPU-based embedded edge server [88]. An offloading mechanism is presented based on the computing gap between the edge and the central cloud. It is particularly costly to build since it depends on GPU resources at the network's edge.

In [22], when one node on the network becomes overburdened, it may upload all or part of its responsibilities to other fogs through a cooperative mechanism. This algorithm improves fog dispersal based on multipliers distributed alternating direction approach. This model focuses on optimizing the Quality of Experience (QoE) and increasing the power efficiency of fog nodes and the tradeoff between these two measures.

Reference [23] provides a multi-layered fog computing system for dynamic resource allocation and offloading utilizing traffic prediction. It defined the problem as a stochastic network optimization problem and offered a solution that reduces power consumption while maintaining queue stability. The shortcoming of the concept is that power usage and overall queue backlog sizes would become inefficient if forecast mistakes happened.

For the concurrent task data offloading, [36] suggests collaborating horizontally with several fog nodes and vertically with a faraway cloud. It took into account the latency in communication between end users and the fog, as well as the time it took to send data and provide services. It also factored in local computation time and waiting times at the fog node linked to queuing.

The authors [89] put their attention on offloading duties from a device to fog and fog-to-fog. The ideal scheduling approach for the two queues, such as low and high priority in a fog node, is examined in this model. The stability of both queues is concurrently maintained while more offloaded tasks that are aware of deadlines may be finished. Two techniques comprise the recommended approach: The Lyapunov drift-plus-penalty method and fog computing collaboration are used to construct a priority-aware scheduling approach based on the fog nodes' queue status. According to the simulation findings, with the same resource setup, this strategy may ensure that more tasks are finished within the allotted time limit.

2) CENTRALIZED TASK OFFLOADING

Centralized task offloading is split into two groups: those that use a SDN and those that don't.

a: SDN-BASED TASK OFFLOADING

In this part, we investigate and evaluate fog offloading strategies based on SDN technology. Most of these techniques are developed for data offloading from IoT or mobile devices to fog.

For an SDN-based fog computing system, a dynamic offloading service between fog nodes is suggested by [2]. Selecting the suitable offloading node and enabling the offloading path by providing an end-to-end bandwidth guarantee are the goals of using SDN technology. To choose an appropriate offloading node, the proposed method makes use of fog node information and real-time network status. Furthermore, a few network parameters may be used to ensure bandwidth throughout the offloading path and create an optimal end-to-end path selection route.

Q-learning offloading choice allows the controller to choose appropriate actions depending on the reward function [29]. In this strategy, the controller may customize their incentive function based on the desired performance. Consequently, the suggested Q-learning-based offloading choice can estimate how good the current offloading will be in the future, resulting in remarkable overall system performance. Furthermore, the suggested solution is compatible with SDN architecture as the reinforcement learning-based decision-making process is on-demand, which allows the controller to construct a reward function depending on the required performance.

By using an integer linear programing (ILP), [90] resolves the multi-hop task offloading issue. To solve the problem effectively, use a greedy heuristic-based approach since the viable set is non-convex. Latency, energy costs, multi-hop routes, and fluctuating network factors like link utilization and SDN rule capacity are all part of the greedy approach. The contribution 1) the best decision is to compute a work locally or remotely; 2) the best fog node selection; 3) the best offloading route selection. With fewer fog devices, the recommended technique can attain excellent performance.

In [31], SDN approaches are used to optimize resource management and load balance across a network of Cloudlets. To balance the distribution of different tasks offloaded from mobile devices while optimizing resource utilization, it is handled as a mixed-integer linear programming (MILP) optimization model. Taking into account both communications and computation delay, available resources, and a task with a restricted deadline, the issue of a balanced distribution of incoming requests throughout the Cloudlet network is outlined. To emulate the recommended architecture, experiment with utilizing Mininet-WIFI and Floodlight as the SDN controller. The limitations of this method are the available resources and the demands of the consumers.

Reference [11] created a blend offloading structure to improve the offloading choice selection process by selecting the optimal offload node and assuring the infeasible request's needed deadline and the average processing cost of the fog node. It employs a mathematical approach known as Binary Linear Programming (BLP) to determine to offload destination fog. The fog node's SDN controller will choose the optimal offload target fog by comparing the needed deadline of the infeasible request with the response time of each available parked and moving car.

Reference [91] suggests a four-layer network paradigm based on SDN for the Industrial Internet of Things (IIoT). Depending on the requirements of different controllers, all possible computation offloading locations indirectly manage the production equipment to tasks at the closest power steering device. SDN controllers oversee data transmission routing, while lightweight containers such as Docker and Micro are utilized to offer computing.

Reference [92] provide algorithm in fog system resource management to ensure each task's specific service quality and optimize resource consumption by collaborating amongst fog computing nodes. Develop a common heterogeneous task download and resource method with Deep Recurrent Reinforcement Learning to optimize tasks within time restrictions. Depending on their buffer and resource state, the proposed method may offload their tasks into nearby nodes. Prevent unfairness in resource allocation slices with varied priorities, resulting in a higher average success rate.

Which cloudlet should be utilized to offload a particular task from mobile devices is determined by [93] based on the load assessment of cloudlets in multi-cloudlet networks. Furthermore, SDN is used by this system to load balance between cloudlets in wireless mobile networks. Additionally, it suggests an admission control mechanism to limit the acceptance of task assignment requests made if there are more requests than the network can handle.

Authors in [94] considered the multi-hop route and the impact of vehicle mobility while proposing a software-defined vehicle network offloading technique. It provided an ILP-based optimization problem for determining the ideal number of fog nodes for a given network in order to lower operating and capital costs.

The dynamic offloading for soft real-time workloads in an SDN-based fog architecture is investigated in research [95]. Preliminary results have emphasized the impact of intra- and inter-fog cluster analysis on the expense of missing the deadline.

Study [50] explores the offloading problem in IoV systems based on SDN and fog computing. It proposes an energy-aware dynamic offloading technique to extend the battery life of the IoV system and run more apps. The system model calculation and application transfer incur cost, and the heuristic searching optimization technique sometimes need assistance in order to find the best answer. However, testing on a real IoV edge network was not done in the experiment scenario.

The Programming Protocol-independent Packet Processors(P4) framework allows for the direct execution of resource and system functions on the programmable data plane, including in the newly introduced P4. This functionality enables the shifting of specific tasks from the controller to specialized hardware, such as P4 switches. With P4, programmers can specify packet behavior in the data plane [13]. To optimize the efficiency of task-offloading solutions and reduce computational overhead and latency in IoT networks, [13] propose a P4-assisted task-offloading scheme for fog-based IoT networks. However, if the fog server has the necessary resources, the offloaded task from IoT device will be executed immediately; otherwise, the task will wait in the queue until those resources become available in the fog. As a result, the system becomes less effective, and task processing experiences slight delays.

b: CENTRALIZED TASK OFFLOADING (WITHOUT SDN)

Some research provides centralized architectures without using SDN. Authors in [96] provide Fogbus, a blockchain-based platform that integrates IoT, fog, edge, and cloud communications via user identification and data encryption. Duties are controlled by a fog called master fog, who distributes tasks to other fog nodes called workers. The mechanism chooses master fog and worker fog nodes at random. When confronted with enormous volumes of data and activities, system flexibility is reduced, and modifying and transmitting data from multiple nodes to identify the final processing node delays and system performance. This complex technique employs a rudimentary blockchain algorithm that the system administrator may activate or stop. As the blockchain is activated, the system latency rises immediately compared to the inactive state.

Reference [97] provides a cost-effective compute offloading architecture for use in industrial networks. It works based on a fog federation in which a master fog controller manages the flow of traffic and data from IIoT sensors to the various fog nodes. Furthermore, it hired a policy-based reinforcement learning approach using a Q-learning algorithm and a controller-based device adaption strategy to regulate emergency-based service requests effectively and route them toward the fog devices along the shortest path.

Reference [98] introduces a hierarchical paradigm for the IoT, fog, and the cloud. Distributed task execution may control global energy usage and enable highly scalable IoT applications. As a proof of concept, analyze the efficacy of a three-tier design by considering the processing needs of several IoT applications in medicine, multimedia, geolocation, and text. It evaluates three use cases employing real-world datasets: fog-only, cloud-only, and fog-cloud cooperative.

V. EVALUATION AND COMPARATIVE ANALYSIS

The research about the offloading fog given in the offloading section were examined and compared in this part. Table 2 provides an overview of the studied algorithms, including the technique employed, the usage of SDN and the kind of protocol utilized, the metrics employed the type of simulator used, area of algorithm, and their notable features and constraints. As shown in the examination of the articles, the majority of the papers and algorithms are concerned with IoT to fog offloading, and a few methods have been described in the offloading section from one fog to another fog.

Several things could be improved when looking at fog-to-fog algorithms utilizing SDN. Fog selection is a decision-making method with a high real-time processing requirement and is one of the main obstacles in the decision-making process and destination. On the other hand, reducing system efficiency by increasing the amount of fog and network traffic is a significant difficulty that most of these algorithm's face. In addition, the network data plan could be more helpful in choosing the connection for offloading to the destination. Additionally, connection traffic and data plans must be appropriately considered when choosing a fog, which might increase the latency in data transmission to the target fog and cause the offloading process to halt and fail if the link fails. The following is a summary of the comparative evaluation and notable shortcomings in current approaches:

A. FOG OFFLOADING ALGORITHM TYPES

Based on the reviewed research, we can categorize task offloading techniques as indicated in Figure 4 fog computing employs a variety of task-offloading strategies. Machine learning uses data-driven judgements to adapt to changing situations, but it may need much training data and computing power. When well-modelled, mathematical optimization gives near-optimal solutions but is challenging and less adaptable. Heuristic approaches are simple and efficient, but they are not ideal. Blockchain-based offloading improves trust and decentralization while potentially adding overhead. The technique of choosing is determined by application requirements and resource availability while balancing flexibility, complexity, and trust factors. To summarize, fog computing task offloading strategies each have their own benefits and drawbacks that suit different application needs. To achieve a balance between flexibility and cost, the decision should take into account the unique demands of the application as well as the available resources.

The combination of mathematical algorithms, AI algorithms, and blockchain technology with fog computing has resulted in varied and complex solutions for data and resource management. Each of these approaches has distinct advantages and constraints, making them crucial in the realm of fog computing.

Mathematical techniques are often used in fog offloading for various purposes, such as task offloading, load balancing, energy optimization, and delay reduction. Their focused and precise approach offers practical answers for optimization difficulties, such as minimizing energy consumption and distributing computing workloads evenly. Nevertheless, they only sometimes provide the most favorable outcomes and may need significant processing resources since their effectiveness relies greatly on the particularities of the situation and the design of the algorithm.

Although optimization and game theory based solutions are effective in some constrained contexts, matching-based techniques provide potential benefits due to their distributed nature and low computing cost methodology [82]. However, AI algorithms in fog computing are used for predictive analytics [97], dynamic decision-making [29], and adaptive resource management [92]. They possess exceptional flexibility and aptitude for learning, making them well-suited for dynamic and unexpected circumstances. The primary advantage of AI is its capacity to automate intricate decision-making procedures and predict forthcoming requirements. However, it also presents problems such as the need for extensive datasets, intricacy in training and implementing models, and sometimes, the opaqueness of decision-making processes.

Blockchain technology [30], a recent addition to the field of fog computing, provides a clear benefit in terms of security and the preservation of data integrity. Blockchain guarantees a high degree of security by facilitating transparent and distributed transactions and communications inside fog networks, according to its distributed and tamper-evident characteristics. This capability is essential in remote computing systems, where maintaining the accuracy and reliability of data is of utmost importance. Nevertheless, blockchain faces a lot of constraints. Scalability problems arise, especially when dealing with higher transaction volumes, and the maintenance of a blockchain network needs significant resources. Furthermore, the process of incorporating blockchain into current systems has its own distinct set of difficulties.

Ultimately, the selection of mathematical algorithms, AI, and blockchain in fog computing is determined by specific criteria such as security, flexibility, utilization of resources, and intricacy of the environment. Mathematical algorithms give precise and efficient solutions in certain situations, while AI brings flexibility and predictive capabilities. Additionally, blockchain ensures unmatched security and integrity in distributed systems. Frequently, using a hybrid strategy that capitalizes on the advantages of these varied technologies provides the most efficient answer in the complex realm of fog computing.

Fog-to-fog offloading difficulties and challenges need a study and enhancement of route selection and decision selection algorithms and the successful use of SDN-based network characteristics. Due to the advancement of artificial intelligence and machine learning in multiple sectors, many of the approaches that have been seen have turned to employ machine learning, deep learning, or AI algorithms in fog computing algorithms. Although these algorithms have significantly improved predictions for fog offloading, employing them presents several difficulties owing to their

TABLE 2. Summary of task offloading algorithms.

Article	Method	SDN	Metrics						Remarks	Simulation Environments	Area	Cloud offload
			Delay	Energy	Deadline	Transmission	Processing cost	Number of Tasks				
[2]	SDN based Algorithm for ranking hops	Yes	✓			✓			Linear algorithm for decision making and search hops in controller. The number of recourses of nearest selected nodes are lower than previous framework. By increasing the number of Fogs, the delay time increased and throughput came down	Mininet	Fog to fog	No
[14]	Establishing a neural network training environment for a fog network on top of an Akka-based distributed processing platform.	No	✓						Has relatively lower accuracy and lower latency compared to centralized model.	Real Testbed	Fog to fog	No
[37]	The offloaded tasks can be assigned to the best RSPVs using a mathematical model created by the CPLEX optimization software.	No			✓	✓	✓	✓	Fog act as offload coordinator among IoV and Roadside Parked Vehicles (RSPV)	-	Fog to fog	Yes
[82]	Distributed task offloading architecture based on matching theory	No	✓						To keep track of other neighbor fogs available resources, fog must regularly gather data from its surrounding neighborhoods. The expense of communicating through each fog is quite significant.	Simulator not mentioned	IoT to fog and Fog to Fog	Yes
[83]	Distributed task offloading algorithm from IoT devices to fogs using Game theory.	No	✓	✓		✓	✓		Have relatively higher average processing time compared to other baselines.	Python	IoT to fog	Yes
[95]	Real-time tasks being dynamically offloaded in an SDN-based fog architecture	Yes			✓			✓	focus on the consequences of the proposed intra- and inter-fog cluster offloading	-	Fog to fog	Yes
[13]	Employing P4 to assist with task offloading in Fog-based IoT networks.	Yes	✓					✓	When measured in accordance with various server load status control intervals, TOS-SDN performs better initially, but as the load status control interval grows, it eventually equals TOS-P4 in terms of average waiting times.	Real Testbed	IoT to fog	No
[20]	Placing operators and allocating resources in a distributed system employing cloud and fog computing	Yes				✓		✓	Resource needs and latency thresholds, which are aspects of stream processing requests, are not taken into consideration.	Mininet	Fog to cloud	Yes
[50]	Fog offloading using SDN in IoV systems.	Yes	✓	✓		✓	✓	✓	The ideal answer cannot always be found using a heuristic search optimization technique.	Real Testbed	IoV to fog	Yes
[29]	Use reinforcement learning technique that Q-learning was combined with the greedy algorithm	Yes	✓						Limit number of fog node.	Realistic network	Fog to fog	No
[90]	Greedy-heuristic-based strategy	Yes	✓	✓					Limit to IoT to fog and with fewer fog devices, the suggested strategy can obtain good performance	Mininet	IoT to fog	No
[68]	Employ column pivoting linear algebra with singular-value decomposition and QR factorization.	Yes	✓	✓				✓	Limit to static IoT device,	Mininet	IoT to fog (MEC)	No
[11]	Use mathematical model named Binary Linear Programming (BLP)	Yes	✓		✓		✓		Don't consider energy consumption	MATLAB	IoT to fog	Yes
[70]	Use Q-learning for allocation of communication and computing resources	Yes	✓			✓	✓		The various environmental variables of the Internet of Vehicles system's real functioning are not considered.	MATLAB	IoT to Edge (MEC)	No
[91]	For IoT scenario suggest a four-layer SDN-based network design.	Yes	✓	✓					Limit devices are including in the real network.	Real Network	IoT to fog	Yes
[69]	Link reliability may be improved by using machine learning (ML) and multi objective optimization (MOO) methods.	Yes				✓			Limit to link optimization	Mininet	IoT to fog	No
[92]	Combines recurrent layer with deep Q-learning.	Yes	✓						Simulated Network consist of 5 fog nodes only	Python-TensorFlow	IoT to fog	Yes
[93]	Load balancing between cloudlet by using SDN, also task assignment in wireless cloudlets network	Yes	✓						Limited to mobile devices task offloading in multi-cloudlet networks	Numerical simulations	IoT to fog	No
[94]	Use a greedy-heuristic approach	Yes	✓	✓			✓	✓	vehicle-to-vehicle communication are not considered, and fog determines statically.	python-based simulations	IoT to fog	No
[30]	Formulated a formal mathematical model of fog-to-fog.	No	✓					✓	depend on Web-based app, consider nodes are reachable always.	MATLAB	Fog-to-fog	No
[96]	Blockchain base algorithm	No	✓	✓		✓		✓	Depend on special architecture for implementation. Algorithm design is Limited to cloud PaaS Infrastructure.	Aneka	IoT to fog	Yes
[22]	A distributed optimization algorithm to share workload among fog nodes based on distributed alternating direction method of multipliers	No		✓					Complexity of algorithm for each fog	Not specified	Fog-to-fog	No
[23]	dynamic offloading and resource allocation in fog network with traffic prediction (stochastic network optimization)	No		✓					In the event of forecast mistakes there will be a decline in efficiency in both overall queue backlog sizes and power usage.	Simulation by Python	Fog-to-fog	Yes
[36]	The optimization issue expressed as a quadratically constrained quadratic programming problem that is solved via semidefinite relaxation.	No	✓			✓	✓		Low efficiency and high complexity	Monte Carlo and MATLAB	Fog-to-fog	Yes
[89]	Priority-aware scheduling strategy is defined using the Lyapunov drift-plus-penalty algorithm, and fog cooperation under the task scheduling strategy.	No				✓	✓		Limited to task scheduling in fog	Monte Carlo simulations	Fog-to-fog	No
[97]	Master fog controller uses a probabilistic method to select the most capable fog device for offloading and uses the Q learning to determine the effective route to reach the target fog.	No	✓	✓			✓	✓	Depend on central fog controller that in terms of unavailability of that the system can't operate.	Python	IoT to fog	No
[98]	Mathematical model with three layers to process tasks arriving from IoT and determine to offload them to a fog or cloud,	No	✓	✓					Global Gateway act as broker to distribute task among IoT, fog and clouds. Failure in Global Gateway may make all the system process down.	SFogSim	IoT to fog and IoT to cloud	Yes
[73]	Addresses leveraging SDN for energy-efficient fog resource management.	Yes		✓					Neglect to handle security issues related to virtual machine migrations, particularly when sensitive data and vital services are at stake in smart city environments.	CloudSim and iFogSim	Fog to fog	Yes
[74]	Task offloading in SDN fog environment	Yes	✓				✓		The depth first search (DFS) technique for pathfinding among fog nodes lacks optimality in finding the shortest route and may require excessive memory in deep or infinite branches.	Python	Fog to fog	Yes

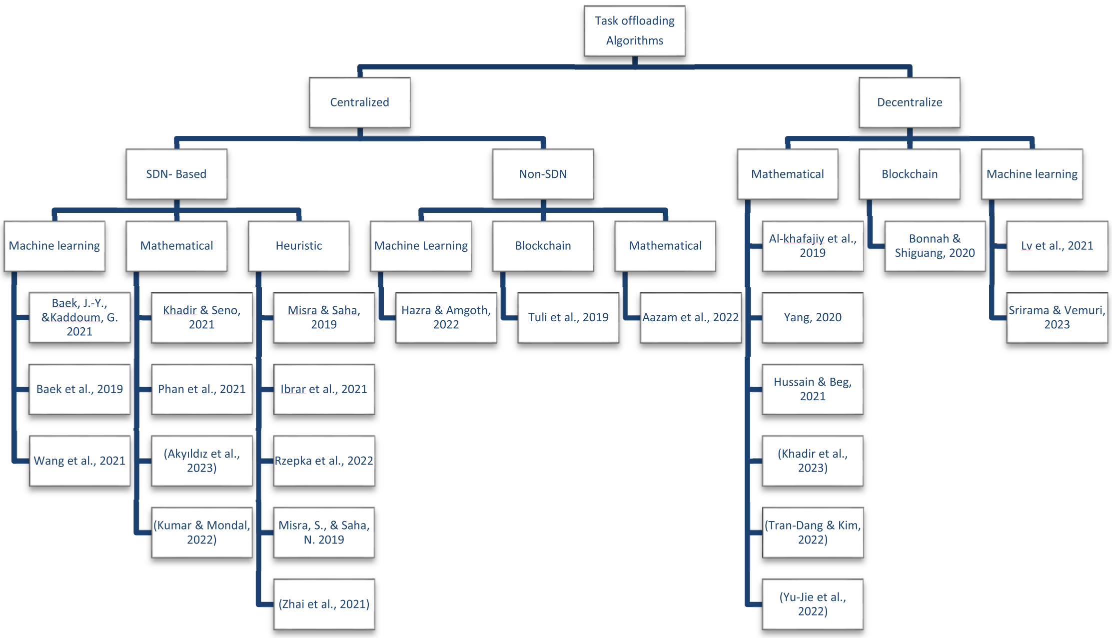

FIGURE 4. Task offloading algorithm types.

high computational cost and the limited processing power in the network's fog. Challenges like:

Computing resources are plentiful for artificial intelligence, machine learning, and deep learning algorithms, while computing resources are scarce in fog. On the other hand, the amount of communication load and bandwidth required to transmit data between devices for fog calculations adds to the difficulty of fog calculations.

B. FOG COMPUTING ARCHITECTURE

The effects of centralized, distributed, and SDN techniques on fog computing algorithms differ. Algorithms can optimize task offloading in a centralized system with a global view of resources and network circumstances, resulting in theoretically optimum solutions but presenting a single point of failure and scalability difficulties. Distributed techniques disperse decision-making across fog nodes, depending on local data and cooperation, making them more robust but less globally optimum. SDN-based designs provide dynamic network management, improving algorithm flexibility and responsiveness to real-time network changes, but their efficacy depends on algorithm integration and network programming. Task offloading centralized and distributed schemes features comparison are shown in Table 3. As can be seen, centralized networks, particularly SDN, provide several benefits despite their limitations and low flexibility. Centralized management using controllers in the SDN network allows for more simple controlling and managing of the computing resources of various fogs in the network with high capabilities and low loads on fog nodes.

In dynamic fog environments, where node availability, network circumstances, and workload vary, many of the current techniques may not be able to adapt properly. Inflexible strategies may not function as well as they could under changing circumstances, which might affect the system's overall performance.

TABLE 3. Task offloading networking schemes comparision.

Distributed Task offloading:	Centralized Task offloading	
	Without SDN	SDN- based
1)Decentralize	1)Centralize	1)Centralize
2)Nodes collect fogs information's.	2)Server collect fogs information's.	2)SDN controller collect fogs information's.
3)Hard to manage.	3)Manage by Server.	3)Manage by Controller.
4)Mixed data and control plane.	4)Mixed data and control plane.	4)Separate data and control plane.
5)Need coordination between fogs to prevent failures.	5)Server manages fogs.	5)Controller control the networks and fogs.
6)Fog connection and detections challenge.	6)Fog connection monitor by server.	6)Fog connection monitor by SDN controller.
7)Task offloading needs discussion between fogs.	7)Server coordinate task offloading between fogs.	7)Controller coordinate task offloading between fogs.
8)Task offloading process is time consuming.	8)Task offloading process accordance to server making decision time.	8)Task offloading process in lower time.
9)Task offloading algorithm add load on nodes.	9)Task offloading algorithm add load on server.	9)Task offloading algorithm add load on controllers.
10)Hard to manage tasks in dynamic network.	10)Can manage task in dynamic situation.	10)Can manage task in dynamic situation.
11)Task offloading limited to neighbor's fog.	11)Task offloading to all network's fogs.	11)Task offloading to all network's fogs.

Mitigating the negative impacts of a highly dynamic nature, such as the lack of fog device availability and end device mobility support, is an essential concern. SDN is a potential network architecture for fog computing because of its centralized, intelligent view and control of the network [95]. The following is an overview of the benefits and limitations of SDN architecture for fog offloading, including SDN protocols discussion:

1) SDN ADVANTAGES

SDN enables the immediate acquisition of network state, allowing for real-time centralized network management based on current network status and user-defined rules. Furthermore, this results in advantages in optimizing network setups and enhancing network performance [99], which is crucial for fog networks to offload tasks immediately without delay.

Configuration is crucial in network administration, particularly when incorporating new equipment into an established network. The difficulty stems from the diversity among network device manufacturers and their configuration interfaces, resulting in laborious and error-prone manual setup procedures. These errors need extensive troubleshooting efforts. SDN solves this problem by consolidating the control plane across different network devices, enabling centralized and automated setup via software control. This not only streamlines network administration but also allows for adaptive optimization depending on current network circumstances. The use of SDN in network configuration greatly affects fog computing networks by improving their capacity to quickly and easily manage and adjust fog devices to enable new applications and services.

In light of the present network condition and demand, flow rules may be simply adjusted dynamically and optimally using SDN [100].

SDN significantly impacts mathematical algorithms [2], [11], [13], [20], [31], [50], [68], [90], [91], [95] by streamlining network resource management in fog computing, enhancing load balancing, task offloading, and latency optimization. The dynamic allocation of resources empowers these algorithms to optimize distribution based on real-time data and preset criteria. Additionally, SDN's adaptability allows swift adjustments, bolstering algorithmic agility amidst network changes. In the realm of AI algorithms [20], [29], [69], [70], [92], SDN facilitates efficient data handling and routing crucial for decision-making processes. Its adaptability enables real-time resource optimization, augmenting AI system performance and scalability across diverse networks. Moreover, in blockchain integration [30], SDN aligns with the secure, distributed nature of transactions, enhancing network security and optimizing resource allocation. It also fosters improved interoperability among fog computing nodes, facilitating seamless integration of blockchain systems.

2) SDN CHALLENGES

Previous research has encountered many obstacles with conventional SDN-based handover, which leads to increased latency and packet losses because of centralized control. The increase in mobility cars puts a load on the main SDN controller, making it difficult to fulfill quality of service requirements and causing frequent handover problems. Further limiting their efficacy in high-mobility circumstances requires comprehensive simulation evaluations. In order to close this gap, article [10] presents a Vehicular ad-hoc networks (VANET) architecture that places changeover procedures at the edge of the network by merging SDN and fog computing technologies. This adaptable solution eases the load on core networks by decentralizing changeover management using zone SDN controllers and fog computing vehicles while meeting the needs of highly mobile and data-intensive services.

Incorporating SDN into fog computing poses significant obstacles that need resolution. These tasks include guaranteeing the capacity to handle many fog devices and adaptability, improving security and privacy within a centralized control system, establishing compatibility across different technologies, and optimizing resource management to distribute the workload efficiently. Additionally, addressing the critical challenges of optimizing traffic flows while minimizing latency, enhancing the energy efficiency of SDN controllers and fog nodes, ensuring consistent Quality of Service and reliability in dynamic fog environments, and seamlessly integrating SDN into current network infrastructures is essential.

The number of SDN controllers used in reviewed research is limited, so multi-controller scenario challenges need to be considered and discussed in massive networks. It is crucial to address these shortcomings to ensure the effective adoption of SDN in fog computing. This will need ongoing innovation and cooperation in the industry.

3) SDN PROTOCOLS

The OpenFlow [29], [50] protocol, which is well recognized in the field of Software-Defined Networking (SDN), provides extensive support and compatibility. However, it is limited in terms of flexibility because of its rigid architecture. Also it also has some security and scalability concerns. P4 [13], a more recent and adaptable protocol, enables the customization of packet processing. However, it is more complex and lacks widespread compatibility. In addition to these, other protocols such as NetConf and YANG provide network configuration management by providing structured data models and transactions. However, they may not possess the extensive programmability of P4 or the broad acceptance of OpenFlow. Each protocol fulfils distinct network needs, with OpenFlow being characterized by a higher level of standardization, P4 providing enhanced control capabilities, and NetConf or YANG concentrating on configuration management.

The selection of either OpenFlow or P4 for fog offloading relies on the particular demands of the network. OpenFlow's standardization and compatibility render it well-suited for contexts that priorities interoperability and stability. On the other hand, the ability of P4 to define packet processing behaviors is beneficial in situations that need customized and specialized data handling. Hence, if the task of offloading fog requires packet processing and flexibility that are highly specialized, P4 would be the more suitable choice. However, for broader and standardized implementations, OpenFlow may be the preferred option.

The particular needs and attributes of the fog computing environment will determine which SDN protocol is best for fog offloading. While Netconf is well-known for its simplicity and ease of use, making it a suitable match for easy network administration, OpenFlow is recognized for its fine-grained control and flexibility, making it excellent for dynamic and complicated network management. A network's complexity, scalability, required functionality and accessible knowledge are among the variables that determine which SDN protocol is best. Other protocols, including BGP-SDN, ONOS, and Faucet each offer advantages and disadvantages. Depending on their particular requirements, certain fog computing architectures may even combine these protocols to handle different aspects of task offloading and network management.

4) SDN WITH MACHINE LEARNING

The integration of SDN with Machine Learning may provide several advantages for Fog offloading. SDN offers a centralized perspective of the network, enabling the optimization of the decision-making process for selecting the most suitable Fog node for offloading. Machine learning may be used to create compute offloading strategies that enhance the efficiency and dependability of Fog computing. The advantages of integrating Software-Defined Networking (SDN) with Machine Learning (ML) for Fog offloading include enhanced utilization of storage and computing resources, improved performance metrics like latency, energy consumption, and Quality of Service, and the ability to dynamically allocate services based on objectives such as power consumption, security, and QoS constraints. Nevertheless, it is essential to take into account the drawbacks as well. The system's complexity may result in elevated maintenance expenses. Acquiring substantial data to train machine learning models may be a significant obstacle. The confidentiality and integrity of the data being processed by the Fog nodes may be compromised. To address these challenges, employing lightweight methodologies such as federated learning might be effective in fog offloading.

C. METRICS FOR VALIDATION

During fog offloading, route selection, latency, and energy metrics are crucial, yet not all research studies encompass these elements based on Table 2. Factors like problem complexity, limited resources, or research focus contribute to this omission. Some studies prioritize energy efficiency [50], [73], [83], [98], overlooking metrics like latency or route selection, while majority concentrate solely on delay optimization, disregarding energy concerns or routes. Some researchers highlighted path selection like [2], [69], [74], [85], [97]. Due to experimental constraints or study scope, researchers sometimes explore only a subset of these metrics.

However, disregarding any of these measures can significantly impact system performance. Neglecting energy efficiency may deplete IoT and Fog device battery life while overlooking latency can degrade user experience and system performance. Ignoring route selection may lead to poor data transfer and heightened network congestion. Therefore, considering all criteria remains paramount for optimal system performance in constructing an efficient fog offloading system.

In prior frameworks, time issues are only partially explored, and the computing power of resources needs to be better used. The computational strain increases the overall delay time, deployment cost, and energy consumption of IoT devices or resource-enriched fog nodes, requiring an efficient task management strategy to address this issue. Real-time processing is hampered by the high latency and system overload caused by the AI techniques utilized in these systems.

D. SIMULATION LIMITATION

Most current articles have undergone testing and evaluation in simulated contexts, with only methods [13], [14], [50] applied in actual or near-real situations. Furthermore, various kinds of simulators have been employed, each with its own merits and drawbacks. The wide range of simulators and the intricate nature of different network topologies, particularly SDN, will significantly challenge the effectiveness of the suggested algorithms in the real world.

While simulations carried out using simulators such as iFogSim and Mininet and Matlab, provide insightful results, real-world validation is essential. From simulation to real-world application, there might be unanticipated difficulties or new factors to take into account. Although there are many theoretical ideas, there may not be as much opportunity for these techniques' practical application or real-world assessment. Real-world implementation and thorough testing under various conditions are essential to verify its efficacy and functionality of proposed algorithms.

E. DATASET

A primary obstacle is the need for uniformity in the datasets used for modelling and executing fog-offloading studies and publications. This makes it challenging to generalize the data and compare the outcomes of various investigations [101]. More real-world data is needed to support the simulation conclusions. This may result in exaggerated expectations and erroneous forecasts [101]. Additionally, the particular use case and the surrounding environment significantly impact fog computing systems' performance. As a result, it's critical to carefully choose the simulation tools and datasets suitable for the particular use case [102].

The majority of studies use randomized data or fail to identify the application as data and tasks that are delegated to other fog nodes. Nevertheless, a limited number of researches, such as [14], have used real datasets. Consequently, the outcomes of most of experiments may not be applicable to addressing real-world applications that need low latency or real-time execution.

The absence of appropriate datasets and a simulation environment that accurately reflects the intricate nature of the fog space is a significant challenge.

F. SCALABILITY AND THROUGHPUT

As the number of nodes in the network increases, the system's efficiency is significantly diminished. The issue of managing massive networks of fog will provide a significant obstacle to the system. In the technique described in reference [2], the maximum number of fogs is limited to 16, and only a single SDN controller is used. Just 5 fogs used in [29], [90] and in real testbed fog node limited to 2 fog in [50], 6 fogs in [13]. Other algorithms have also been presented with more fogs, but only a limited number of fogs have been tried in small spaces and networks. It has been observed that as the number of fogs increases, there is a considerable loss in efficiency.

The use of more fog nodes in the experiments resulted in a reduction in the overall waiting time for tasks [13]. Delay or waiting time is one of the main metrics of related researches that is in front of the efficiency of systems, so should be considered.

Scalability is a significant challenge, particularly in systems with many devices and fogs. High overhead approaches may need help scaling efficiently, affecting their viability in real-world applications. This can be true of communication overhead, compute overhead, or resource allocation. It is crucial to thoroughly evaluate the scalability of the suggested algorithms, particularly in practical smart city implementations, where the quantity of devices and fog resources may be much greater.

G. SECURITY AND PRIVACY

Maintaining data security and privacy throughout offloading procedures is an important yet difficult component. Many current techniques may disregard strong security safeguards, putting sensitive data at risk. The [12]merges fog computing, SDN, and blockchain to create a security framework for IoT in agriculture. By combining blockchain technology with SDN controllers, this design emphasizes safe IoT communication. This increases security and dependability, particularly for devices with limited resources.

If the fog nodes in smart healthcare systems lack robust security protocols, there is a risk that malevolent users may be able to steal users' private data. In addition, fog computing must address emerging issues, like resource limited IoT devices and insider assaults. In order to address these difficulties, [7] suggests implementing a secure authentication system for fog nodes in intelligent healthcare based on SDN. The system involves the implementation of an authentication algorithm in the SDN gateway to verify the credibility of the fog node. The IoT devices only need to transmit their privacy and functional properties to the SDN gateway to reduce the computational burden on the IoT devices.

In the context of fog offloading, where the network environment is constantly changing, it is crucial to prioritize secure communication, authentication, and authorization. Nonetheless, these aspects have been overlooked in the majority of previous research and have not been taken into account. Software-Defined Networking (SDN) enables more effective management of these issues via its centralized control structure. The centralized method enables the application of security rules and network settings in a dynamic and flexible manner, capable of adjusting to evolving network circumstances and threats. SDN facilitates enhanced network visibility, a critical factor in detecting and addressing security breaches and anomalies. SDN facilitates the implementation of encryption, firewalls, and intrusion detection systems by unifying control.

Conversely, in the absence of SDN, maintaining security in fog offloading settings necessitates the use of decentralized methods. This involves the implementation of distributed firewalls and intrusion detection systems at several network nodes. Effective communication in such situations depends significantly on effective peer-to-peer authentication techniques and resilient encryption mechanisms. Periodic updates and manual adjustments are essential to align security rules with the changing network environment. This technique requires more coordination among various network components and may exhibit less adaptability in rapidly evolving circumstances as compared to SDN-enabled systems. However, it enables a decentralized approach to security, which might be advantageous in situations where centralized control is impractical.

A complete security architecture that includes secure communication, dynamic authentication, strong authorization, network segmentation, monitoring, blockchain integration, frequent updates, and user education is required due to the changing fog offloading scenario. Every aspect plays a vital role in strengthening the fog environment while adjusting to its constantly changing circumstance.

H. OTHER CHALENGES

As indicated in Table 2, most current studies do not incorporate cloud offloading in their simulations and implementations. This omission can lead to delays in many scenarios due to the absence of a comprehensive and accurate integration of cloud services. If the cost and delay associated with cloud offloading are less than those of using other fog nodes, the efficiency of such algorithms in evaluating cloud offloading remains unaddressed. Consequently, when tasks are only offloaded to other fog nodes instead of the cloud, this may result in increased delays and costs, leading to reduced system efficiency.

Due to the dynamic nature of the fog environment, it comprises many fogs with distinct memory capacities and processing capabilities. Conversely, the ongoing tasks vary in size and have varying deadlines. The intelligent organization of these tasks is a challenge faced by the majority of systems and algorithms since it aims to maximize the efficiency and capacity used by the fog computing infrastructure. For instance, in a scenario where there are three distinct fogs with varying available capacities and three different sizes of tasks (small, medium, and large), the optimal approach would involve transferring larger tasks to a fog with higher capacity and medium tasks to a fog with a proper capacity, smaller tasks to a fog with a sufficient capacity. Suppose a fog with more computational capacity is allocated to a minor task for any reason like it first come. In that case, the computational capacity of other fogs may not be sufficient for the enormous task, resulting in a potential delay in completing the task owing to the absence of a suitable fog with enough capacity. While several algorithms for resource and task management merely include these factors, managing resources and tasks concurrently to optimize task offloading efficiency and load balancing is still a significant and unavoidable difficulty in large-scale systems and real-time tasks that are ignored in most researches.

One potential option to address this difficulty and enhance fog systems load balancing is integrating SDN with machine learning techniques. SDN may get the latest information on forthcoming tasks and fog resources. Machine learning can then use the knowledge stored in SDN to forecast future events and tasks and choose the most appropriate fog resource for offloading.

Most previous studies assumed that the task size and fog capacity were the same. However, instead of using actual tasks, they have used randomly generated data, which fails to accurately depict the present state of the fog network in terms of task transmission, task processing, and real-time response.

Addressing these gaps highlights the need for more flexible and comprehensive offloading approaches in the fog computing realm. Fog offloading solutions must improve with strategies that balance energy efficiency, scalability, security, flexibility, and latency in dynamic settings in a way that considers real-world implementation circumstances.

VI. DISCUSSION

When a fog is overwhelmed, tasks should be transferred to another fog since it cannot process them (known as fog-to-fog offloading). In real-time computing, the process of determining and choosing the optimal destination node with adequate processing capacity in the quickest possible time is critical. Furthermore, most models disregard heterogeneity in computer infrastructure.

In most present algorithms, when a fog gets overloaded, it asks that the central server introduce the destination node for offloading. Before determining which node to deploy, the central server evaluates and compares the available nodes. It takes a long time to request the central server, perform the decision algorithm, respond to the initial node, and ultimately upload data to the destination node.

Also, choosing the optimum method to convey data from the main fog to the target fog is difficult since the selected path may be delayed due to traffic congestion. Because of traffic congestion on a particular route, another fog node with an acceptable and sufficient bandwidth for data transmission may be used.

The issues in fog-to-fog offloading include making a judgment about picking fog with adequate resources in the shortest period and choosing a suitable way to transmit data.

In older frameworks, time issues are explored from a limited perspective, and computing resources could be used more effectively. Implementing an efficient task management strategy is required to address this issue since computational strain increases the overall delay time, deployment costs, and energy consumption of IoT devices or resource-enriched fog nodes. Real-time processing is hampered by the AI algorithms utilized in these techniques, which also introduce significant delay and overload into the system.

When looking into fog-to-fog algorithms utilizing SDN, many things could be improved. Fog selection is one of the main obstacles in the decision-making process and destination because it requires a decision-making algorithm that is time consuming and high cost in real-time processing. On the other hand, increasing the amount of network traffic and fog reduces system efficiency, which is a significant difficulty faced by the majority of these algorithms. The network data plan is not believed to successfully choose the connection for offloading to the destination. Additionally, connection traffic and data plans should be adequately taken into account when choosing a fog, which might cause data transmission to the target fog to be delayed. If the link fails, the offloading process will also halt and fail.

A summary of the methods used in task offloading schemes is provided in TABLE 4. Fog-to-fog offloading concerns and difficulties need a study and enhancement of decision selection and route selection algorithms with appropriate use of SDN-based networks' capabilities. To expedite and enhance the fog offloading algorithms, it effectively employs the characteristics and capabilities of SDN networks, and artificial intelligence is crucial. Due to the advancement of artificial intelligence and machine learning in multiple sectors, many of the diverse fog offloading techniques that have been seen have turned to employing AI. These algorithms present several difficulties owing to their high computational cost and the limited computing resources available in the network for fog, even though they have been highly successful in improving estimates for fog and clouds. Challenges like:

1. While computational resources are scarce in fog, they are ample for computing machine learning, deep learning, and artificial intelligence algorithms.

2. Contrastingly, the amount of bandwidth and communication burden required to transmit data between devices for offloading decisions makes fog offloading more challenging.

The term "link prediction-based" refers to a technique for maximizing traffic offloading that is based on link prediction. The link prediction-based strategy prioritizes traffic offloading by selecting relevant seed nodes for efficient data transmission while keeping quality of service (QoS) in mind. To optimize overall performance, it is vital to establish a balance between traffic dumping and minimizing time delay [103]. Link prediction methods can assist in determining the optimum link and node to offload.

In prior studies of resource management that can be addressed as future research, heterogeneity or homogeneity of fog nodes is an essential factor that keeps the same for different fogs. Another issue for future task management studies is partial or whole task offloading. The algorithms should divide and distribute tasks efficiently based on task size and priority. Depending on the fog processing constraints and task size, each task may offload partially or fully. Aside from

TABLE 4. Summary of methodology used in task offloading schemes.

Author	Offloading type			Architecture			Protocol	Objective			Algorithm Type	Prediction		Implementation	
	IoT to Fog	Fog-to-Fog	Fog to Cloud	Centralized (No SDN)	Distributed	SDN		Path selection	Fog Selection	Energy		Fog Selection	Path	Real Test bed	Simulation
[2]		✓				✓	OpenFlow	✓	✓		Mathematical (Exhaustive search)				✓
[14]	✓	✓			✓	-	-		✓		Machin Learning (Neural Network)			✓	
[20]	✓					✓	OpenFlow		✓		Mathematical (Heuristic)				✓
[37]	✓		✓		✓	-	-		✓		Mathematical (Optimization)				✓
[82]	✓	✓			✓	-	-		✓		Mathematical (Matching Theory)				✓
[95]		✓				✓	-		✓		Mathematical (optimization problem)				✓
[13]	✓					✓	P4		✓		Mathematical (optimization)			✓	
[50]	✓					✓	OpenFlow			✓	Mathematical (Heuristic)			✓	
[29]		✓				✓	OpenFlow		✓		Machin Learning (Q-Learning mixed greedy)	✓		✓	
[90]	✓					✓	OpenFlow		✓		Mathematical (Greedy-Heuristic)				✓
[31]	✓					✓	Not specified				Mathematical (Optimization)				✓
[68]	✓					✓	OpenFlow		✓		Mathematical(heuristic)				✓
[11]	✓					✓	OpenFlow		✓		Mathematical(optimization)				✓
[70]	✓					✓	Not specified		✓		Q-Learning				✓
[91]	✓					✓	OpenFlow		✓		Mathematical			✓	
[69]	✓					✓	OpenFlow	✓			Machine learning				✓
[92]	✓					✓	Not specified		✓		Q-Learning				✓
[93]	✓					✓	Not specified		✓		Mathematical				✓
[94]	✓					✓	Not specified		✓		Greedy (Heuristic)				✓
[30]		✓			✓	-	-		✓		Blockchain				✓
[96]	✓	✓		✓		-	-		✓		Mathematical				✓
[22]		✓			✓	-	-				Mathematical				✓
[83]	✓				✓	-	-		✓	✓	Mathematical (Weighted Potential Game)				✓
[23]		✓	✓		✓	-	-				Mathematical	✓			✓
[36]	✓	✓	✓		✓	-	-		✓		Mathematical				✓
[89]		✓			✓	-	-				Mathematical				✓
[85]	✓	-	-		✓	-	-	✓	✓		Mathematical			✓	
[97]	✓			✓		-	-	✓	✓	✓	Q-Learning		✓		✓
[98]	✓		✓	✓		-	-		✓	✓	Mathematical				✓
[73]		✓				✓	Not specified		✓	✓	Differential Evolutionary				✓
[74]		✓				✓	OpenFlow	✓	✓		Heuristic greedy algorithm				✓

time and delay, energy consumption and algorithm computation cost are essential metrics for fog offloading techniques that are often overlooked in many suggested algorithms and should be considered in future studies.

While SDN has many benefits, it also has certain drawbacks. Latency remains an issue since faraway fog nodes or data centers introduce inevitable delays. The complexity of SDN deployment, possible security issues from centralized management, and increased network overhead may impede adoption. Interoperability concerns might develop, especially in heterogeneous fog computing environments, and the initial expense of specialized hardware and software may dissuade some organizations from using SDN for fog offloading. SDN's applicability must be determined by carefully examining individual use cases and needs.

Fog computing simulation has a number of drawbacks, such as issues with model accuracy, scalability, the lack of real-time elements, overhead, abstraction, and dependence on behavioral assumptions. It may be difficult to effectively simulate the dynamic nature of fog computing with constantly changing workloads and network circumstances. Further impediments include incomplete network models, problems with validation and verification, and low predictive power. Large-scale and resource-intensive simulations may also be difficult to execute due to resource limitations. Although simulations provide insightful information, they should be utilized with caution, and the findings should be interpreted considering these limitations. Real-world testing and simulation combined may lead to a more comprehensive knowledge of fog computing systems.

VII. CONCLUSION AND FUTURE WORK

Most of the examined related fog offloading algorithms do not take a holistic approach to resource management, task management, and task offloading in the same system, instead focusing on one or two of these three critical components. Lake of this viewpoint will encounter several obstacles in implementing these algorithms in real-world complicated settings with various types of tasks and diverse fogs hardware and software with changing resources simultaneously. As a result, fog offloading systems need further development to be highly adaptable to IoT's diverse and extensive real-time task response.

SDN provides various advantages for fog offloading in computer systems. It enables network flexibility by enabling dynamic and programmable configurations to react to changing workloads. Centralized control streamlines decision-making for task offloading, optimizing routing, and enhancing service quality. SDN is a crucial tool for practical fog computing since it improves traffic optimization and scales to handle the rising number of IoT devices and fog nodes.

SDN can manage the fog network effectively. However, its centralized and unique architecture does increase the possibility of initial implementation expenses. Machine learning may be helpful for fog processing if it transfers little data across the network while using far fewer computing resources. Machine learning may determine the decision-making process in choosing the connection link for data transmission and the fog destinations for offload. Selecting an efficient machine learning algorithm that decreases the computational burden on the network with its few connections and computation resource will be a problem solver. Therefore, future studies aim to speed up and enhance the fog-to-fog algorithm by efficiently using the capabilities and characteristics of SDN networks and artificial intelligence.

Furthermore, the implementation of additional SDN controllers and Fog networks on a wider scale, together with conducting rigorous testing in real-world scenarios and networks, is viable. likewise, exploring the integration of SDN networks with both centralized and decentralized networks outside the realm of SDN may serve as a viable approach to accommodate and merge novel and more expansive networks and smart city challenges.

ACKNOWLEDGMENT

Any options, finding, and conclusion or recommendation expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Air Force.

REFERENCES

[1] C. Mouradian, D. Naboulsi, S. Yangui, R. H. Glitho, M. J. Morrow, and P. A. Polakos, "A comprehensive survey on fog computing: State-of-the-art and research challenges," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 1, pp. 416–464, 1st Quart., 2018, doi: 10.1109/COMST.2017.2771153.

[2] L.-A. Phan, D.-T. Nguyen, M. Lee, D.-H. Park, and T. Kim, "Dynamic fog-to-fog offloading in SDN-based fog computing systems," *Future Gener. Comput. Syst.*, vol. 117, pp. 486–497, Apr. 2021, doi: 10.1016/j.future.2020.12.021.

[3] D. Rahbari and M. Nickray, "Task offloading in mobile fog computing by classification and regression tree," *Peer-to-Peer Netw. Appl.*, vol. 13, no. 1, pp. 104–122, Jan. 2020, doi: 10.1007/s12083-019-00721-7.

[4] N. Mazumdar, A. Nag, and J. P. Singh, "Trust-based load-offloading protocol to reduce service delays in fog-computing-empowered IoT," *Comput. Electr. Eng.*, vol. 93, Jul. 2021, Art. no. 107223, doi: 10.1016/j.compeleceng.2021.107223.

[5] A. Hakiri, B. Sellami, P. Patil, P. Berthou, and A. Gokhale, "Managing wireless fog networks using software-defined networking," in *Proc. IEEE/ACS 14th Int. Conf. Comput. Syst. Appl. (AICCSA)*, Oct. 2017, pp. 1149–1156, doi: 10.1109/AICCSA.2017.9.

[6] I. Sarkar, M. Adhikari, N. Kumar, and S. Kumar, "A collaborative computational offloading strategy for latency-sensitive applications in fog networks," *IEEE Internet Things J.*, vol. 9, no. 6, pp. 4565–4572, Mar. 2022, doi: 10.1109/JIOT.2021.3104324.

[7] C. Ke, Z. Zhu, F. Xiao, Z. Huang, and Y. Meng, "SDN-based privacy and functional authentication scheme for fog nodes of smart healthcare," *IEEE Internet Things J.*, vol. 9, no. 18, pp. 17989–18001, Sep. 2022, doi: 10.1109/JIOT.2022.3161935.

[8] J. Wang and D. Li, "Adaptive computing optimization in software-defined network-based Industrial Internet of Things with fog computing," *Sensors*, vol. 18, no. 8, p. 2509, Aug. 2018, doi: 10.3390/s18082509.

[9] S. Chandramohan, M. Senthilkumaran, and M. Sivakumar, "Adaptive computing optimization for industrial IoT using SDN with edge computing," in *Proc. 6th Int. Conf. Comput. Methodol. Commun. (ICCMC)*, Mar. 2022, pp. 360–365, doi: 10.1109/ICCMC53470.2022.9754048.

[10] M. K. Murtadha and B. M. Mushgil, "Flexible handover solution for vehicular ad-hoc networks based on software defined networking and fog computing," *Int. J. Electr. Comput. Eng.*, vol. 13, no. 2, p. 1570, Apr. 2023, doi: 10.11591/ijece.v13i2.pp1570-1579.

[11] A. A. Khadir and S. A. H. Seno, "SDN-based offloading policy to reduce the delay in fog-vehicular networks," *Peer-to-Peer Netw. Appl.*, vol. 14, no. 3, pp. 1261–1275, May 2021, doi: 10.1007/s12083-020-01066-2.

[12] S. Padhy, M. Alowaidi, S. Dash, M. Alshehri, P. P. Malla, S. Routray, and H. Alhumyani, "AgriSecure: A fog computing-based security framework for agriculture 4.0 via blockchain," *Processes*, vol. 11, no. 3, p. 757, Mar. 2023, doi: 10.3390/pr11030757.

[13] O. Akyıldız, Í. Kök, F. Y. Okay, and S. Özdemir, "A P4-assisted task offloading scheme for fog networks: An intelligent transportation system scenario," *Internet Things*, vol. 22, Jul. 2023, Art. no. 100695, doi: 10.1016/j.iot.2023.100695.

[14] S. N. Srirama and D. Vemuri, "CANTO: An actor model-based distributed fog framework supporting neural networks training in IoT applications," *Comput. Commun.*, vol. 199, pp. 1–9, Feb. 2023, doi: 10.1016/j.comcom.2022.12.007.

[15] C. Stergiou, K. E. Psannis, B.-G. Kim, and B. Gupta, "Secure integration of IoT and cloud computing," *Future Gener. Comput. Syst.*, vol. 78, pp. 964–975, Jan. 2018, doi: 10.1016/j.future.2016.11.031.

[16] N. Koroniotis, N. Moustafa, and E. Sitnikova, "A new network forensic framework based on deep learning for Internet of Things networks: A particle deep framework," *Future Gener. Comput. Syst.*, vol. 110, pp. 91–106, Sep. 2020, doi: 10.1016/j.future.2020.03.042.

[17] C. Stergiou, K. E. Psannis, B. B. Gupta, and Y. Ishibashi, "Security, privacy & efficiency of sustainable cloud computing for big data & IoT," *Sustain. Comput., Informat. Syst.*, vol. 19, pp. 174–184, Sep. 2018, doi: 10.1016/j.suscom.2018.06.003.

[18] A. Botta, W. de Donato, V. Persico, and A. Pescapé, "Integration of cloud computing and Internet of Things: A survey," *Future Gener. Comput. Syst.*, vol. 56, pp. 684–700, Mar. 2016, doi: 10.1016/j.future.2015.09.021.

[19] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the Internet of Things," in *Proc. 1st, Ed., MCC Workshop Mobile Cloud Comput.*, Aug. 2012, pp. 13–15.

[20] M. Rzepka, P. Boryło, M. D. Assunção, A. Lasoń, and L. Lefèvre, "SDN-based fog and cloud interplay for stream processing," *Future Gener. Comput. Syst.*, vol. 131, pp. 1–17, Jun. 2022, doi: 10.1016/j.future.2022.01.006.

[21] R. Rapuzzi and M. Repetto, "Building situational awareness for network threats in fog/edge computing: Emerging paradigms beyond the security perimeter model," *Future Gener. Comput. Syst.*, vol. 85, pp. 235–249, Aug. 2018, doi: 10.1016/j.future.2018.04.007.

[22] Y. Xiao and M. Krunz, "QoE and power efficiency tradeoff for fog computing networks with fog node cooperation," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, May 2017, pp. 1–9.

[23] X. Gao, X. Huang, S. Bian, Z. Shao, and Y. Yang, "PORA: Predictive offloading and resource allocation in dynamic fog computing systems," *IEEE Internet Things J.*, vol. 7, no. 1, pp. 72–87, Jan. 2020, doi: 10.1109/JIOT.2019.2945066.

[24] S. Yi, Z. Hao, Z. Qin, and Q. Li, "Fog computing: Platform and applications," in *Proc. 3rd IEEE Workshop Hot Topics Web Syst. Technol. (HotWeb)*, Nov. 2015, pp. 73–78.

[25] V. C. Pujol and S. Dustdar, "Fog robotics—Understanding the research challenges," *IEEE Internet Comput.*, vol. 25, no. 5, pp. 10–17, Sep. 2021, doi: 10.1109/MIC.2021.3060963.

[26] I. Ahammad, M. A. R. Khan, and Z. U. Salehin, "QoS performance enhancement policy through combining fog and SDN," *Simul. Model. Pract. Theory*, vol. 109, May 2021, Art. no. 102292, doi: 10.1016/j.simpat.2021.102292.

[27] L. Yu, L. Chen, Z. Cai, H. Shen, Y. Liang, and Y. Pan, "Stochastic load balancing for virtual resource management in datacenters," *IEEE Trans. Cloud Comput.*, vol. 8, no. 2, pp. 459–472, Apr. 2020.

[28] T. H. Luan, L. Gao, Z. Li, Y. Xiang, G. We, and L. Sun, "A view of fog computing from networking perspective," 2016, *arXiv:1602.01509*.

[29] J.-Y. Baek, G. Kaddoum, S. Garg, K. Kaur, and V. Gravel, "Managing fog networks using reinforcement learning based load balancing algorithm," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2019, pp. 1–7, doi: 10.1109/WCNC.2019.8885745.

[30] M. Al-khafajiy, T. Baker, H. Al-Libawy, Z. Maamar, M. Aloqaily, and Y. Jararweh, "Improving fog computing performance via fog-2-fog collaboration," *Future Gener. Comput. Syst.*, vol. 100, pp. 266–280, Nov. 2019, doi: 10.1016/j.future.2019.05.015.

[31] S. Shahryari, S.-A. Hosseini-Seno, and F. Tashtarian, "An SDN based framework for maximizing throughput and balanced load distribution in a cloudlet network," *Future Gener. Comput. Syst.*, vol. 110, pp. 18–32, Sep. 2020, doi: 10.1016/j.future.2020.04.009.

[32] A. A. Khan, M. Abolhasan, W. Ni, J. Lipman, and A. Jamalipour, "A hybrid-fuzzy logic guided genetic algorithm (H-FLGA) approach for resource optimization in 5G VANETs," *IEEE Trans. Veh. Technol.*, vol. 68, no. 7, pp. 6964–6974, Jul. 2019, doi: 10.1109/TVT.2019.2915194.

[33] M. Polverini, J. Galán-Jiménez, F. G. Lavacca, A. Cianfrani, and V. Eramo, "Improving dynamic service function chaining classification in NFV/SDN networks through the offloading concept," *Comput. Netw.*, vol. 182, Dec. 2020, Art. no. 107480, doi: 10.1016/j.comnet.2020.107480.

[34] M. Sookhak, F. R. Yu, Y. He, H. Talebian, N. S. Safa, N. Zhao, M. K. Khan, and N. Kumar, "Fog vehicular computing: Augmentation of fog computing using vehicular cloud computing," *IEEE Veh. Technol. Mag.*, vol. 12, no. 3, pp. 55–64, Sep. 2017, doi: 10.1109/MVT.2017.2667499.

[35] M. Aazam, S. Zeadally, and K. A. Harras, "Offloading in fog computing for IoT: Review, enabling technologies, and research opportunities," *Future Gener. Comput. Syst.*, vol. 87, pp. 278–289, Oct. 2018, doi: 10.1016/j.future.2018.04.057.

[36] M. Mukherjee, S. Kumar, C. X. Mavromoustakis, G. Mastorakis, R. Matam, V. Kumar, and Q. Zhang, "Latency-driven parallel task data offloading in fog computing networks for industrial applications," *IEEE Trans. Ind. Informat.*, vol. 16, no. 9, pp. 6050–6058, Sep. 2020, doi: 10.1109/TII.2019.2957129.

[37] A. A. Khadir, S. A. H. Seno, B. F. Dhahir, and R. Budiarto, "Efficient-cost task offloading scheme in fog-Internet of Vehicle networks," *Comput. Syst. Sci. Eng.*, vol. 45, no. 2, pp. 2223–2234, 2023, doi: 10.32604/csse.2023.032316.

[38] A. Yousefpour, A. Patil, G. Ishigaki, I. Kim, X. Wang, H. C. Cankaya, Q. Zhang, W. Xie, and J. P. Jue, "QoS-aware dynamic fog service provisioning," 2018, *arXiv:1802.00800*.

[39] N. McKeown, "Software-defined networking," in *Proc. INFOCOM Keynote Talk*, vol. 17, Jan. 2009, pp. 30–32.

[40] S. Khorsandroo, A. G. Sánchez, A. S. Tosun, J. Arco, and R. Doriguzzi-Corin, "Hybrid SDN evolution: A comprehensive survey of the state-of-the-art," *Comput. Netw.*, vol. 192, Jun. 2021, Art. no. 107981, doi: 10.1016/j.comnet.2021.107981.

[41] P. Upadhyaya, O. Farooq, M. R. Abidi, and Y. V. Varshney, "Continuous Hindi speech recognition model based on kaldi ASR toolkit," in *Proc. Int. Conf. Wireless Commun., Signal Process. Netw. (WiSPNET)*, Mar. 2017, pp. 786–789.

[42] F. Y. Okay and S. Ozdemir, "Routing in fog-enabled IoT platforms: A survey and an SDN-based solution," *IEEE Internet Things J.*, vol. 5, no. 6, pp. 4871–4889, Dec. 2018, doi: 10.1109/JIOT.2018.2882781.

[43] A. A. Diro, H. T. Reda, and N. Chilamkurti, "Differential flow space allocation scheme in SDN based fog computing for IoT applications," *J. Ambient Intell. Humanized Comput.*, vol. 15, no. 2, pp. 1353–1363, Feb. 2024, doi: 10.1007/s12652-017-0677-z.

[44] M. H. Kashani and E. Mahdipour, "Load balancing algorithms in fog computing," *IEEE Trans. Services Comput.*, vol. 16, no. 2, pp. 1505–1521, Mar. 2023, doi: 10.1109/TSC.2022.3174475.

[45] Z. Arefian, M. R. Khayyambashi, and N. Movahhedinia, "Delay reduction in MTC using SDN based offloading in fog computing," *PLoS ONE*, vol. 18, no. 5, May 2023, Art. no. e0286483, doi: 10.1371/journal.pone.0286483.

[46] S. Askar, B. Hassan, and T. Rashid, Aug. 19, 2021, "SDN based fog computing: A review," *Zenodo*, doi: 10.5281/zenodo.5221036.

[47] S. S. Jazaeri, S. Jabbehdari, P. Asghari, and H. H. S. Javadi, "Edge computing in SDN-IoT networks: A systematic review of issues, challenges and solutions," *Cluster Comput.*, vol. 24, no. 4, pp. 3187–3228, Dec. 2021, doi: 10.1007/s10586-021-03311-6.

[48] J. Son and R. Buyya, "Priority-aware VM allocation and network bandwidth provisioning in software-defined networking (SDN)-enabled clouds," *IEEE Trans. Sustain. Comput.*, vol. 4, no. 1, pp. 17–28, Jan. 2019, doi: 10.1109/TSUSC.2018.2842074.

[49] S. Ahmad and A. H. Mir, "Scalability, consistency, reliability and security in SDN controllers: A survey of diverse SDN controllers," *J. Netw. Syst. Manage.*, vol. 29, no. 1, p. 9, Jan. 2021, doi: 10.1007/s10922-020-09575-4.

[50] Y. Zhai, W. Sun, J. Wu, L. Zhu, J. Shen, X. Du, and M. Guizani, "An energy aware offloading scheme for interdependent applications in software-defined IoV with fog computing architecture," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 6, pp. 3813–3823, Jun. 2021, doi: 10.1109/TITS.2020.3044177.

[51] L. Zhao, K. Yang, Z. Tan, X. Li, S. Sharma, and Z. Liu, "A novel cost optimization strategy for SDN-enabled UAV-assisted vehicular computation offloading," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 6, pp. 3664–3674, Jun. 2021, doi: 10.1109/TITS.2020.3024186.

[52] H. Hao, J. Zhang, and Q. Gu, "Optimal IoT service offloading with uncertainty in SDN-based mobile edge computing," *Mobile Netw. Appl.*, vol. 27, no. 6, pp. 2318–2327, Dec. 2022, doi: 10.1007/s11036-021-01796-4.

[53] X. Xu, Q. Huang, H. Zhu, S. Sharma, X. Zhang, L. Qi, and M. Z. A. Bhuiyan, "Secure service offloading for Internet of Vehicles in SDN-enabled mobile edge computing," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 6, pp. 3720–3729, Jun. 2021, doi: 10.1109/TITS.2020.3034197.

[54] A. Acheampong, Y. Zhang, X. Xu, and D. Appiah Kumah, "A review of the current task offloading algorithms, strategies and approach in edge computing systems," *Comput. Model. Eng. Sci.*, vol. 134, no. 1, pp. 35–88, 2023, doi: 10.32604/cmes.2022.021394.

[55] S. Iftikhar, S. S. Gill, C. Song, M. Xu, M. S. Aslanpour, A. N. Toosi, J. Du, H. Wu, S. Ghosh, D. Chowdhury, M. Golec, M. Kumar, A. M. Abdelmoniem, F. Cuadrado, B. Varghese, O. Rana, S. Dustdar, and S. Uhlig, "AI-based fog and edge computing: A systematic review, taxonomy and future directions," *Internet Things*, vol. 21, Apr. 2023, Art. no. 100674, doi: 10.1016/j.iot.2022.100674.

[56] M. Y. Akhlaqi and Z. B. M. Hanapi, "Task offloading paradigm in mobile edge computing-current issues, adopted approaches, and future directions," *J. Netw. Comput. Appl.*, vol. 212, Mar. 2023, Art. no. 103568, doi: 10.1016/j.jnca.2022.103568.

[57] M. S. Sofla, M. H. Kashani, E. Mahdipour, and R. F. Mirzaee, "Towards effective offloading mechanisms in fog computing," *Multimedia Tools Appl.*, vol. 81, no. 2, pp. 1997–2042, Jan. 2022, doi: 10.1007/s11042-021-11423-9.

[58] N. Kumari, A. Yadav, and P. K. Jana, "Task offloading in fog computing: A survey of algorithms and optimization techniques," *Comput. Netw.*, vol. 214, Sep. 2022, Art. no. 109137, doi: 10.1016/j.comnet.2022.109137.

[59] D. H. Abdulazeez and S. K. Askar, "Offloading mechanisms based on reinforcement learning and deep learning algorithms in the fog computing environment," *IEEE Access*, vol. 11, pp. 12555–12586, 2023, doi: 10.1109/ACCESS.2023.3241881.

[60] R. Buyya and S. N. Srirama, *Fog and Edge Computing: Principles and Paradigms*. Hoboken, NJ, USA: Wiley, 2019.

[61] K. Lee, B. N. Silva, and K. Han, "Algorithmic implementation of deep learning layer assignment in edge computing based smart city environment," *Comput. Electr. Eng.*, vol. 89, Jan. 2021, Art. no. 106909, doi: 10.1016/j.compeleceng.2020.106909.

[62] A. Sacco, F. Esposito, and G. Marchetto, "A federated learning approach to routing in challenged SDN-enabled edge networks," in *Proc. 6th IEEE Conf. Netw. Softwarization (NetSoft)*, Jun. 2020, pp. 150–154, doi: 10.1109/NetSoft48620.2020.9165506.

[63] E. O. Zaballa, D. Franco, M. Aguado, and M. Berger, "Next-generation SDN and fog computing: A new paradigm for SDN-based edge computing," in *Proc. Fog-IoT*, 2020, pp. 9:1–9:8.

[64] R. Mancini, S. Tuli, T. Cucinotta, and R. Buyya, "iGateLink: A gateway library for linking IoT, edge, fog, and cloud computing environments," in *Intelligent and Cloud Computing* (Smart Modelling for Engineering Systems). Singapore: Springer, 2021, pp. 11–19, doi: 10.1007/978-981-15-5971-6_2.

[65] W. Yánez, R. Mahmud, R. Bahsoon, Y. Zhang, and R. Buyya, "Data allocation mechanism for Internet-of-Things systems with blockchain," *IEEE Internet Things J.*, vol. 7, no. 4, pp. 3509–3522, Apr. 2020, doi: 10.1109/JIOT.2020.2972776.

[66] R. Mahmud, K. Ramamohanarao, and R. Buyya, "Edge affinity-based management of applications in fog computing environments," presented at the 12th IEEE/ACM Int. Conf. Utility Cloud Comput., Dec. 2019.

[67] J. Zhu, J. Hu, M. Zhang, Y. Chen, and S. Bi, "A fog computing model for implementing motion guide to visually impaired," *Simul. Model. Pract. Theory*, vol. 101, May 2020, Art. no. 102015, doi: 10.1016/j.simpat.2019.102015.

[68] M. Ibrar, L. Wang, G.-M. Muntean, N. Shah, A. Akbar, and K. I. Qureshi, "SOSW: Scalable and optimal nearsighted location selection for fog node deployment and routing in SDN-based wireless networks for IoT systems," *Ann. Telecommun.*, vol. 76, pp. 331–341, Apr. 2021, doi: 10.1007/s12243-021-00845-z.

[69] A. Akbar, M. Ibrar, M. A. Jan, A. K. Bashir, and L. Wang, "SDN-enabled adaptive and reliable communication in IoT-fog environment using machine learning and multiobjective optimization," *IEEE Internet Things J.*, vol. 8, no. 5, pp. 3057–3065, Mar. 2021.

[70] K. Wang, X. Wang, and X. Liu, "A high reliable computing offloading strategy using deep reinforcement learning for IoVs in edge computing," *J. Grid Comput.*, vol. 19, no. 2, p. 15, Jun. 2021, doi: 10.1007/s10723-021-09542-6.

[71] K. Ambika and M. B. Moses, "An efficient SG-DACM framework for data integrity with user revocation in role based multiuser cloud environment," *Comput. Commun.*, vol. 155, pp. 84–92, Apr. 2020, doi: 10.1016/j.comcom.2020.03.006.

[72] Y. Li, A.-C. Orgerie, I. Rodero, B. L. Amersho, M. Parashar, and J.-M. Menaud, "End-to-end energy models for edge cloud-based IoT platforms: Application to data stream analysis in IoT," *Future Gener. Comput. Syst.*, vol. 87, pp. 667–678, Oct. 2018, doi: 10.1016/j.future.2017.12.048.

[73] S. Choudhury, B. Pradhan, S. A. J. Francis, and D. S. Roy, "An energy efficient fog level resource management scheme for software defined cities," *Sustain. Energy Technol. Assessments*, vol. 57, Jun. 2023, Art. no. 103289, doi: 10.1016/j.seta.2023.103289.

[74] S. S. Nejad, A. Khademzadeh, A. M. Rahmani, and A. Broumandnia, "Resource allocation for fog computing based on software-defined networks," *Int. J. Electr. Comput. Eng.*, vol. 13, no. 6, pp. 7099–7109, Dec. 2023, doi: 10.11591/ijece.v13i6.pp7099-7107.

[75] S. Tuli, R. Sandhu, and R. Buyya, "Shared data-aware dynamic resource provisioning and task scheduling for data intensive applications on hybrid clouds using Aneka," *Future Gener. Comput. Syst.*, vol. 106, pp. 595–606, May 2020, doi: 10.1016/j.future.2020.01.038.

[76] S. Forti, M. Gaglianese, and A. Brogi, "Lightweight self-organising distributed monitoring of fog infrastructures," *Future Gener. Comput. Syst.*, vol. 114, pp. 605–618, Jan. 2021, doi: 10.1016/j.future.2020.08.011.

[77] H. Baniata, A. Anaqreh, and A. Kertesz, "PF-BTS: A privacy-aware fog-enhanced blockchain-assisted task scheduling," *Inf. Process. Manage.*, vol. 58, no. 1, Jan. 2021, Art. no. 102393, doi: 10.1016/j.ipm.2020.102393.

[78] A. Aral, I. Brandic, R. B. Uriarte, R. De Nicola, and V. Scoca, "Addressing application latency requirements through edge scheduling," *J. Grid Comput.*, vol. 17, no. 4, pp. 677–698, Dec. 2019, doi: 10.1007/s10723-019-09493-z.

[79] C.-G. Wu, W. Li, L. Wang, and A. Y. Zomaya, "An evolutionary fuzzy scheduler for multi-objective resource allocation in fog computing," *Future Gener. Comput. Syst.*, vol. 117, pp. 498–509, Apr. 2021, doi: 10.1016/j.future.2020.12.019.

[80] Z. Lv, D. Chen, R. Lou, and Q. Wang, "Intelligent edge computing based on machine learning for smart city," *Future Gener. Comput. Syst.*, vol. 115, pp. 90–99, Feb. 2021, doi: 10.1016/j.future.2020.08.037.

[81] X. Chen, "A security integration model for private data of intelligent mobile communication based on edge computing," *Comput. Commun.*, vol. 162, pp. 204–211, Oct. 2020, doi: 10.1016/j.comcom.2020.08.026.

[82] H. Tran-Dang and D.-S. Kim, "A many-to-one matching based task offloading (MATO) scheme for fog computing-enabled IoT systems," in *Proc. Int. Conf. Adv. Technol. Commun. (ATC)*, Oct. 2022, pp. 239–244, doi: 10.1109/ATC55345.2022.9942992.

[83] S. Yu-Jie, W. Hui, and Z. Cheng-Xiang, "Balanced computing offloading for selfish IoT devices in fog computing," *IEEE Access*, vol. 10, pp. 30890–30898, 2022, doi: 10.1109/ACCESS.2022.3160198.

[84] E. Bonnah and J. Shiguang, "DecChain: A decentralized security approach in edge computing based on blockchain," *Future Gener. Comput. Syst.*, vol. 113, pp. 363–379, Dec. 2020, doi: 10.1016/j.future.2020.07.009.

[85] M. M. Hussain and M. M. S. Beg, "CODE-V: Multi-hop computation offloading in vehicular fog computing," *Future Gener. Comput. Syst.*, vol. 116, pp. 86–102, Mar. 2021, doi: 10.1016/j.future.2020.09.039.

[86] S. Yang, "A joint optimization scheme for task offloading and resource allocation based on edge computing in 5G communication networks," *Comput. Commun.*, vol. 160, pp. 759–768, Jul. 2020, doi: 10.1016/j.comcom.2020.07.008.

[87] I. Alghamdi, C. Anagnostopoulos, and D. P. Pezaros, "Data quality-aware task offloading in mobile edge computing: An optimal stopping theory approach," *Future Gener. Comput. Syst.*, vol. 117, pp. 462–479, Apr. 2021, doi: 10.1016/j.future.2020.12.017.

[88] J. Kim, S. Ullah, and D.-H. Kim, "GPU-based embedded edge server configuration and offloading for a neural network service," *J. Supercomput.*, vol. 77, no. 8, pp. 8593–8621, Aug. 2021, doi: 10.1007/s11227-021-03623-9.

[89] M. Mukherjee, M. Guo, J. Lloret, R. Iqbal, and Q. Zhang, "Deadline-aware fair scheduling for offloaded tasks in fog computing with inter-fog dependency," *IEEE Commun. Lett.*, vol. 24, no. 2, pp. 307–311, Feb. 2020, doi: 10.1109/LCOMM.2019.2957741.

[90] S. Misra and N. Saha, "Detour: Dynamic task offloading in software-defined fog for IoT applications," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 5, pp. 1159–1166, May 2019, doi: 10.1109/JSAC.2019.2906793.

[91] G. Li, S. Hua, L. Liu, X. Zheng, and H. Ma, "SDN based computation offloading for industrial Internet of Things," in *Proc. 16th Int. Conf. Mobility, Sens. Netw. (MSN)*, Dec. 2020, pp. 402–409.

[92] J. Baek and G. Kaddoum, "Heterogeneous task offloading and resource allocations via deep recurrent reinforcement learning in partial observable multifog networks," *IEEE Internet Things J.*, vol. 8, no. 2, pp. 1041–1056, Jan. 2021.

[93] G. S. S. Chalapathi, V. Chamola, C.-K. Tham, S. Gurunarayanan, and N. Ansari, "An optimal delay aware task assignment scheme for wireless SDN networked edge cloudlets," *Future Gener. Comput. Syst.*, vol. 102, pp. 862–875, Jan. 2020, doi: 10.1016/j.future.2019.09.003.

[94] S. Misra and S. Bera, "Soft-VAN: Mobility-aware task offloading in software-defined vehicular network," *IEEE Trans. Veh. Technol.*, vol. 69, no. 2, pp. 2071–2078, Feb. 2020, doi: 10.1109/TVT.2019.2958740.

[95] N. Kumar and A. Mondal, "Work in progress: Dynamic offloading of soft real-time tasks in SDN-based fog computing environment," in *Proc. Int. Conf. Embedded Softw. (EMSOFT)*, Oct. 2022, pp. 7–8, doi: 10.1109/EMSOFT55006.2022.00011.

[96] S. Tuli, R. Mahmud, S. Tuli, and R. Buyya, "FogBus: A blockchain-based lightweight framework for edge and fog computing," *J. Syst. Softw.*, vol. 154, pp. 22–36, Aug. 2019, doi: 10.1016/j.jss.2019.04.050.

[97] A. Hazra and T. Amgoth, "CeCO: Cost-efficient computation offloading of IoT applications in green industrial fog networks," *IEEE Trans. Ind. Informat.*, vol. 18, no. 9, pp. 6255–6263, Sep. 2022, doi: 10.1109/TII.2021.3130255.

[98] M. Aazam, S. U. Islam, S. T. Lone, and A. Abbas, "Cloud of Things (CoT): Cloud-fog-IoT task offloading for sustainable Internet of Things," *IEEE Trans. Sustain. Comput.*, vol. 7, no. 1, pp. 87–98, Jan. 2022, doi: 10.1109/TSUSC.2020.3028615.

[99] W. Xia, Y. Wen, C. H. Foh, D. Niyato, and H. Xie, "A survey on software-defined networking," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 1, pp. 27–51, 1st Quart., 2015, doi: 10.1109/COMST.2014.2330903.

[100] O. Salman, I. Elhajj, A. Chehab, and A. Kayssi, "IoT survey: An SDN and fog computing perspective," *Comput. Netw.*, vol. 143, pp. 221–246, Oct. 2018.

[101] S. Taheri-abed, A. M. E. Moghadam, and M. H. Rezvani, "Machine learning-based computation offloading in edge and fog: A systematic review," *Cluster Comput.*, vol. 26, no. 5, pp. 3113–3144, Oct. 2023, doi: 10.1007/s10586-023-04100-z.

[102] H. Tran-Dang and D.-S. Kim, "FRATO: Fog resource based adaptive task offloading for delay-minimizing IoT service provisioning," *IEEE Trans. Parallel Distrib. Syst.*, vol. 32, no. 10, pp. 2491–2508, Oct. 2021, doi: 10.1109/TPDS.2021.3067654.

[103] Y. Zhang, J. Li, Y. Li, D. Xu, M. Ahmed, and Y. Li, "Cellular traffic offloading via link prediction in opportunistic networks," *IEEE Access*, vol. 7, pp. 39244–39252, 2019, doi: 10.1109/ACCESS.2019.2891642.

MOHAMMAD REZA REZAEE is currently an Assistant Professor with the Department of Information Technology, Faculty of Computer Science, Kabul Education University (KEU), Kabul, Afghanistan. He is currently pursuing the Ph.D. degree in computer networking with the Department of Communication Technology and Network, Faculty of Computer Science and Information Technology, University Putra University Malaysia (UPM), Selangor, Malaysia. His research interests include autonomous vehicles, swarm intelligence, computer networks, cryptography, and cloud computing.

NOR ASILAH WATI ABDUL HAMID (Senior Member, IEEE) received the Ph.D. degree in computer science from The University of Adelaide, in 2008. From 2013 to 2015, she was a Visiting Scholar with the High-Performance Computing Laboratory, George Washington University, Washington, DC, USA. She is currently the Head of the Laboratory of Computational Sciences and Mathematical Physics, Institute for Mathematical Research, University Putra Malaysia (UPM), Selangor, Malaysia. She is also an Associate Professor with the Department of Communication Technology and Network, Faculty of Computer Science and Information Technology, UPM. She has authored or coauthored more than 80 journals and conference proceedings in her research area. Her work was funded by the government and some by industry. Her research interests include parallel and distributed high-performance, cloud, and data-intensive computing. She received the CUDA Teaching Center Award from NVIDIA, in 2015, and established the CUDA Laboratory. She serves as a reviewer for various renowned journals and conference proceedings.

MASNIDA HUSSIN (Senior Member, IEEE) received the Ph.D. degree from The University of Sydney, Australia, in 2012. She is currently an Associate Professor with the Department of Communication Technology and Network, Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, Malaysia. She was involved in several network projects, such as cloud computing, network cloudification, and green computing. She has published several numbers of articles related to parallel and distributed computing. Her research interest includes resource management, such as scheduling, allocation, and edge analytics for large-scale parallel and distributed computing. She has become an editorial board member/reviewer in several national and international journals.

ZURIATI AHMAD ZUKARNAIN (Member, IEEE) received the bachelor's and master's degrees in physics and education from Universiti Putra Malaysia (UPM), Selangor, Malaysia, in 1997 and 2000, respectively, and the Ph.D. degree in quantum computing and communication from the University of Bradford, U.K., in 2005. From 2006 to 2011, she was the Head of the Department of Communication Technology and Networks, Faculty of Computer Science and Information Technology, UPM, where she is currently a Professor. From 2012 to 2015, she was also the Head of the Section of High Performance Computing Services, Institute of Mathematical Research, UPM. Her research interests include computer networks, distributed systems, mobile and wireless, network security, quantum computing, and quantum cryptography.