

Review

# An Overview of Fog Data Analytics for IoT Applications

Jitendra Bhatia <sup>1</sup>, Kiran Italiya <sup>2</sup>, Kuldeepsinh Jadeja <sup>3</sup>, Malaram Kumhar <sup>1</sup> , Uttam Chauhan <sup>4</sup>,  
Sudeep Tanwar <sup>1,\*</sup> , Madhuri Bhavsar <sup>1</sup>, Ravi Sharma <sup>5</sup>, Daniela Lucia Manea <sup>6</sup>, Marina Verdes <sup>7,\*</sup>  
and Maria Simona Raboaca <sup>6,8</sup> 

<sup>1</sup> Department of Computer Science and Engineering, Institute of Technology, Nirma University, Ahmedabad 382481, India

<sup>2</sup> Amazon, Bangalore 560055, India

<sup>3</sup> School of Earth and Space Exploration, Arizona State University, Tempe, AZ 85287, USA

<sup>4</sup> Department of Computer Engineering, Vishwakarma Government Engineering College, Gujarat Technological University, Ahmedabad 382424, India

<sup>5</sup> Centre for Inter-Disciplinary Research and Innovation, University of Petroleum and Energy Studies, Dehradun 248001, India

<sup>6</sup> Faculty of Civil Engineering, Technical University of Cluj-Napoca, Constantin Daicovicu Street, No. 15, 400020 Cluj-Napoca, Romania

<sup>7</sup> Department of Building Services, Faculty of Civil Engineering and Building Services, Technical University of Gheorghe Asachi, 700050 Iasi, Romania

<sup>8</sup> National Research and Development Institute for Cryogenic and Isotopic Technologies—ICSI Rm. Valcea, Uzinei Street, No. 4, P.O. Box 7 Laureni, 240050 Ramnicu Valcea, Romania

\* Correspondence: sudeep.tanwar@nirmauni.ac.in (S.T.); marina.verdes@academic.tuiasi.ro (M.V.)

**Abstract:** With the rapid growth in the data and processing over the cloud, it has become easier to access those data. On the other hand, it poses many technical and security challenges to the users of those provisions. Fog computing makes these technical issues manageable to some extent. Fog computing is one of the promising solutions for handling the big data produced by the IoT, which are often security-critical and time-sensitive. Massive IoT data analytics by a fog computing structure is emerging and requires extensive research for more proficient knowledge and smart decisions. Though an advancement in big data analytics is taking place, it does not consider fog data analytics. However, there are many challenges, including heterogeneity, security, accessibility, resource sharing, network communication overhead, the real-time data processing of complex data, etc. This paper explores various research challenges and their solution using the next-generation fog data analytics and IoT networks. We also performed an experimental analysis based on fog computing and cloud architecture. The result shows that fog computing outperforms the cloud in terms of network utilization and latency. Finally, the paper is concluded with future trends.

**Keywords:** Internet of things; artificial intelligence; blockchain; fog computing; data analytics



**Citation:** Bhatia, J.; Italiya, K.; Jadeja, K.; Kumhar, M.; Chauhan, U.; Tanwar, S.; Bhavsar, M.; Sharma, R.; Manea, D.L.; Verdes, M.; et al. An Overview of Fog Data Analytics for IoT Applications. *Sensors* **2023**, *23*, 199. <https://doi.org/10.3390/s23010199>

Academic Editors: Weizhe Zhang and Ibrahim A. Elgendy

Received: 12 December 2022

Revised: 20 December 2022

Accepted: 21 December 2022

Published: 24 December 2022



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

The world is changing rapidly, and novel and disruptive paradigms are shaping our future. One such paradigm is the Internet of things (IoT), which was presented many years ago but is constantly evolving. Everything we use in our daily lives becomes more intelligent as sensors are integrated. Every device, such as radio frequency identification (RFID) tags, actuators, mobile phones, and so on, permeates our daily lives in both industrial and domiciliary fields [1]. It is projected that over 31 billion devices will be linked to the Internet by 2020, and this figure will rise further as 5G technology matures. The industry is estimated to be worth USD one trillion by 2025 [2].

IoT has now diffused into various domains that were never thought of earlier. These include healthcare, home automation, aviation, industrial automation, vehicular networks, data analytics, mobiles, smart cities, agriculture, and wearables, including watches, shoes, t-shirts, etc. [3]. However, the most prominent effect of IoT is on industrial automation,

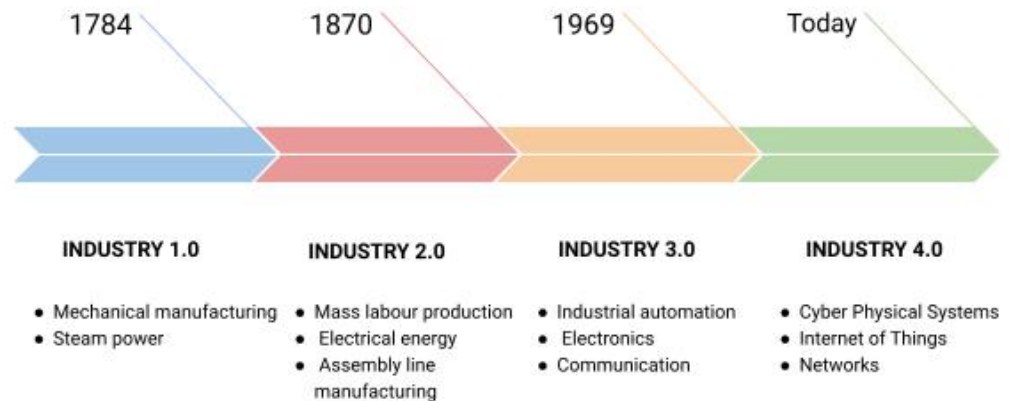
or Industry 4.0, where every device or machine is connected and communicates among themselves [4,5].

Human behavior is unpredictable and sometimes changes. Robotic systems should receive assistance from the cloud's backend, which collects data from wearables and sensors to decide the activities that robotic systems should carry out. A robotic arm utilized in many industrial applications is controlled using an IoT interface. By capturing the movement and performing the same activities repeatedly, it minimizes human effort [6]. IoT and robotic systems are closely collaborating as Industry 4.0 develops, reshaping their relationships and enabling the creation of next-generation gadgets. The Internet of robotic things (IoRT) is a notion that results from the combination of robotic agents and the Internet of things (IoT), and it opens up new opportunities for both industrial and academic disciplines. The fourth industrial revolution's primary outcome is creating and spreading cyber-physical systems (CPSs). Applications for CPS technology include gas distribution, transportation systems, medical equipment, and electrical power grids, among many others. In CPSs, networks interface with physical systems while conducting in-depth analysis and data extraction. IoT and CPSs provide a good framework for the growth of a new field of study called the Internet of robotic things (IoRT). In this scenario, IoRT represents the core component of robotics-integrated IoT systems, where cloud computing and networking can be implemented to accomplish elaborated tasks, enabling robots to share, network, and gather various types of information from humans and others machines [7,8].

### 1.1. Background

Before we deeply understand the IoT paradigm, let us first understand the industry's evolution through the various revolutions taking place at stipulated time intervals, each time having a novel, promising, and disruptive concept upon which that revolution has been based. Figure 1 shows the timeline of manufacturing, starting from the first industrial revolution and ending at the current state of Industry 4.0. We start with Industry 1.0, the first industrial revolution, which took place around the late 18th/early 19th century. Initially, human resources were involved in the production processes. Due to this, there was a high chance of errors, and the industries needed to be more scalable. During this revolution, the mechanization of production took place, with the introduction of steam and water-propelled engines into the industrial realm, which resulted in a tremendous increase in efficiency and scaling. Small-scale industries could now serve large companies and increase their customer base. The next revolution, Industry 2.0, took place in the 19th/early 20th century. The main factor behind the second revolution was introducing and developing machines running on electrical energy and assembly line production techniques for increasing efficiency and bringing modularization into effect so that divisions of work or labor based on skill sets could be made possible. The mass production of products began during this era. The production efficiency increased significantly due to a further decrease in human intervention as humans were now required more for monitoring and maintenance purposes and less for production. The Industry 3.0 revolution started in the late 20th century when electronic hardware such as transistors and integrated circuits were introduced to machines. These electronic devices needed software to operate them. The software also automated other tasks, including inventory management, optimum resource utilization, etc. The involvement of fully automated machines reduced the risk of errors and increased the production rate significantly. This era marked the era of computers and, lately, the Internet too.

The next and most recent industrial revolution, which is still ongoing, is the Industry 4.0 revolution, which thrives on making machines or devices "smart" [9]. It adds more value to the Industry 3.0 generation by adding communication between the connected devices/machines to minimize human involvement in production. Industry 4.0 is a byproduct of the IoT paradigm and focuses on real-time decision making by the devices based on various external and internal stimuli.



**Figure 1.** Industrial revolution.

### 1.2. Introduction to IoT and Fog Computing

As we have seen in the previous section, Industry 4.0 focuses on smart devices. The devices become smart by getting equipped with sensors or actuators that help them to identify their surroundings, process the data collected from them in the brain, make a decision, and then react according to the decision. This brings us to the IoT paradigm and the definition of IoT, which is somewhat as follows.

The network of all of these smart devices that are connected to sense, communicate, and interact within them and with the external environment on real-time data and make decisions by processing these data is collectively called the Internet of things or the IoT. All IoT devices are connected centrally to the cloud to compute the enormous amount of data that IoT devices collect [10,11]. The cloud also has a vast data storage capacity and is generally remotely located from the devices. Furthermore, the cloud provides different kinds of services, such as software as a service (SaaS), platform as a service (PaaS), and infrastructure as a service (IaaS), heading towards anything as a service (XaaS).

The vast number of interconnected devices drives out huge amounts of data in a raw format that needs to be processed, and decision making needs to be performed based on that data in real time. However, only one cloud is remotely situated and serves many devices, all of which need to respond quickly to the data. Consider the case of a driverless car that sends all of its sensors' data to the cloud to make decisions [12,13]. The car's camera sends out the image of the signal turning red just now, but as the decision to stop comes from the cloud, it has already crossed the signal, causing an accident. There are many such use cases in the real world where decisions need to be made instantaneously. At that time, it was clear that there were better things to carry out than sending all of the data directly to the cloud and waiting for the response to act. Therefore, a new paradigm is known as edge computing or fog computing. Although most people attribute both as the same thing, and it is not easy to distinguish between them, there is a subtle difference separating them. Other important paradigms similar to these have also been developed, such as dew computing or mobile cloud computing or a hybrid of any of these, which will be discussed later in the paper [14]. Fog computing aims at becoming a mediator between IoT devices and the cloud by cloning activities performed by the cloud but with lesser strength than the cloud itself. This process aims at bringing efficiency and scalability in two ways: vertical (by reducing the time taken to reach the cloud) and horizontal (fewer data needed to process and manage) [15,16]. However, the cloud now has a role. The high latency problem of the cloud can be overcome by using idle resources of various devices near users. However, fog computing still relies on the cloud for complex processing, which does not require real-time attention. The fog itself addresses the tasks that need such real-time attention. Its abstracted view is passed on to the cloud for complex data analytics, maintaining a global view of the interconnected devices and pushing local updates to the connected fog nodes [17].

### 1.3. Need for Fog Computing in IoT

We have gone through the paradigms of IoT and the introduction of fog computing into the IoT paradigm. Therefore, now, we will dive deep into the discussion as to why we should be using fog or edge computing in the IoT-driven world and why it is so disruptive [18]. We will first look at how our network would look with fog computing.

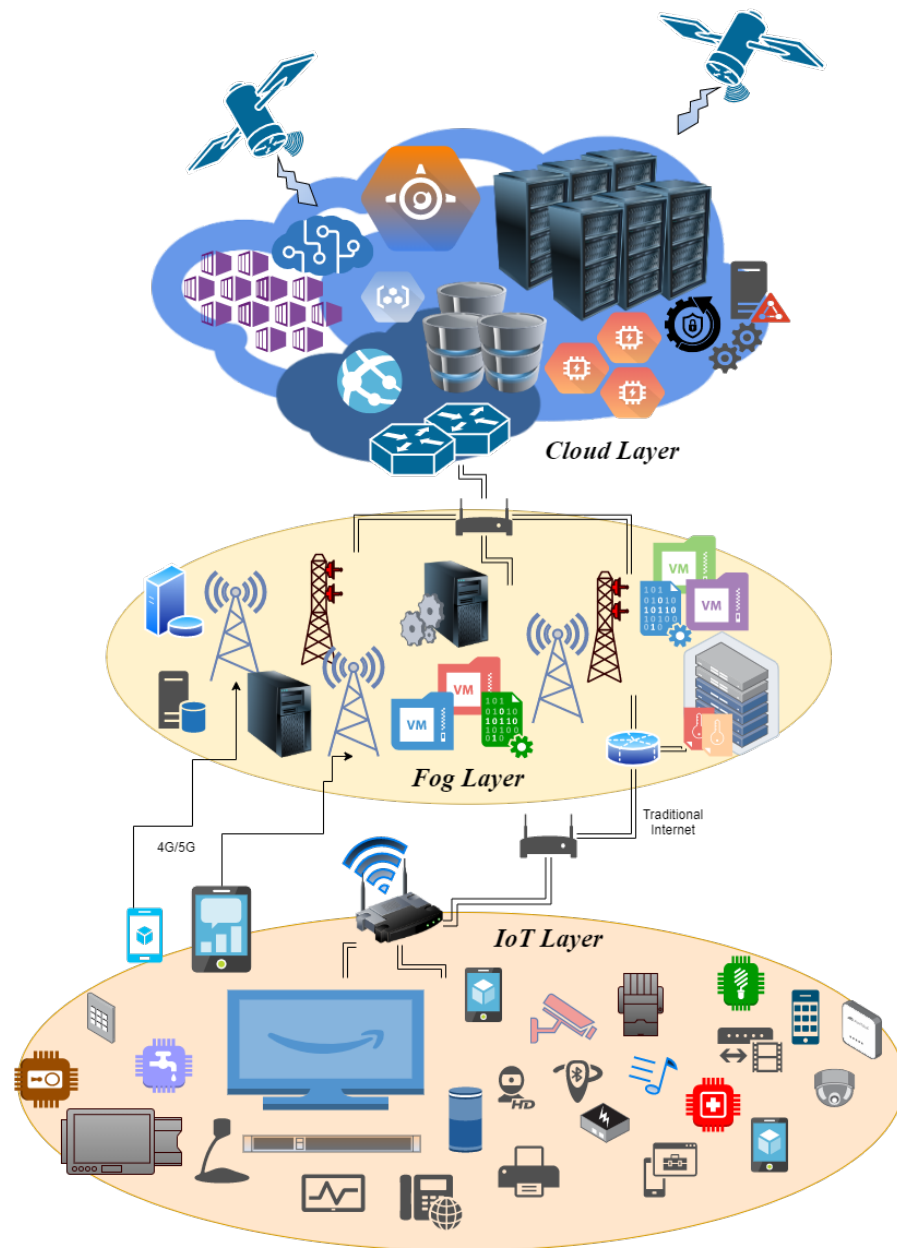
Cloud has more power, while fog has more agility, and agility will beat power when real-time processing and decision making is involved. With great power comes great responsibility, so the cloud has to make computations for all of the devices. In contrast, the fog has to make computations related only to locally connected devices. The devices connected mainly consist of sensors or actuators, which generate huge amounts of data every second. These data are also very raw, contain noise, and need to be pre-processed before some meaning can be derived. For example, it has been shown that a typical driverless car generates several megabytes of data every minute. This is where fog computing might be useful. Fog computing extends the already existing concept known as edge Computing. The edge computing paradigm states that computing and storage are performed directly on the end devices or near them. At the same time, fog computing addresses near computing but is separated from edge devices. The fog computing model is a distributed computing or decentralized model as opposed to the centralized model of the cloud [19]. Every fog node forms a layer of the interconnected network with one another next to the layer of interconnected IoT devices. The role of these fog nodes is not only to process data generated by the devices but also to transmit the decisions to the cloud via other fog nodes so that all of the fog nodes can have a common view of the system as a whole [20].

## 2. Integration of Fog Computing and IoT

The traditional cloud computing paradigm cannot resolve the low latency and faster response needs of time-critical applications, and that is where fog provides a solution to these problems. The fog computing paradigm effectively handles the problem of vast data collection from IoT devices with a low latency and faster response time, and support demands of quality of service [21]. Smart healthcare devices and wearable sensors monitoring patients will need an immediate response in critical situations, and fog can meet these requirements effectively. Fog's architecture inherently supports agility and flexibility compared to the cloud, which may be packed with much power but does not provide agility, needing computation for sensor networks. Many works have proposed architecture with different perceptions that are most suitable for dealing with various problems specific to certain use cases. However, the fundamental architecture remains the same, with the three-tier architecture consisting of cloud, fog, and IoT layers. Figure 2 describes the architecture advocating programmability, flexibility, and efficient data analytics aspects of fog nodes.

### 2.1. Cloud Layer

The cloud layer comprises a cloud server loaded with huge data storage and processing capabilities and services provided as APIs or direct connections. This layer is responsible for collecting pre-processed data from connected fog nodes and storing them in persistent storage on the cloud. With ample computing and storage resources in the cloud, it will perform tasks that are not feasible in the fog servers, ensuring global coordination and control amongst all connected fog nodes and servers. It could also provide centralized services such as maintenance and security enforcement policies and ensure flexibility for accommodating future modifications. Inspired by modern trends in big data analytics and artificial intelligence, the cloud layer can be programmed to further improve the management and efficiency of fog layer components from centralized processing and to learn from data collected from the fog layer.

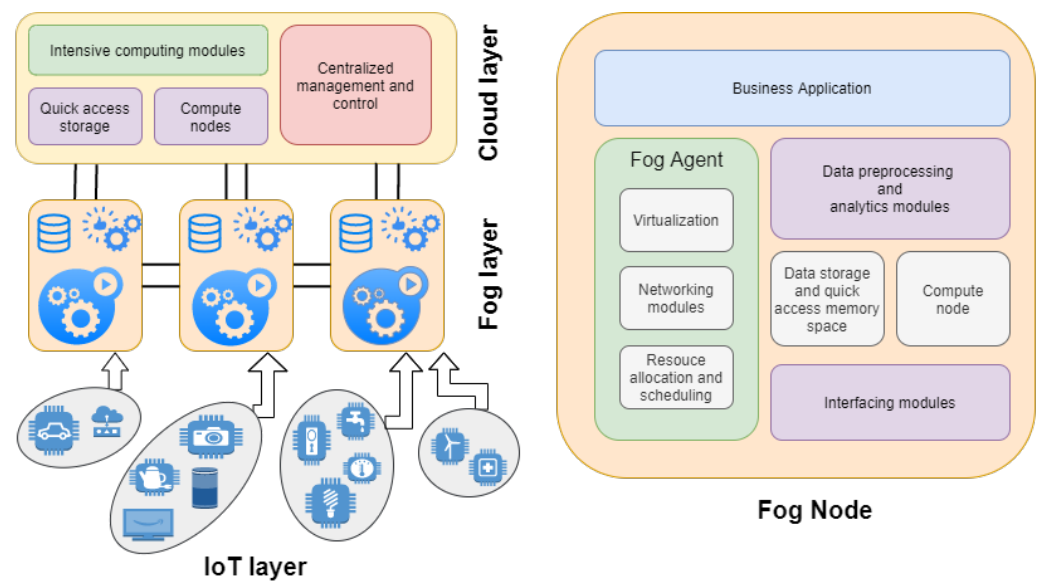


**Figure 2.** Layered architecture with fog computing.

### 2.2. Fog Layer

The fog layer resides between the cloud layer and the IoT layer and is a layer of heterogeneous nodes. For example, fog nodes could range from high-end servers, gateway devices, edge routers, computers, mobile devices, and smart vehicles to sensors with little processing capabilities or supporting different networking technologies, such as high-speed physical links or multiple wireless access technologies such as WiFi, 4G/5G, and LTE. For that matter, fog nodes have an abstraction layer that abstracts out the discrepancy of underlying hardware and technologies and exposes a uniform and seamless interface for management and control. Furthermore, multiple fog nodes interact with each other for data and processing coordination. Figure 3 shows the architecture of the typical fog node.





**Figure 3.** Architecture of fog node.

### 2.2.1. Fog Agent

The fog agent handles the entire fog node management, which holds core functionality modules such as virtualization, network management, and resource allocation and scheduling. Physical resources are abstracted to the upper layers and provide support for creating virtual hardware components and environments for running processes and services in the node. Due to this mechanism, it is easy to allocate resources according to the processing needs at the run-time by creating virtual machine instances on top of the virtualized infrastructure with the help of HyperVisor. Virtual machines are created to host various services and applications to serve the IoT data processing needs with a dynamic allocation of required virtualized hardware. This functionality helps to efficiently allocate and scale processing needs to certain limits using resources in the node.

Fog has inherent challenges regarding networking, whether inter-VM networking or with an external device. For that matter, the agent has networking modules that elegantly handle complex networking tasks and provides an abstraction layer, simplifying it further. It uses VNFs to provide network services with benefits of hardware independence, a high resilience, quick replacement, and easy configuration and deployment. From recent advancements in networking, it can also be customized to use case-specific networking modes to improve the throughput and reduce the latency in data packet communication.

A fog node can deal with many simultaneous connections demanding various resources and services for different tasks. In that situation, the fog agent acts as a resource allocator and manager for serving virtualized resources and de-allocating when the task is finished. The agent includes service orchestration functionality and policies for life-cycle management with a global messaging bus to send control signals for synchronization. It also deals with secure inter-process communication and consistent data resource sharing between different VMs. Various resource allocation strategies are implemented in the resource management module. These should be selected carefully because allocating resources efficiently is necessary to reduce latency in communication.

In addition, the fog agent also provides APIs to satisfy programmability needs at a low level in the node architecture. This feature supports the customization of fog nodes to accommodate functionalities such as real-time data synchronization between multiple devices, defining QoS, creating custom computing policies, and centralized control with mobile devices.

### 2.2.2. Interfacing Modules

Today's easily accessible IoT devices pose a huge challenge of interfacing them with some uniform access method. Therefore, fog nodes must have interfacing modules through which data exchange between IoT devices is possible. They could be the standard interface or follow some proprietary protocol. Interfacing modules will handle all complexities, such as sense, the establishment of the communication channel, bandwidth, connection type, and the nature of the data stream regarding connected device specifications. One service loaded with these interfacing modules will be made available, continuously running in the background for handling incoming requests and servicing them in combination with other resources following the requirement of the connected device.

### 2.2.3. Data Storage and Quick Access Memory Space with Compute Node

This component corresponds to the primary advantage of the fog computing paradigm by processing data on the network's edge rather than the data going to the cloud. Fog nodes have sufficient storage and processing power to process the data from IoT devices. Devices working as fog nodes can have various types of memory and compute resources, so this flexibility needs to be addressed by the fog agent through virtualization. There could be many types of storage memory available at the fog node, such as local storage memory, and faster memory, such as RAM and cache memory. For the processing, the compute node will have processors and GPUs, and for machine learning and deep processing, the compute node will have TPUs and ML processors. Virtualized resources will be available for applications to serve incoming connections, and the fog agent will handle allocations on a requirement base.

### 2.2.4. Data Pre-Processing and Analytics Modules

Data collected from IoT devices could be huge, so they need to be processed efficiently without wasting much processing power and time. Various optimized pre-processing and analytical modules are loaded in the memory to accommodate these facilities. They are designed to process data in a parallel manner, which is crucial for serving multiple devices requesting services. In addition, some fog nodes may have AI-enabled hardware for advanced prediction and learning mechanisms with corresponding libraries. For example, nodes with the Intel processor family "Myriad" have SDK with a neural network compiler and Google Edge TPU with TPU libraries in order to use dedicated resources for deep learning in fog computing [22].

### 2.2.5. Business Application

This is an essential component dictating the behavior of the fog node. It is easily programmable on top of fog resources and defines the services provided by the fog. The application will determine the role of the fog node in the network with IoT devices, which could be simple interfacing and centralized monitoring, or much more complex, such as deep analytics on real-time data and synchronization in combination with constant checking with other services to provide complete IoT network management. Systems such as smart health devices systems, home automation solution, parking space suggestion systems, or smart city management could be developed with the help of a network of multiple fog nodes leveraging a distributed nature. Furthermore, the architecture of the applications could be inspired by a monolithic, micro-services-based model or distributed containerization to exploit various advantages of scalability and QoS [23,24]. For example, frameworks such as Fogernetes [25], based on Kubernetes, could be used to develop the distributed application.

## 2.3. IoT Layer

The IoT layer includes all IoT devices on the network. There could be two types of IoT devices: fixed and mobile. Fixed IoT devices are located in particular fields or locations, such as smart home assistants, smart door locks and ambient light sensors, RFID tags

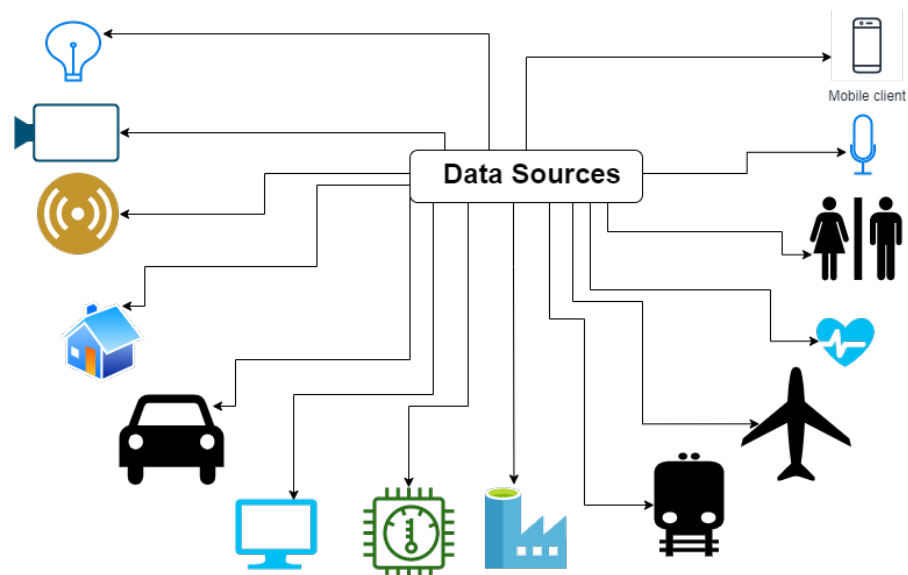
and sensors, air quality monitoring systems, and smart alarm devices connected with physical links or wirelessly. On the other hand, mobile IoT devices could be portable and easily carried by users such as smartphones, wearable IoT devices, and vehicles. These devices form a layer of IoT devices and are connected in an ad hoc fashion with the network, generally via wireless access. Generally, these devices have restrictions on the data that they can process, and a limited bandwidth. Therefore, their main function is to collect data and provide it to the upper layer for processing and storage. Different devices can have different interfaces to interact with each other, which poses a challenge in data communication between IoT devices for automation needs. However, interfacing modules in the fog agent solve this issue and provide a standard for easy data sharing amongst IoT devices with greater agility.

### 3. Data Analytics

#### 3.1. Data Generation Sources

Due to the IoT revolution, almost everything is becoming a source for data generation. As a result, a tremendous amount of data are generated every second. These data need to be pre-processed before something useful can be derived from them because only some of the generated data are relevant or useful. This section will look at the various sources from which data get generated.

As we can see from Figure 4, typical data sources include mobiles, various types of sensors and actuators including thermostats, engines of airplanes, factories, mobiles, computers, automobiles such as driverless cars, metros, human health data, smart devices such as Google Home, Alexa Echo Dots, smart homes, smart shoes, watches, and, in general, all wearables, etc., and the number of items on the list increases all of the time. The data generated from these sources can be subtly classified into three major types of sources, which are as follows.



**Figure 4.** Data generation sources.

##### 3.1.1. Passive Data Sources

These data sources do not communicate data actively with the network or the intermediary nodes. These sources need to be pushed to the active state before data transmission takes place, and they transmit the data only when requested to them. These typically have a low power consumption and are present in remote places. We can take an example of a sensor that measures salinity in a sample. Here, the sensor will get active only when an API call is made to obtain readings.



### 3.1.2. Active Data Sources

These data sources generate data continuously in the form of streams of data bytes, just like a driverless car. The data keep flowing to the nodes, as opposed to the passive sources, where we have to request data. Thus, the data need to be accepted in real-time, and the applications running on top need to be very accurate and sophisticated in the data transfer, as no data should be lost or misplaced. They should be handled well for further processing.

### 3.1.3. Dynamic Data Sources

The data sources are termed as smart data sources. Here, communication with the application takes place instead of just data passing. The communication is bidirectional and also dynamic or in real-time. We can take an example of a surveillance camera with a small code of face detection embedded into it. When a criminal's face is recognized, the camera can send an alarm message to the cloud-based criminal dataset program and communicate with it. Thus, these devices can not only send data but also alter the data format; for example, the camera does not send the whole image to the cloud, but only sends data in the form of whether the criminal has been identified.

## 3.2. Data Analytics: A Brief Introduction

Big data is a term that needs to be understood before we understand data analytics. It has been a topic of propensity to talk about and is just a byproduct of the Industry 4.0 revolution. It is a term that refers to the large amount of the raw dataset directly generated by the sensors or actuators. It can range from highly unstructured to highly organized and comes in various combined formats, including text, image, video, etc. To deal with such a huge amount of highly unstructured, complex, and hybrid data, we use data analytics or, more specifically, big data analytics. Big data analytics refers to a vigorous and convoluted analysis and deriving semantics to make accurate decisions [26]. Data analytics has become the most important task in the IoT and fog paradigm, as everything today is data-driven. The advent of big data also creates a new need in the market for data analysts and scientists, who require great skill and expertise to obtain some value from the vast amount of data [27]. The basic flow of data analytics is depicted by Figure 5.

## 3.3. Current Trends in Data Analytics

Data analytics has many opportunities and a large scope for improvisation considering its vast applications in the IoT and fog paradigm. Some of the current trends in data analytics include:

- Continuous intelligence;
- Graph analytics;
- Commercial AI/ML
- Conversational AI analytics/ NLP;
- Augmented data analytics;
- Automatic data and content management;
- Persistent memory servers.

We will look at a couple of them in detail below.

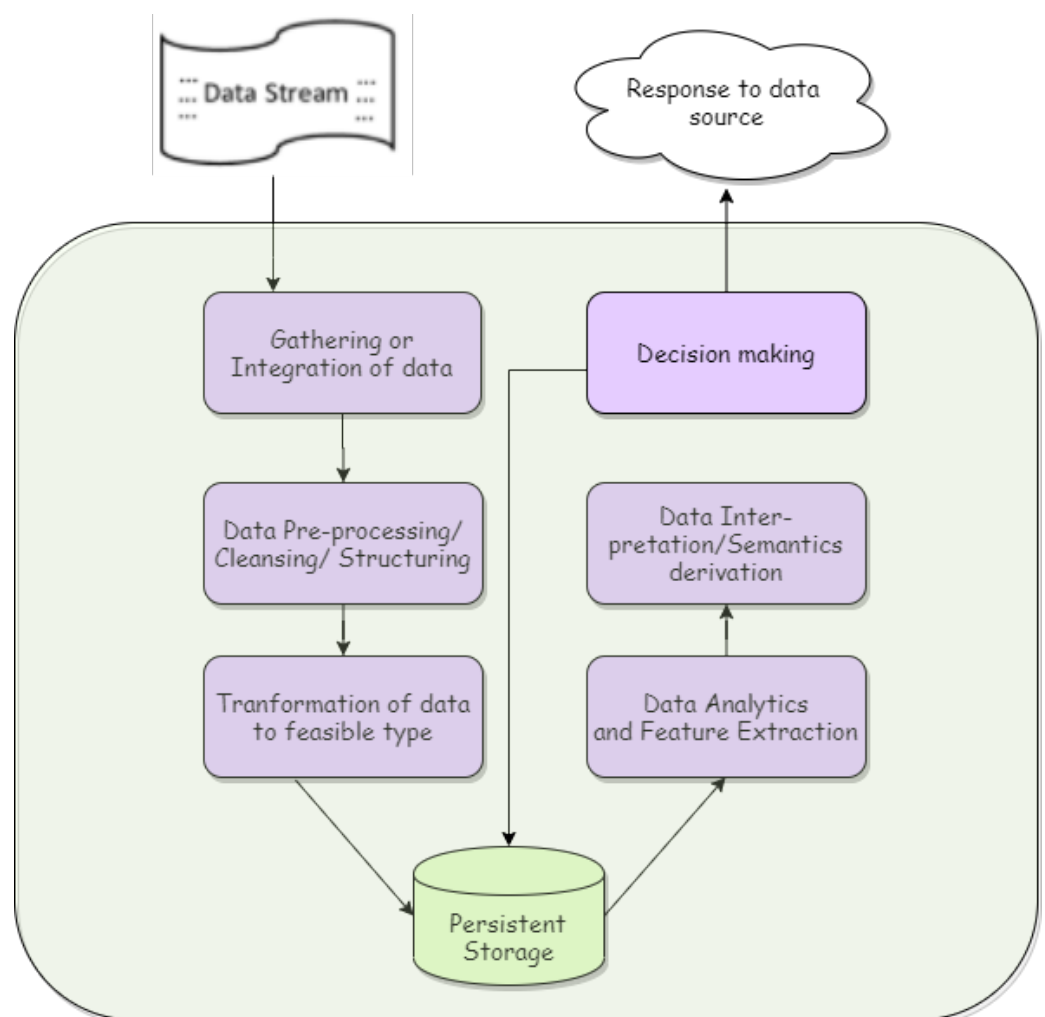
### 1. Augmented Data Analytics

This is considered as the future of data analytics. The main motivation for a company to perform data analytics is insight generation from data. In the present scenario, the industry has a shortage of data scientists/analysts, and the need is increasing even more. The McKinsey Global Institute estimated that the U.S. economy could be short of around 250,000 data scientists by 2024. Even if the gap gets filled somehow, data scientists are not business experts; they can perform all tasks independently and must be under the constant scrutiny of business analysts. Thus, augmented data analytics is emerging to overcome all of the barriers because it reduces a company's dependence

on data scientists by automatically generating insights. It does so with the help of complex and advanced machine learning and artificial intelligence algorithms.

## 2. Persistent/In-memory storage

Persistent storage is also one of the emerging trends, and will help foster data analytics even more. As the data generated are increasing exponentially, there is a current need for better ways to store these data so that they can be accessed rapidly with fewer latency issues. Persistent memory (PM) combines the byte-addressability of DRAMs and the non-volatility of disks and flashes. PM can be supported either through direct DAX or block access. The use of PM can be performed in three ways. Firstly, the applications can use it as an external or augmented storage entity and are not concerned about the non-volatility. In this case, the applications do not need any changes. Secondly, the applications can use its non-volatility property against DRAMs. Here, the applications themselves need to be modified to use the persistence property of PM. In the third case, the applications may use just PM instead of flash or drives.

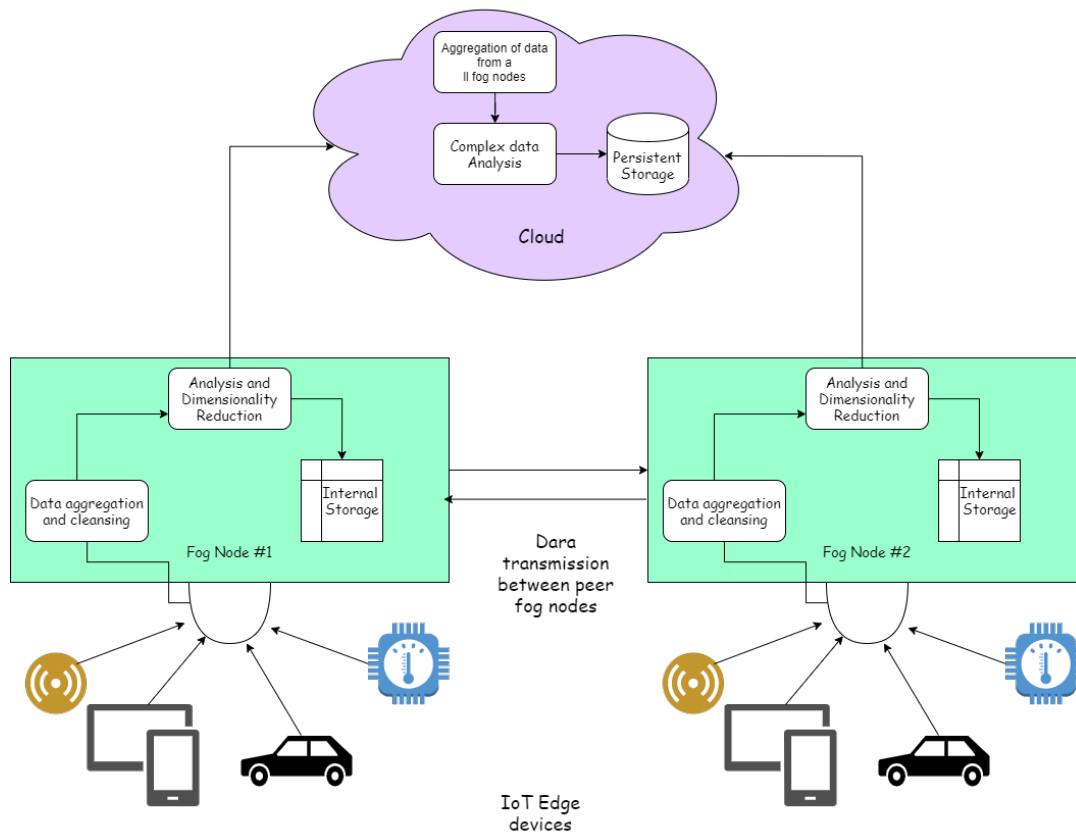


**Figure 5.** Flow diagram for data analysis.

### 3.4. Role of Fog Computing in Data Analytics

As we have already discussed, big data contain the following characteristics: volume, velocity, variety, and veracity. Dealing with all of these for real-time applications by ensuring availability and deliverability is a cumbersome task for the cloud. Data processing and analytics need to rely on robust and highly scalable messaging systems, commanding software engines for data stream processing and scalable data storage solutions. Fog computing has much potential for solving issues with huge data storage and for quick

data analysis to respond to numerous events that call for prompt decision making and action [28]. Therefore, we introduced the paradigm of fog computing earlier in the paper. Now, we will see how fog computing will help us to achieve all of the benefits that it is known for, especially in data analytics. The role of fog computing in data analysis and the segregation of jobs from the cloud is best explained by Figure 6.



**Figure 6.** Data analytics using fog computing.

Fog computing presents a distributed method of computing, as opposed to the holistic approaches of the cloud [29]. Fog data analytics is a field where fog computing is used for data analytics at the edge of the network to reduce latency for applications needing a high availability and faster response times. The fog nodes are installed near the edge of the network, and they distribute the load of data going to the cloud by using virtualization techniques and absorbing all of the raw data from the IoT edge. A fog node can be thought of as a mini cloudlet. Then, these nodes can also perform data analytics on their own, responding directly to the devices, offloading local insights to the cloud in the background, and communicating with neighbor nodes about the learning. Augmented data analytics is widely used for automatic data processing and insight generation. The fog nodes cannot perform all data analytics work like the cloud due to the little resources of the cloud, but they are sufficient for local analytics. The cloud then performs a more complex analysis of all of the data collected from different fog nodes for obtaining global and heuristic insights into data [30].

#### 4. Current State-of-the-Art

In the era of faster communication and even faster computational needs, fog computing prevails in the industry with such requirements by processing data closer to the source. Modern advancements such as big data analytics, machine learning, and deep learning are combined with fog for generating data analytics on the fly with adaptive learning [31–34]. Leveraging the technology of blockchain to accommodate privacy and security

features in the fog layer is proposed in [35–37]. Software-defined networks (SDNs) with customized routing protocols are used to increase the energy efficiency and performance between fog nodes [38]. With SDNs and named data networking (NDN) [39], traditional networking concepts are redesigned to harness new advantages in fog computing. They drive innovations in various fields and transform the experience for greater efficiency and convenience. Recent developments such as quantum computing paradigms are also considered to optimize the performance of fog-based systems for parallel processing and dynamic load scheduling using quantum-computing-inspired optimization (QCIO) [40]. Finally, some areas harnessing the power of fog computing with IoT devices are explored for state-of-the-art implementation.

#### 4.1. Smart Factories

IoT has enabled automation and obviated the need for human interaction, adding new capabilities, such as logging every event and making intelligent decisions at certain times. Having such advantages, IoT is most suitable for driving automation at various levels in the industry. For IoT technology used in the industry, there exists another specific term: the industrial Internet of things (IIoT). These IIoT devices include actuators, RFID tags, robots, flow control and conditioning devices, and sensors for various manufacturing, testing, and quality control applications. The cloud with these devices provides centralized control and helps to automate tasks with deploying applications on the cloud, managing IIoT devices communication and data analytics [41]. However, this implementation needs to deliver the expected value for some time-critical systems because of latency issues due to the round-trip distance between IIoT devices and cloud services. Fortunately, the problem can be remedied by enabling fog computing with IIoT to handle real-time data processing and analytics. Fog nodes host monitoring and control services that make it possible to serve low-latency requirements as they are closer to the IIoT devices, along with the implementation of SDNs to develop control plane behavior and employing time-sensitive network (TSN) protocols for optimizing and sustaining the network's effectiveness [42]. With fog nodes with a sufficient processing power, real-time data analytics, such as the prediction of quality and estimation of faults, are generated with a high availability.

#### 4.2. Healthcare

The healthcare industry is going through a total paradigm shift from cloud-based applications to fog-based IoT device data management. This transition is due to its requirement of a high availability with geospatial awareness and an instant response to events reported by health-monitoring sensors. Body area networks (BANs) are most popular for monitoring the health of individuals without any human observer in a friendly way. They also provide notifications to relatives and instantly contact emergency services with complete automation. Fog computing power requirements of healthcare have severe time constraints. Hospitals deploy fog nodes for collecting data from nearby IoT devices and accommodate the need to monitor the health condition of many patients. Fog enables smart healthcare, which notifies appropriate staff members when patients are in critical condition with their geospatial location. Various methods have used deep learning to detect epileptic seizures [43], heart disease [44], and cancer classification [45] in combination with fog nodes and data from IoT devices on the fly. Systems such as fog ambient assisted living (FAAL) [46] have been developed to help people suffering from neurological disorders, and personalized healthcare support for remote patients with diabetes is making a tremendous impact on people's lives [47].

#### 4.3. Home Automation

IoT devices have a huge role in home automation, making it a smart home with connected devices and data synchronization over the network of IoT devices. These devices are connected with wireless access technologies and can make intelligent decisions according to situations and conditions. Devices such as intelligent lights that adjust the

lights according to the available sunlight at a given time, smart thermostats that dynamically adjust temperatures in combination with devices such as air conditioners or heaters, and security cameras that intelligently recognize people and track their movements are driving home automation and the quality of living to new heights. Data from all of these devices need to be processed in real-time for the best experience and to receive immediate support in emergencies. In addition, with adaptive learning mechanisms to constantly shape the services of all of these smart devices to serve the best experience to the user, fog nodes embedded with machine learning and deep learning hardware provide fully customized and user-centric automation without any direct human interaction. In modern scenarios, all smart IoT devices are connected to the central network and can be accessed and controlled from mobile and wearable devices. These smart devices, in coordination with each other, through sharing real-time data, have also resulted in an overall reduction in energy consumption.

#### *4.4. Vehicular Networks*

The number of vehicles on the road significantly increases each year. Managing communication more securely and effectively is more difficult due to urbanization and population convergence. A driver mistake is a major factor in many accidents and catastrophes. Because of this, it is crucial to monitor driver behavior [48] continuously. The concept of connected things is popular; it has made its way to the network of vehicles to exchange data using wireless technology and several other devices, such as RSUs and computational resources, present in the vehicles. Data aggregation and communication become even more complicated when multiple sensors, in addition to sensors for the vehicle and surroundings, perform this function. Intuitively, these networks are highly latency-sensitive and need a quick response, which could be accommodated by using fog computing and computational resources present in the vehicles to act as fog nodes. This implementation supports a low latency in data communication between RSUs and vehicles, and vehicle-to-vehicle communication happens by sending different types of messages to nearby vehicles. These messages could be a query, response, or cooperative, serving the purpose of intelligent data sharing [49]. The vehicular network has great advantages as it helps the driver to safely change the lane, provides estimated congestion at certain points, and suggests alternative routes to avoid congestion. The smart network of connected vehicles implemented with the SDN approach further amplifies its capabilities and advantages [50]. Many traffic signals combined with vehicular networks would result in effective traffic management and a reduced waiting time [51].

#### *4.5. Disaster Management*

Currently, many cameras and monitoring devices are available throughout the cities, which can detect disasters such as fire, smoke, or crashes. Processing live streaming data from the sources as a method of identification could be a task that needs immediate analytics generation and decision making. Conventional cloud-based solutions would be scalable but suffer from a lack of processing on edge devices to make quick decisions, resulting in a higher latency. The implementation of fog makes edge computing possible, satisfying real-time analytical needs for taking certain actions. Systems using fog computing to forecast and monitor disastrous phenomena [52] at coastal areas and ensure the safety of the people and infrastructure have also been proposed. The same could be implemented in the smart home environment for fire and smoke detection [53] at any location with the help of sensing devices and fog nodes. Disastrous events such as a gas leakage, fire, and short-circuit could be averted with fog processing and IoT sensing.

## **5. Research Challenges**

### *5.1. Low-Latency Transmission*

IoT devices generate a large amount of data, so we need efficient data processing capabilities to handle that large amount of data within certain time constraints. On the

other hand, real-time latency-sensitive applications pose challenges that are not solved only by efficient data processing but also stipulate that the processed data are within a few milliseconds. Some solutions have been proposed to satisfy these requirements, but active research remains because of the inherent complexity and design challenges.

The challenge of low latency is amplified dramatically when many devices depend on each other for data and responses. Data must often be processed to some extent so that they can be served to another device because of that particular device's interfacing and structural data needs. At this point, if the latency constraint is not taken into account correctly, it will hugely impact the performance and effectiveness of the system of inter-dependent IoT devices because there is already a delay due to some level of pre-processing. A higher latency will worsen the situation, resulting in a waiting time for data and a poor performance, eliminating the comparative advantage of fog computing. Table 1 compares various works that show latency sensitiveness in different areas.

As a partial solution, Dew Computing can be considered by packing pre-processing capabilities within the devices and establishing a direct communication path between them. However, if devices could use data from multiple devices, this approach could be more economical, and another alternative should be explored. For example, Magurawalage et al. [54] suggest the concept of aqua computing for improving the user experience and offer clone architecture to accommodate lower latency requirements by cloning user-specific data in the edge devices so that they are directly accessible to the user without any delay. The Internet provider is considered to manage these clones to serve the user on demand and is responsible for updating them for the time being. However, managing user-specific clones would be costly in terms of resources.

Additionally, no perfect universal solution can exist for low-latency applications, and some trade-off is needed for the high availability, latency, and redundancy. If we consider an application-specific solution, then a large amount of heterogeneity would be added to the network, resulting in difficult tasks to manage.

**Table 1.** Research work on latency sensitivity in different areas.

Work	Scope	Focused Aspects	Strength	Weakness
Naranjo et al. [55]	Smart city	Latency, energy consumption	Lower energy consumption, heterogeneous communications between IoT devices	Low scalability, lack of real-time data processing
Singh et al. [56]	Smart grid	Network utilization, latency, energy utilization	Context-aware information with reduced latency, better energy and network usage	Lack of resource cost measures, high computational complexity, unevaluated overhead
Mahmud et al. [57]	e-Healthcare	Latency, energy consumption, network utilization	Lowered energy consumption with low response time	Mobility is ignored, latency caused by high computation
Chamola et al. [42]	Fog implementation with cloudlets	Response time	reduced network latency for SDN	Energy consumption was not evaluated, latency caused by high computation
Romeo et al. [7]	Robotics application	Power consumption, latency	Modeled battery discharge profiles, reduction in power consumption, low latency	Accuracy was not investigated, low scalability
Alam et al. [58]	Mobile application	Response time, energy consumption	Low execution time and low latency, suited for multi-agent architecture	Mobility, privacy, and context awareness were lacking
Ahn et al. [59]	General	Energy, wait time	Considered gap in the wait time, energy expenditure for different devices, latency	Heterogeneity was not considered, inter-dependency of IoT devices, high computational complexity

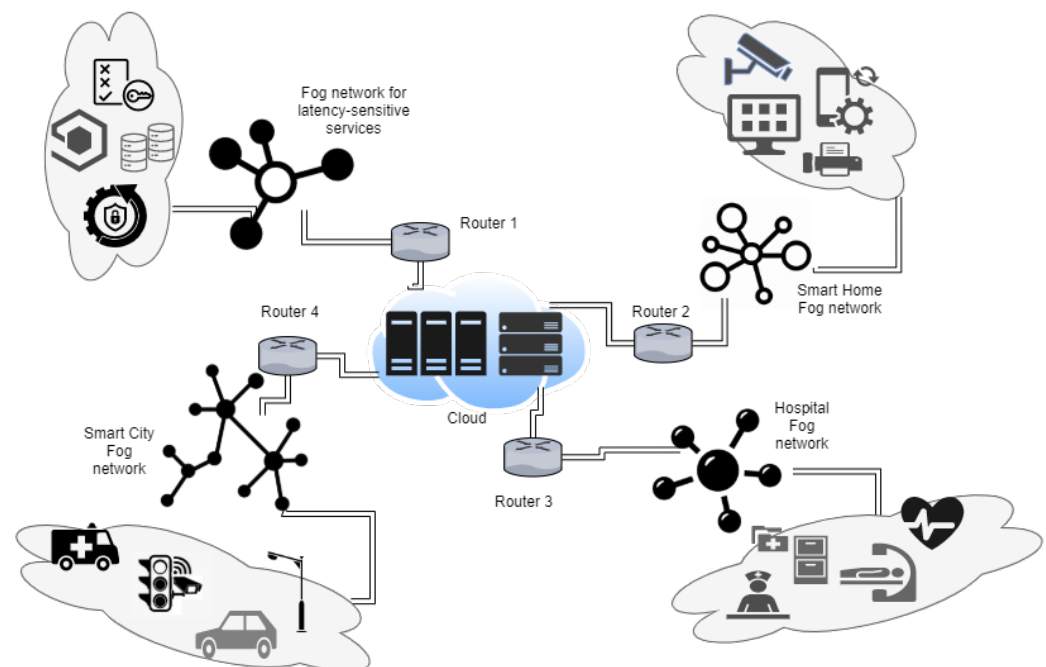
## 5.2. Heterogeneity and Interoperability

Myriad IoT devices [60] present in the market have a different set of data and interfacing requirements, which poses a great challenge in integrating it with an entire network without incurring the additional cost of computation and hardware. Devices such as thermostats, ambient light sensors, fire and smoke detection sensors, digital voice assistants, smart door locks, and connected media devices induce a large amount of heterogeneity in the network, and data communication between these devices needs a uniform interface.



Fog nodes solve interfacing issues by abstracting the interfacing complexity and providing a uniform standardized interface for easy communication and real-time data stream sharing. Furthermore, fog node communication protocols should be designed in a way that can cope with future changes. Some protocols proposed in [55,61] are used for energy-efficient routing and navigation in heterogeneous networks.

The fog layer could be deployed as distributed fog nodes with someone acting as the controller node for larger system requirements. Here, too, nodes need to be of different types, and they still should work in harmony. In case of failure of the controller node, some other node should act as a controller to handle the operations without any problems and continue to serve connected devices. Devices should be easily connected to the network, and data transmission between different types of devices should be seamless to support the best user experience. In smart homes, most devices are connected to a central wireless network and can be easily operated and controlled with different mobile devices. For example, digital assistance and a stereo system connected with the wireless network are controlled using smartphones or tablets, which should support platform independence and uniform APIs. The heterogeneity of devices can be observed from Figure 7.



**Figure 7.** Fog network of heterogeneous devices.

### 5.3. Programmability

Applications deployed on top of the fog nodes are meant to serve the user's specific interests. The type of data storage and nature of pre-processing and data sharing could differ from one use case to another. Therefore, fog nodes should accommodate programmability needs without restricting the platform and underlying hardware. For example, a computer operating on a Windows operating system could act as a fog node or a smartphone with Android; different characteristics and capabilities of devices are taken into account when deploying an application. A smartphone cannot handle continuous data processing tasks because of battery constraints; therefore, the programmability of these types of devices is limited, and similar considerations should be figured out when designing a fog computing environment. It may be challenging to develop a generic fog node setup that is compatible with many devices.

If the fog node is equipped with high-end processing resources, it could be leveraged to improve data analytics and storage capabilities, which would help to adjust fog network

parameters to maximize the quality of the user experience. Machine learning and deep learning can be employed for that purpose. Many works have proposed such solutions in [62–64], helping to reduce the energy consumption [56], low-latency delivery, and QoS, as well as bandwidth optimization. However, the complexity of the technology stack and architecture of the fog node will directly affect the effectiveness of the data analytics services deployed on the fog node. Fog nodes have a limited data storage capacity compared to the cloud, so data that could be cached into the fog node for quick access are limited. Therefore, appropriately policy managing the cache in such a way that minimum external contact is needed and no unnecessary data remain present is necessary. The author in [65] proposes a new caching scheme known as the Steiner tree-based optimal resource caching scheme. Fog nodes use this tree to reduce total path costs and minimize resource caching costs. However, if the fog layer is composed of many other fog nodes, it further complicates the policy that provides a high availability, consistency, and coherency.

#### 5.4. Quality of Service

The sheer amount of requests and network traffic from connected devices jeopardize the network performance. This results in significant delays, broken data streams, dropped calls, and glitches in video calls. Moreover, only some of the services in the network hold equal priority; some time-sensitive application [66] needs to be serviced quickly and continuously with required resources; otherwise, it would result in troublesome delays and fragmented data packets. On the other hand, applications such as a file transfer could bear some delay in the data transfer and still have intact packets. Therefore, ensuring the QoS is essential for managing the network effectively and delivering the values demanded by different services.

The need for QoS parameters such as reliability, the continuous availability of bandwidth and low-latency servicing, resource provisioning, high-priority data transfer, and mobility poses a great system design challenge. Real-time data application needs a continuous bandwidth and low-latency servicing for the best user experience. A revolutionary black-box multi-algorithm described in [67] shows how end-to-end latency might be reduced by 60%–70% while largely utilizing the temporal locality. The processing and networking time are the two parameters used to measure the end-to-end latency. Similarly, the fog node must be able to prioritize data processing needs and serve required resources so that the execution does not introduce any further delay. Authors in [68] proposed a method for estimating resource allocation, based on how many resources will be utilized according to historical data and customer usage patterns, named media fog resource estimation (MeFoRE). Table 2 compares various works by authors in their research and shows which QoS aspects were considered by them.

With traditional networking, fog computing is refrained from its true potential for QoS and flexibility with a lower latency by constantly learning solutions from modern developments. SDN gracefully decouples the control plane from the data plane and emulates centralized control over the data plane. A fog layer implementing SDN networking would benefit from flexibility and data forwarding control offered by SDN. A fog node can work as an SDN controller that guides the data plane according to network requirements, which will ultimately help to improve the QoS and efficiency of data analytics tasks, eliminating latency issues.

Fog-based systems can deliver a high-quality user experience to end users. First, the preference of services is decided based on the behavioral usage patterns, geographical location, and mobility context, and then the user experience is optimized from gained values of parameters. The work of [69] describes delivering the QoS to users in IoT ecosystems that leverage a virtual infrastructure proposed as the self-organizing fog of things (SOFT-IoT). The fog of things gateway has been set up to work as a smart device based on fog computing in the environment.

### 5.5. Scalability

Scalability is among the main characteristics, the need of which gave birth to cloud capabilities near the end devices so that, instead of overwhelming the cloud with a huge amount of data, fog focuses on filtering it at the fog layer. By 2022, USD 2.5 million will be spent every minute on the IoT, and one million new IoT devices will be sold every hour. Now, processing at the edge would help to overcome the problem more efficiently at this scale. However, not only computing at the edge but also the coordination of such a large number of devices needs to be handled as requirements grow over time. This presents the challenge of designing a scalable solution to accommodate the data analytics needs of collected data from connected devices.

A variety of use cases have different approaches for solving the particular problem, and they might be inherently disparate, serving those specific requirements. For example, for the management of smart home devices, one fog node equipped with sufficient resources might be enough. However, regarding connected devices throughout the city, a large number of fog nodes is certainly required. In addition, the resources need to be scaled with the increased number of users. The relevance of placing services at various levels of the tree topology, which is rooted at the cloud data center, is discussed in [59] to optimize the performance and scalability. The work [70] presented multi-tier fog architecture consisting of ad hoc and dedicated nodes with dedicated and opportunistic computing resources, primarily focusing on maximizing analytics service utilities. The authors in [71,72] proposed systems that support real-time efficient data processing for vehicular fog networks and easy traffic management.

**Table 2.** Comparison of QoS aspects from various works.

Work	Scope	Quality of Experience	Energy Efficiency	Delay Sensitiveness	Reliability	In-Network Caching
Stojme vic et al. [68]	Machine-to-machine networks	✓		✓		
Huang et al. [66]	Vehicular networks	✓		✓		
Craciunescu et al. [73]	e-health applications	✓		✓		
Dantu et al. [74]	Smartphone-based applications	✓		✓		
Sarkar et al. [75]	IoT-based applications		✓	✓		
Dastjerdi et al. [76]	IoT-based applications	✓	✓	✓		
Zhu et al. [77]	Website rendering			✓		✓
Hu et al. [78]	Mobile applications		✓	✓		
Hao et al. [79]	Ubiquitous computing		✓	✓		✓
Mubeen et al. [80]	Automation applications			✓		
Shih et al. [81]	Radio access networks		✓	✓	✓	
Prazeres et al. [69]	IoT-based applications		✓	✓	✓	
Flores et al. [82]	Social-aware device-to-device communication			✓	✓	
Fan et al. [83]	Web-based applications			✓		✓
Wang et al. [54]	Device-to-device communications		✓	✓	✓	✓
Su et al. [65]	Latency-sensitive applications		✓	✓		✓

Not only the placement and efficiency of fog nodes but also the number of fog nodes and type are the factors affecting the network's performance. It is also a necessity for the placement of nodes to be flexible according to the demand at a certain time. In the case of a smart city, fog placement should be dynamic enough to support changing the number of users at any time and to continue operating and servicing users. To support this goal, ref. [84] leverages the mobility characteristic of buses to deploy fog nodes, presenting a fuzzy-based real-time auto-scaling (FRAS) mechanism. Auto-scaling will surely help to improve the overall performance and QoS, even at times when there is a high load on resources. The author of [85] integrated the hypervisor technique with container virtualization and constructed an integrated virtualization fog platform for deploying industrial applications based on a virtual network function. The FRAS mechanism presented in the paper provides a dynamic, rapid, lightweight, and low-cost solution to service auto-scaling problems.

### 5.6. Authentication and Access

As we have mentioned earlier, IoT devices are increasing exponentially. These devices access and share resources/services such as storage, PaaS, and SaaS with the omnipotent cloud [86]. Thus, the authentication of these devices becomes the responsibility of the cloud in order to identify legitimate users and keep any other person or bot with malicious intent at bay. Sometimes, these illicit users attack the servers and keep their servers busy by bombarding spam requests. IoT devices are quite vulnerable to cyber-attacks, especially Dos and DDos attacks [87]. Thus, a robust authentication technique with a quick response time must be used. Some people claim that using a cloud-based technique prevents the devices from malicious attacks [88]. Although authentication via the cloud is secure and robust, it has a latency problem and is also not scalable[89]. Hence, we have to accept the help of fog nodes by performing some authentication processes at the edge in fog nodes [30].

Authentication and access for the end devices is a challenge in the fog architecture due to many reasons. Firstly, the fog architecture already comes with a resource constraint as it has to serve many devices with limited resources. Thus, ensuring the continuous availability of fog nodes is also a challenge in the authentication scheme. Secondly, if fog nodes have to carry out authentication at the edge, they either have to perform the authentication or communicate with the authenticating agent through API calls. This task increases the workload of the data transfer channels of the fog node, and the available bandwidth also reduces security-related threats because of the transfer of sensitive data, such as credentials near the edge. Therefore, many authentication schemes have been proposed to date. One of the techniques used is the single sign-on (SSO) authentication scheme. In SSO, a single login is required, and it provides access to all of the services within the system, where no further authentication is necessary, but this scheme has a backdoor for man-in-middle attacks [90–92]. Moreover, if the SSO provider gets compromised, then the security of all other devices under that are compromised as well [86]. Therefore, multi-tier and multi-factor authentication schemes are used to ensure that the security of users is not compromised.

From the above discussion, it is clear that authentication and access is a double-edged sword. If more focus is placed on making the authentication system more robust by introducing a multi-tier authentication architecture, user experience and convenience is at risk. On the other hand, if user convenience is catered for, the security might get compromised. Dealing with both simultaneously by maintaining the right amount of balance between the two is one of the greatest challenges in authentication in fog-enabled architecture. Table 3 summarizes the various authentication techniques implemented to date, along with their pros and cons.

**Table 3.** Comparison of authentication techniques.

Authors	Features	Auth Tiers	Control Agent	Pros	Cons
Banyal RK et al. [93]	Arithmetic-based CAPTCHA calculation, OTP, and IMEI-based authentication mechanism	3	Server	Resistance to multiple attack type	Computational complexity
Emam AHM [94]	Authentication through dynamic link sent on registered email address	2	Server	Cost-effective	Email address may get compromised
Kumar S et al. [95]	OTP entered through personal device	2	Client and Server	OTP needs to be entered using personal device	Access lost if registered Device is lost or damaged
Usman AA et al. [96]	Security token generated using preshared pin numbers, location, and time	2	Client and Server	Cost-effective	Clock synchronization problem
Liu S et al. [97]	Sends QR code to register mobile device's Bluetooth address	2	Server	Cost-effective	Mutual authentication not considered
Ahmad S et al. [98]	Smart card and biometric scanning	3	Server	Multiple factors required to access services	Smart card mandatory
Singh TG et al. [56]	Based on graphical patterns	2	Server	User convenience	Patterns are predictable
Soni P et al. [99]	Splitting and distribution of OTP over different channels	3	Server	Difficult to listen covertly	User inconvenience
Dhamija A [100]	Requires hardware token	2	Server	Resistant to different types of attacks	Expensive solution

### 5.7. Prediction and Optimization

One of the major applications of fog architecture is vehicular networks because of their low latency requirements and continuous and uninterrupted availability. Much is at stake regarding handling vehicular networks through the fog environment. The main intricacy here is that the vehicles are constantly moving at different speeds [101]. On the contrary, the fog nodes that they connect to do not move and are stationary or anchored at specific points on the streets. Thus, during motion, the vehicles need to constantly connect and disconnect to different fog nodes that come their way very quickly so that they can perform the actual communication that they intend to. This is where the role of prediction becomes important. Latest and state-of-the-art machine learning and deep learning algorithms are used for a better prediction accuracy. Some authors have also emphasized using caching policies [102,103] for content popularity and user preference prediction using the online gradient descent (OGD) method [104], prediction in terms of choosing the nearest fog node, and also cost prediction. The challenge here also lies in the state transfer during the constant connection and disconnection with the fog nodes. As the vehicles move, they detach from present fog nodes and reconnect to the next. Therefore, the vehicle's state needs to be transferred to the next fog node to perform further computations and reduce the overhead of calculating everything repeatedly.

Optimization is the biggest challenge in fog computing at present. Optimization, both in terms of availability and cost, is necessary to ensure on-demand serviceability, efficient resource allocation, energy consumption, security, reliability, resource usage, and bandwidth optimization, as well as thwarting unnecessary overheads. Optimization can also be talked of in terms of data-driven, code-driven, task-driven, or a combination of these. Many parameters can be considered for optimization in the fog-driven environment, as the performance depends on many parameters. An optimization problem can be defined according to the application by setting constraints on some parameters to achieve desired results [105]. For example, the power consumption constraint applies to end devices because they run on batteries, whereas this is not the case with fog nodes as they do not operate on batteries. The challenge here is that, although individual optimization problems have been defined, they are yet to be integrated into one coherent and cohesive unit for overall optimization. Thus, we need to find different optimization techniques for different sub-optimizations within the system. As we already know, the fog environment is a distributed environment, so any algorithm applied for any purpose in the fog should be distributed. This is also a challenge in optimization, as designing a distributed algorithm requires special skills and knowledge of the distributed architecture domain.

In big and convoluted networks, there is generally a multi-tier fog architecture with more than one layer of fog nodes. The computation power of a fog node is directly proportional to its distance from the edge, which means that the farther a fog node is from the edge, the more computation power it has. Still, because of being far away, it is associated with a latency overhead. This imbalance is termed the power–latency tradeoff. Therefore,

when the workload on a particular fog node increases, it can offload part of its workload to the next layer's nodes, and this is where optimization is needed. The authors in [106] mentioned a novel approach for predictive offloading and stochastic network optimization in resource allocation by using a queuing model for optimization.

### 5.8. Orchestration

The fog paradigm has envisioned a distributed computing and allocation approach as opposed to the centralized approach of the cloud paradigm. Different orchestration techniques have been implemented in the cloud. Orchestration can be defined as a sole, uniform, and centralized approach that accounts for coordinating communications among different applications and services. Thus, orchestration is related to automating the interaction processes among various services, but applies to centralized systems. However, as we have seen, fog uses a decentralized approach with benefits. Moreover, fog has some other characteristics apart from its decentralized approach, such as resource constraints and heterogeneity. Thus, introducing orchestration into the fog paradigm is a challenge. In this context, a novel concept called choreography can be employed. Choreography refers to maintaining a global view of all applications or services across the edge devices through information sharing, thereby updating all local changes to the global view. Thus, choreography uses a distributed approach, performs orchestration-related functionalities, and could be our solution for implementing at the fog–edge interface/link or the southbound region.

Many different orchestration agents have been proposed by different authors with their detailed architecture, which is summed up in the paper by Karima Velasquez et al. [107]. The concept of virtualization has been used by many authors for the same purpose [108], while some authors have also proposed containerization in orchestration in the fog environment [109]. Most authors also propose software-defined networking (SDN) for obtaining centralized control of the network and an automatic management of the global view of the network as soon as any local changes occur. However, Jaeger [110] proposed an orchestration architecture based on network function virtualization (NFV). The general approach by many of the authors has been focused on sub-problems, and very few authors tend to choose a hybrid approach involving both orchestration and choreography, with orchestration at the north-bound region (between the cloud and the fog) and choreography at the southbound region (between the edge devices and the fog). Table 4 shows some related work in the field of orchestration.

### 5.9. Resource Scheduling and Allocation

Cloud computing can be thought of as an infinite warehouse of resources. Still, latency is always an issue due to the plethora of devices connected to the cloud, limited bandwidth available, and geographical separation of the cloud from devices. To tackle this, we introduced the concept of fog computing. However, due to the geographical vicinity of the edge devices, fog nodes are generally less powerful than the cloud. Moreover, they contain limited resources compared to the cloud. This makes achieving maximum throughput through scheduling and resource allocation challenging. When the term resource management is discussed, it is an amalgamation of load balancing, task offloading, resource provisioning, and allocation. Here, we discuss resource scheduling and resource allocation.



**Table 4.** Comparison of various works on orchestration.

Author	Network Architecture	Type	Unique Feature
Zaalouk et al. [111]	SDN-based	Orchestration-focused	Security-oriented, in charge of turning on/off applications that deal with security issues.
Mayoral et al. [108]			Migration of virtual machines between different network domains.
Vilalta et al. [112]			Hierarchical SDN architecture for heterogeneous wireless and optical networks. Also introduces end-to-end provisioning and recovery procedures in a multi-domain network.
Jaeger [110]	NFV-based		Focused on extending the European Telecommunications Standards Institute (ETSI) NFV reference architecture to manage and orchestrate security functions.
Furtado et al. [113]	-	Choreography-focused	Uses middleware for choreography able to automatically deploy and execute services. The middleware is also responsible for monitoring the service composition execution and for performing automatic resource provisioning and service reconfiguration to achieve agreed QoS levels.
Cherrier et al. [114]	SDN-based		Studies the impact of using orchestration and choreography in wireless sensor and actuator networks (WSANs) using mathematical analysis and also application experiments.
Velasquez et al. [107]	-	Hybrid approach	Uses orchestration along with choreography to achieve distributed as well as centralized management simultaneously.

Resource scheduling can be thought of as the optimization of the assignment of various tasks submitted by the edge devices to the fog by meeting the required QoS levels while also ensuring time complexity. Thus, resource scheduling is an optimization solution for scheduling a set of submitted tasks  $T_1, T_2, T_3, \dots, T_k$ , to a set of fog nodes  $F_1, F_2, F_3, \dots, F_p$ , with various QoS requirements, such as cost, time minimization, or the availability and optimization of the optimization function for scheduling time [115]. The resource-scheduling problem is considered an NP-hard problem and uses meta-heuristic algorithms to find feasible and near-optimal solutions in linear time. The resource scheduling approach is divided into three categories based on the time when the scheduling takes place. Static scheduling takes place when tasks simultaneously reach the fog nodes, and the decisions for scheduling are made before submitting tasks [116–119]. Thus, there should be prior knowledge of the demands and available resources for this type of schedule to take place, which is not always the case, as obtaining all knowledge beforehand is not possible in all cases in the fog environment. However, the dynamic scheduling algorithms [76,83,120–124] do not require all prior knowledge, and the scheduling of the tasks takes place after the tasks get submitted to the fog nodes, thus allowing for more flexibility for scheduling algorithms. Some authors [57] used a hybrid approach involving both static and dynamic scheduling according to the use case. Table 5 recapitulates all different scheduling techniques.

**Table 5.** Comparison of resource-scheduling techniques.

Authors	Case Study	Algorithm Used	Performance Measurement	Pros	Cons
Static approaches					
Bitam et al. [118]	General	Bees life algorithm	CPU execution time, Allocated memory	-Managing allocated memory -Low CPU execution time	-Static scheduling -Low scalability
Fan et al. [83]	General	Ant colony optimization	Total profit, Guarantee Ratio	Maximizing profits of fog providers	High time complexity
Rahbari et al. [123]	EAHD application, Intelligent surveillance application	Symbiotic organisms search	Energy utilization, Network usage, Cost	-Minimizing energy utilization -Low execution cost	High execution time
Kabirzadeh et al. [125]	Intelligent surveillance application	Hyper-heuristic based	Energy consumption, Execution time, Network usage, Cost	-Minimizing energy consumption -Low cost and low time	Low scalability
Dynamic approaches					
Sun et al. [117]	Word count	NSGA-II	Service latency, Stability	-Low execution time -High scalability -Low latency	High cost
Cardellini et al. [119]	Word count, Log stream processing	Adaptive-based	Node utilization, Application latency, Inter-node traffic	-Enhancing runtime scheduling -Low Latency -Low execution time	-Low availability -Low scalability -Centralized topology
Zeng et al. [122]	Image Tasks	Heuristic-based	Task completion time	-Low computation complexity -Low response time	-High memory consumption
Chen et al. [67]	Vehicular cloud application	Heuristic-based	Response time, Queue length	-High dynamic efficiency -Using a formal method -Low time	Simple case study
Urgaonkar et al. [126]	Mobile application	Lyapunov optimization	Queue length, Cost	-Reducing state space -Performing a cost-optimal solution	-High cost -Low scalability
Hybrid approaches					
De Benedetti et al. [127]	Distributed robotics application	Adaptive-based	Scalability, Fault tolerance	-High interaction with IoT devices -Low latency -Low execution time	-Low scalability -High cost

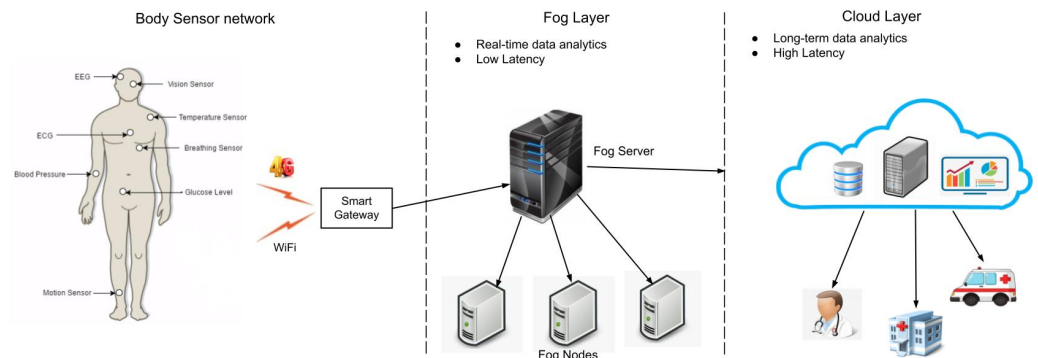
Resource allocation is considered to be an important factor in resource management. Resource allocation in the cloud is different than that in the fog. This is because the cloud is a single cohesive unit with infinite resources, so resource allocation is not a big issue in a cloud environment. Still, the fog nodes are geographically widely distributed and there are limited resources available. As a result, resource allocation in the fog environment is difficult [128] because, in addition to the fog's distribution, there are also QoS requirements for IoT devices that must be met. The approaches to resource allocation can be broadly classified into two categories: auction-based and optimization-based. The auction-based approach [129–132] is similar to a real-time auction, where the IoT devices bid for the available fog nodes and the fog node is sold (allocated) to the highest bidder. A specified auction mechanism is employed to allocate the fog nodes to the IoT resources. The fog nodes listed for bidding are according to the required QoS and other constraints that specific IoT devices require. In the optimization-based technique [58,133–140], a double-matching problem is formulated wherein cloud devices and the fog nodes are coupled for the IoT devices. The problem is considered to be an NP-hard problem for finding an optimal set of fog–cloud pairs for IoT devices while fulfilling various QoS requirements.

## 6. Case Study: Fog Data Analytics in Healthcare

Advancements in technology have significantly benefited the area of healthcare. Recent developments in IoT and healthcare have improved patient healthcare services. Healthcare services are critical; we need instant solutions to problems and real-time patient services. Smart healthcare devices are increasing significantly and generate huge amounts of data that need to be processed and analyzed in real time. Fog computing plays a vital role in managing large volumes of health data and has benefits of a reduced latency, improved energy efficiency, increased reliability, and improved energy efficiency. This case study focused on the health monitoring system using fog computing. Rather than sending the big data generated by the health devices to the cloud layer, they will be processed and analyzed at the fog layer through fog nodes (FNs) to make real-time decisions. Fog computing plays a vital role in healthcare applications: they can decrease the network's data flow and provide an improved latency, security, and preventive care to patients [141].

In this case study, a fog-based healthcare scenario was implemented, where patient health data were collected and transferred to the fog nodes at the fog layer. These data were filtered, pre-processed, and analyzed, and real-time decisions were made for the

better treatment of the patients. Figure 8 represents the IoT-based fog-enabled model for healthcare. Patient data were collected from ECG sensors attached to the patient's body and sent to a smart gateway. Based on the criticality level of the data, the captured data were sent to either the cloud or the fog layer by the smart gateway. The data received at the fog layer will be processed and analyzed to provide timely medical treatment to the patients, reducing latency. If the gathered data are not urgent, they will be sent to the cloud for further analysis and long-term storage. The patient's health information is accessible remotely by family members and medical personnel from both the fog and cloud layers.



**Figure 8.** Fog data analytics in healthcare.

The performance of the proposed fog computing model was analyzed through a simulation and experiments using the iFogSim simulator. The simulation measures latency and network utilization in cloud and fog computing environments. The physical topology in iFogSim includes various ECG sensors, fog devices, and cloud servers. The simulation was conducted with five different physical topology configurations—conf1, conf2, conf3, conf4, and conf5—with monitoring devices 4, 8, 16, 32, and 64, respectively, and measured the latency and network utilization. The monitoring devices used in the configurations had a CPU length of 1200 million instructions and a network length of 20,000 bytes. Tables 6 and 7 present the simulation results of latency and network utilization, respectively, for each configuration in the fog and cloud environment.

**Table 6.** Comparison of latency.

Physical Topology	Latency (ms)	
	Cloud Layer	Fog Layer
Conf-1	221.32	56.32
Conf-2	248.91	64.20
Conf-3	282.45	66.12
Conf-4	1352.67	70.65
Conf-5	3452.77	90.78

**Table 7.** Comparison of network utilization.

Physical Topology	Network Utilization (KBs)	
	Cloud Layer	Fog Layer
Conf-1	151.23	40.43
Conf-2	168.43	43.95
Conf-3	334.78	51.56
Conf-4	890.67	108.37
Conf-5	1098.04	175.46

Figures 9 and 10 show a graphical comparison of the results for the different configurations. The simulation results show that using fog computing will improve the latency and the network utilization. Fog computing improves the quality of service of the complete healthcare system.

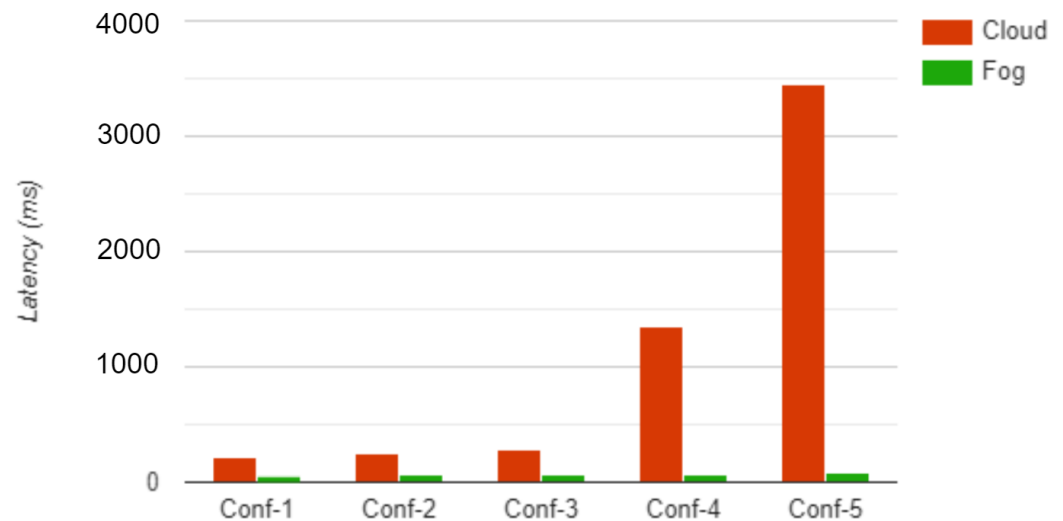


Figure 9. Latency.

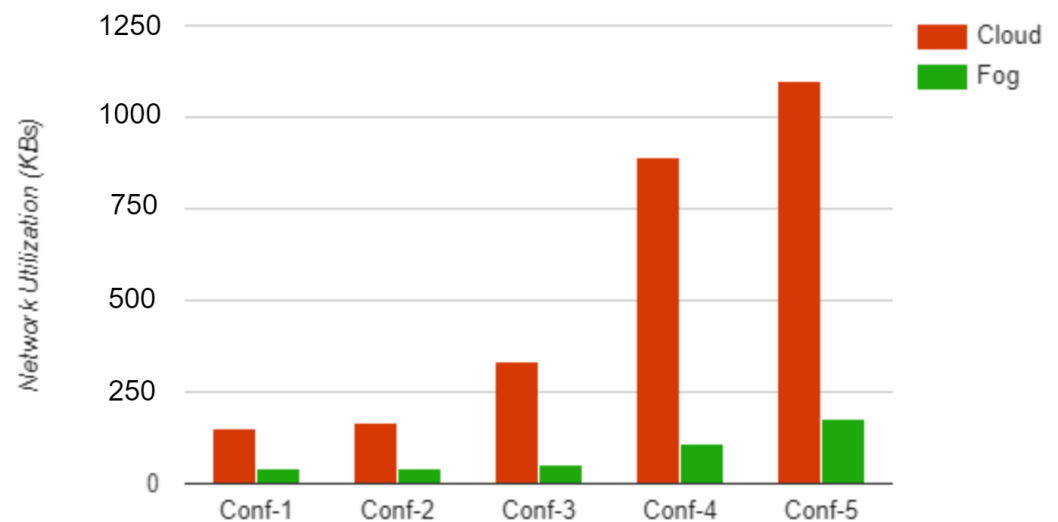
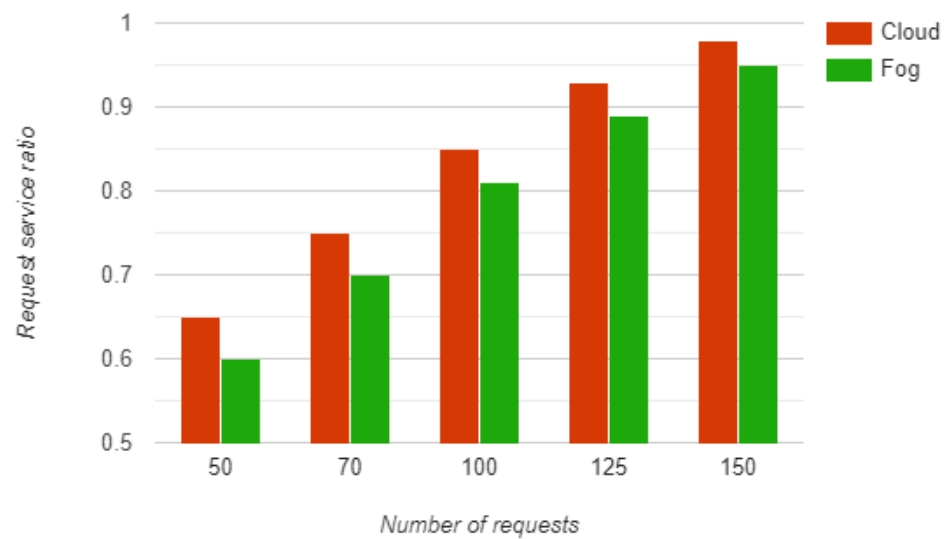


Figure 10. Network utilization.

Fog devices have limited resources compared to the cloud. We evaluated the request service ratio in both the fog and cloud environment. The simulation results show that the cloud environment has the better request service ratio compared to the fog environment due to the limited computational capability of fog devices. Figure 11 shows a graphical comparison of the request service ratio in the fog and cloud environment for a different number of requests.



**Figure 11.** Request service ratio.

## 7. Conclusions

The exponential growth in sensors and smart devices produces significant heterogeneous data. To manage such data, we need efficient solutions deployed near the devices. Fog computing, which works as a middle layer between the cloud layer and the IoT devices, is a solution to the problem of real-time data delivery, especially in critical applications such as e-healthcare systems. Fog data analytics is an emerging solution for handling the huge amount of data produced by smart IoT devices. This paper gave an overview of fog computing, the need for fog computing in IoT, data analytics, and the need for fog data analytics in IoT. The current state-of-the-art in fog data analytics in IoT with various use cases was also covered in this paper. Furthermore, research challenges in processing big data in IoT networks were discussed. Finally, a case study on fog data analytics in healthcare was presented, along with an experimental analysis using fog and cloud computing. The future work motivates researchers to carry out an in-depth review of current state-of-the-art techniques for adopting security in fog data analytics.

**Author Contributions:** Conceptualization: J.B., K.I., M.K., U.C. and S.T.; writing—original draft preparation: K.J., U.C., M.B. and R.S.; methodology: K.J., S.T., D.L.M., M.V. and R.S.; writing—review and editing: S.T., M.S.R., D.L.M., M.V. and R.S.; investigation: U.C., J.B., M.K., M.B. and M.S.R.; supervision: S.T., M.B., M.K. and U.C.; visualization: S.T., D.L.M., M.V. and D.L.M.; software: K.I., S.T., M.K. and M.S.R. All authors have read and agreed to the published version of the manuscript.

**Funding:** This paper was partially supported by UEFISCDI Romania and MCI through BEIA projects SOLID-B5G, T4ME2, DISAVIT, AISTOR, MULTI-AI, ADRIATIC, Hydro3D, PREVENTION, DAFCC, EREMI, ADCATER, MUSEION, iPREMAS, IPSUS, U-GARDEN, CREATE and by European Union’s Horizon 2020 research and innovation program under grant agreement No. 101037866 (ADMA Trans4MErs) and No. 101016567 (VITAL-5G). The results were obtained with the support of the Ministry of Investments and European Projects through the Human Capital Sectoral Operational Program 2014-2020, Contract no. 62461/03.06.2022, SMIS code 153735. This work is supported by Ministry of Research, Innovation, Digitization from Romania by the National Plan of R & D, Project PN 19 11, Subprogram 1.1. Institutional performance-Projects to finance excellence in RDI, Contract No. 19PFE/30.12.2021 and a grant of the National Center for Hydrogen and Fuel Cells (CNHPC)—Installations and Special Objectives of National Interest (IOSIN).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** No data are associated with this research work.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Atzori, L.; Iera, A.; Morabito, G. The internet of things: A survey. *Comput. Netw.* **2010**, *54*, 2787–2805. [CrossRef]
2. Nick, G. How Many IoT Devices Are There In 2022? More Than Ever! Available online: <https://techjury.net/blog/how-many-iot-devices-are-there/> (accessed on 24 November 2022).
3. Farahani, B.; Firouzi, F.; Chang, V.; Badaroglu, M.; Constant, N.; Mankodiya, K. Towards fog-driven IoT eHealth: Promises and challenges of IoT in medicine and healthcare. *Future Gener. Comput. Syst.* **2018**, *78*, 659–676. [CrossRef]
4. Sadhu, P.K.; Yanambaka, V.P.; Abdelgawad, A. Internet of Things: Security and Solutions Survey. *Sensors* **2022**, *22*, 7433. [CrossRef] [PubMed]
5. Salih, K.O.M.; Rashid, T.A.; Radovanovic, D.; Bacanin, N. A Comprehensive Survey on the Internet of Things with the Industrial Marketplace. *Sensors* **2022**, *22*, 730. [CrossRef] [PubMed]
6. Tresa, M.; Francina, S.; Jerlin Oviya, V.; Lavanya, K. A study on internet of things: Overview, automation, wireless technology, robotics. *Ann. Rom. Soc. Cell Biol.* **2021**, *25*, 6546–6555.
7. Romeo, L.; Petitti, A.; Marani, R.; Milella, A. Internet of Robotic Things in Smart Domains: Applications and Challenges. *Sensors* **2020**, *20*, 3355. [CrossRef] [PubMed]
8. Kumari, A.; Gupta, R.; Tanwar, S. Introduction. In *Fog Data Analytics for IoT Applications: Next Generation Process Model with State of the Art Technologies*; Tanwar, S., Ed.; Springer Singapore: Singapore, 2020; pp. 3–17. [CrossRef]
9. Abosata, N.; Al-Rubaye, S.; Inalhan, G.; Emmanouilidis, C. Internet of Things for System Integrity: A Comprehensive Survey on Security, Attacks and Countermeasures for Industrial Applications. *Sensors* **2021**, *21*, 3654. [CrossRef]
10. Goudarzi, M.; Wu, H.; Palaniswami, M.; Buyya, R. An Application Placement Technique for Concurrent IoT Applications in Edge and Fog Computing Environments. *IEEE Trans. Mob. Comput.* **2021**, *20*, 1298–1311. [CrossRef]
11. Tanwar, S.; Vora, J.; Kaneriyi, S.; Tyagi, S.; Kumar, N.; Sharma, V.; You, I. Human Arthritis Analysis in Fog Computing Environment Using Bayesian Network Classifier and Thread Protocol. *IEEE Consum. Electron. Mag.* **2020**, *9*, 88–94. [CrossRef]
12. Tanwar, S.; Tyagi, S.; Budhiraja, I.; Kumar, N. Tactile Internet for Autonomous Vehicles: Latency and Reliability Analysis. *IEEE Wirel. Commun.* **2019**, *26*, 66–72. [CrossRef]
13. Reebadiya, D.; Rathod, T.; Gupta, R.; Tanwar, S.; Kumar, N. Blockchain-based secure and intelligent sensing scheme for autonomous vehicles activity tracking beyond 5g networks. *Peer- Netw. Appl.* **2021**, *14*, 2757–2774. [CrossRef]
14. Naha, R.K.; Garg, S.; Georgakopoulos, D.; Jayaraman, P.P.; Gao, L.; Xiang, Y.; Ranjan, R. Fog Computing: Survey of trends, architectures, requirements, and research directions. *IEEE Access* **2018**, *6*, 47980–48009. [CrossRef]
15. Okay, F.Y.; Ozdemir, S. Routing in fog-enabled IoT platforms: A survey and an SDN-based solution. *IEEE Internet Things J.* **2018**, *5*, 4871–4889. [CrossRef]
16. Kim, T.; Yoo, S.E.; Kim, Y. Edge/Fog Computing Technologies for IoT Infrastructure. *Sensors* **2021**, *21*, 3001. [CrossRef] [PubMed]
17. Atlam, H.F.; Walters, R.J.; Wills, G.B. Fog Computing and the Internet of Things: A Review. *Big Data Cogn. Comput.* **2018**, *2*, 10. [CrossRef]
18. Prasad, V.K.; Bhavsar, M.D.; Tanwar, S. Influence of Monitoring: Fog and Edge Computing. *Scalable Comput. Pract. Exp.* **2019**, *20*, 365–376. [CrossRef]
19. Ahmed, M.; Mumtaz, R.; Zaidi, S.M.H.; Hafeez, M.; Zaidi, S.A.R.; Ahmad, M. Distributed Fog Computing for Internet of Things (IoT) Based Ambient Data Processing and Analysis. *Electronics* **2020**, *9*, 1756. [CrossRef]
20. Oma, R.; Nakamura, S.; Enokido, T.; Takizawa, M. A tree-based model of energy-efficient fog computing systems in IoT. In Proceedings of the 12th International Conference on Complex, Intelligent, and Software Intensive Systems (CISIS-2018), Matsue, Japan, 4–6 July 2018; pp. 991–1001.
21. Asghar, A.; Abbas, A.; Khattak, H.A.; Khan, S.U. Fog Based Architecture and Load Balancing Methodology for Health Monitoring Systems. *IEEE Access* **2021**, *9*, 96189–96200. [CrossRef]
22. Sobacki, A.; Szymański, J.; Gil, D.; Mora, H. Deep learning in the fog. *Int. J. Distrib. Sens. Netw.* **2019**, *15*, 1550147719867072. [CrossRef]
23. Taneja, M.; Jalodia, N.; Byabazaire, J.; Davy, A.; Olariu, C. SmartHerd management: A microservices-based fog computing-assisted IoT platform towards data-driven smart dairy farming. *Softw. Pract. Exp.* **2019**, *49*, 1055–1078. [CrossRef] [PubMed]
24. Bhamare, D.; Samaka, M.; Erbad, A.; Jain, R.; Gupta, L. Exploring microservices for enhancing internet QoS. *Trans. Emerg. Telecommun. Technol.* **2018**, *29*, e3445. [CrossRef]
25. Wöbker, C.; Seitz, A.; Mueller, H.; Bruegge, B. Fogernetes: Deployment and management of fog computing applications. In Proceedings of the NOMS 2018—2018 IEEE/IFIP Network Operations and Management Symposium, Taipei, Taiwan, 23–27 April 2018; pp. 1–7.
26. Mehdipour, F.; Noori, H.; Javadi, B. Energy-efficient big data analytics in datacenters. In *Advances in Computers*; Elsevier: Amsterdam, The Netherlands, 2016; Volume 100, pp. 59–101.



27. Vassakis, K.; Petrakis, E.; Kopanakis, I. Big data analytics: Applications, prospects and challenges. In *Mobile Big Data*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 3–20.
28. Badidi, E.; Mahrez, Z.; Sabir, E. Fog Computing for Smart Cities' Big Data Management and Analytics: A Review. *Future Internet* **2020**, *12*, 190. [[CrossRef](#)]
29. Taneja, M.; Jalodia, N.; Davy, A. Distributed Decomposed Data Analytics in Fog Enabled IoT Deployments. *IEEE Access* **2019**, *7*, 40969–40981. [[CrossRef](#)]
30. Zhao, L. Privacy-Preserving Distributed Analytics in Fog-Enabled IoT Systems. *Sensors* **2020**, *20*, 6153. [[CrossRef](#)] [[PubMed](#)]
31. Gautam, K.; Puri, V.; Tromp, J.G.; Nguyen, N.G.; Van Le, C. Internet of Things (IoT) and Deep Neural Network-Based Intelligent and Conceptual Model for Smart City. In *Frontiers in Intelligent Computing: Theory and Applications*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 287–300.
32. Rokade, A.; Singh, M.; Malik, P.K.; Singh, R.; Alsuwian, T. Intelligent Data Analytics Framework for Precision Farming Using IoT and Regressor Machine Learning Algorithms. *Appl. Sci.* **2022**, *12*, 9992. [[CrossRef](#)]
33. Srinivas, K.; Dua, M. Fog Computing and Deep CNN Based Efficient Approach to Early Forest Fire Detection with Unmanned Aerial Vehicles. In Proceedings of the International Conference on Inventive Computation Technologies, Coimbatore, Tamil Nadu, 29–30 August 2019; pp. 646–652.
34. Devarajan, M.; Ravi, L. Intelligent cyber-physical system for an efficient detection of Parkinson disease using fog computing. *Multimed. Tools Appl.* **2019**, *78*, 32695–32719. [[CrossRef](#)]
35. Tuli, S.; Mahmud, R.; Tuli, S.; Buyya, R. Fogbus: A blockchain-based lightweight framework for edge and fog computing. *J. Syst. Softw.* **2019**, *154*, 22–36. [[CrossRef](#)]
36. Podsevalov, I.; Iakushkin, O.; Kurbangaliev, R.; Korkhov, V. Blockchain as a Platform for Fog Computing. In Proceedings of the International Conference on Computational Science and Its Applications, Saint Petersburg, Russia, 1–4 July 2019; pp. 596–605.
37. Li, M.; Zhu, L.; Lin, X. CoRide: A Privacy-Preserving Collaborative-Ride Hailing Service Using Blockchain-Assisted Vehicular Fog Computing. In Proceedings of the International Conference on Security and Privacy in Communication Systems, Orlando, FL, USA, 23–25 October 2019; pp. 408–422.
38. Kadhim, A.J.; Seno, S.A.H. Energy-efficient multicast routing protocol based on SDN and fog computing for vehicular networks. *Ad Hoc Netw.* **2019**, *84*, 68–81. [[CrossRef](#)]
39. Amadeo, M.; Campolo, C.; Ruggeri, G.; Molinaro, A.; Iera, A. Towards Software-defined Fog Computing via Named Data Networking. In Proceedings of the IEEE INFOCOM 2019—IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Paris, France, 29 April–2 May 2019; pp. 133–138.
40. Bhatia, M.; Sood, S.K.; Kaur, S. Quantumized approach of load scheduling in fog computing environment for IoT applications. *Computing* **2020**, *102*, 1097–1115. [[CrossRef](#)]
41. Maurer, I.; Riva, M.; Hansen, C.; Ortmaier, T. Cloud-based plant and process monitoring based on a modular and scalable data analytics infrastructure. In *Tagungsband des 2. Kongresses Montage Handhabung Industrieroboter*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 33–42.
42. Chalapathi, G.; Chamola, V.; Vaish, A.; Buyya, R. Industrial Internet of Things (IIoT) Applications of Edge and Fog Computing: A Review and Future Directions. *arXiv* **2019**, arXiv:1912.00595.
43. Rocha, E.; Monteiro, K.; Silva, E.; Santos, G.L.; Santos, W.; Endo, P.T. Detecting epileptic seizures using deep learning with cloud and fog computing. In Proceedings of the 2018 IEEE/ACM International Conference on Utility and Cloud Computing Companion (UCC Companion), Zurich, Switzerland, 17–20 December 2018; pp. 19–20.
44. Tuli, S.; Basumatary, N.; Gill, S.S.; Kahani, M.; Arya, R.C.; Wander, G.S.; Buyya, R. HealthFog: An ensemble deep learning based Smart Healthcare System for Automatic Diagnosis of Heart Diseases in integrated IoT and fog computing environments. *Future Gener. Comput. Syst.* **2020**, *104*, 187–200. [[CrossRef](#)]
45. Rajan, J.P.; Rajan, S.E.; Martis, R.J.; Panigrahi, B. Fog Computing Employed Computer Aided Cancer Classification System Using Deep Neural Network in Internet of Things Based Healthcare System. *J. Med. Syst.* **2020**, *44*, 34. [[CrossRef](#)] [[PubMed](#)]
46. Vora, J.; Tanwar, S.; Tyagi, S.; Kumar, N.; Rodrigues, J.J. FAAL: Fog computing-based patient monitoring system for ambient assisted living. In Proceedings of the 2017 IEEE 19th International Conference on e-Health Networking, Applications and Services (Healthcom), Dalian, China, 12–15 October 2017; pp. 1–6.
47. Malathi, D.; Subramaniaswamy, V.; Vijayakumar, V.; Logesh, R. Fog-assisted personalized healthcare-support system for remote patients with diabetes. *J. Ambient. Intell. Humaniz. Comput.* **2019**, *10*, 3747–3760.
48. Aazam, M.; Fernando, X. Fog Assisted Driver Behavior Monitoring for Intelligent Transportation System. In Proceedings of the 2017 IEEE 86th Vehicular Technology Conference (VTC-Fall), Toronto, ON, Canada, 24–27 September 2017; pp. 1–5. [[CrossRef](#)]
49. Javed, M.A.; Nafi, N.S.; Basheer, S.; Bivi, M.A.; Bashir, A.K. Fog-Assisted Cooperative Protocol for Traffic Message Transmission in Vehicular Networks. *IEEE Access* **2019**, *7*, 166148–166156. [[CrossRef](#)]
50. Bhatia, J.; Modi, Y.; Tanwar, S.; Bhavsar, M. Software defined vehicular networks: A comprehensive review. *Int. J. Commun. Syst.* **2019**, *32*, e4005. [[CrossRef](#)]
51. Ning, Z.; Huang, J.; Wang, X. Vehicular fog computing: Enabling real-time traffic management for smart cities. *IEEE Wirel. Commun.* **2019**, *26*, 87–93. [[CrossRef](#)]
52. Melnik, E.; Orda-Zhigulina, M.; Orda-Zhigulina, D.; Ivanov, D.; Rodina, A. Fog computing in new approach for monitoring of hazardous phenomena. *J. Phys. Conf. Ser.* **2019**, *1333*, 072016. [[CrossRef](#)]

53. Khan, S.; Muhammad, K.; Mumtaz, S.; Baik, S.W.; de Albuquerque, V.H.C. Energy-efficient deep CNN for smoke detection in foggy IoT environment. *IEEE Internet Things J.* **2019**, *6*, 9237–9245. [[CrossRef](#)]
54. Magurawalage, C.S.; Yang, K.; Wang, K. Aqua computing: Coupling computing and communications. *arXiv* **2015**, arXiv:1510.07250.
55. Naranjo, P.G.V.; Shojafar, M.; Mostafaei, H.; Pooranian, Z.; Baccarelli, E. P-SEP: A prolong stable election routing algorithm for energy-limited heterogeneous fog-supported wireless sensor networks. *J. Supercomput.* **2017**, *73*, 733–755. [[CrossRef](#)]
56. Singh, S.; Yassine, A. IoT Big Data Analytics with Fog Computing for Household Energy Management in Smart Grids. In Proceedings of the International Conference on Smart Grid and Internet of Things, Niagara Falls, ON, Canada, 11 July 2018; pp. 13–22.
57. Mahmud, R.; Buyya, R. Modelling and simulation of fog and edge computing environments using iFogSim toolkit. In *Fog and Edge Computing: Principles and Paradigms*; Wiley Press: New York, NY, USA, 2019; pp. 1–35.
58. Do, C.T.; Tran, N.H.; Pham, C.; Alam, M.G.R.; Son, J.H.; Hong, C.S. A proximal algorithm for joint resource allocation and minimizing carbon footprint in geo-distributed fog computing. In Proceedings of the 2015 International Conference on Information Networking (ICOIN), Cambodia, 12–14 January 2015; pp. 324–329.
59. Choi, J.; Ahn, S. Scalable Service Placement in the Fog Computing Environment for the IoT-Based Smart City. *J. Inf. Process. Syst.* **2019**, *15*, 440–448.
60. Capra, M.; Peloso, R.; Masera, G.; Ruo Roch, M.; Martina, M. Edge Computing: A Survey On the Hardware Requirements in the Internet of Things World. *Future Internet* **2019**, *11*, 100. [[CrossRef](#)]
61. Wei, W.; Qi, Y. Information potential fields navigation in wireless Ad-Hoc sensor networks. *Sensors* **2011**, *11*, 4794–4807. [[CrossRef](#)] [[PubMed](#)]
62. Barik, R.K.; Priyadarshini, R.; Dubey, H.; Kumar, V.; Mankodiya, K. FogLearn: Leveraging fog-based machine learning for smart system big data analytics. In *Geospatial Intelligence: Concepts, Methodologies, Tools, and Applications*; IGI Global: Hershey, PA, USA, 2019; pp. 1225–1241.
63. Mohammadi, M.; Al-Fuqaha, A.; Sorour, S.; Guizani, M. Deep learning for IoT big data and streaming analytics: A survey. *IEEE Commun. Surv. Tutor.* **2018**, *20*, 2923–2960. [[CrossRef](#)]
64. Fei, X.; Shah, N.; Verba, N.; Chao, K.M.; Sanchez-Anguix, V.; Lewandowski, J.; James, A.; Usman, Z. CPS data streams analytics based on machine learning for Cloud and Fog Computing: A survey. *Future Gener. Comput. Syst.* **2019**, *90*, 435–450. [[CrossRef](#)]
65. Su, J.; Lin, F.; Zhou, X.; Lu, X. Steiner tree based optimal resource caching scheme in fog computing. *China Commun.* **2015**, *12*, 161–168. [[CrossRef](#)]
66. Huang, C.Y.; Xu, K. Reliable realtime streaming in vehicular cloud-fog computing networks. In Proceedings of the 2016 IEEE/CIC International Conference on Communications in China (ICCC), Chengdu, China, 27–29 July 2016; pp. 1–6.
67. Chen, Z.; Hu, W.; Wang, J.; Zhao, S.; Amos, B.; Wu, G.; Ha, K.; Elgazzar, K.; Pillai, P.; Klatzky, R.; et al. An empirical study of latency in an emerging class of edge computing applications for wearable cognitive assistance. In Proceedings of the Second ACM/IEEE Symposium on Edge Computing, San Jose, CA, USA, 12–14 October 2017; pp. 1–14.
68. Aazam, M.; St-Hilaire, M.; Lung, C.H.; Lambadaris, I. MeFoRE: QoE based resource estimation at Fog to enhance QoS in IoT. In Proceedings of the 2016 23rd International Conference on Telecommunications (ICT), Thessaloniki, Greece, 16–18 May 2016; pp. 1–5.
69. Prazeres, C.; Serrano, M. Soft-iot: Self-organizing fog of things. In Proceedings of the 2016 30th International Conference on Advanced Information Networking and Applications Workshops (WAINA), Crans-Montana, Switzerland, 23–25 March 2016; pp. 803–808.
70. He, J.; Wei, J.; Chen, K.; Tang, Z.; Zhou, Y.; Zhang, Y. Multitier fog computing with large-scale IoT data analytics for smart cities. *IEEE Internet Things J.* **2017**, *5*, 677–686. [[CrossRef](#)]
71. Darwish, T.S.; Bakar, K.A. Fog based intelligent transportation big data analytics in the internet of vehicles environment: Motivations, architecture, challenges, and critical issues. *IEEE Access* **2018**, *6*, 15679–15701. [[CrossRef](#)]
72. Nahri, M.; Boulmakoul, A.; Karim, L.; Lbath, A. IoV distributed architecture for real-time traffic data analytics. *Procedia Comput. Sci.* **2018**, *130*, 480–487. [[CrossRef](#)]
73. Craciunescu, R.; Mihovska, A.; Mihaylov, M.; Kyriazakos, S.; Prasad, R.; Halunga, S. Implementation of Fog computing for reliable E-health applications. In Proceedings of the 2015 49th Asilomar Conference On Signals, Systems and Computers, Pacific Grove, CA, USA, 8–11 November 2015; pp. 459–463.
74. Dantu, R.; Dissanayake, I.; Nerur, S. Exploratory Analysis of Internet of Things (IoT) in Healthcare: A Topic Modelling and Co-citation Approaches. *Inf. Syst. Manag.* **2021**, *38*, 62–78. [[CrossRef](#)]
75. Rani, S.; Talwar, R.; Malhotra, J.; Ahmed, S.; Sarkar, M.; Song, H. A novel scheme for an energy efficient Internet of Things based on wireless sensor networks. *Sensors* **2015**, *15*, 28603–28626. [[CrossRef](#)] [[PubMed](#)]
76. Gupta, H.; Vahid Dastjerdi, A.; Ghosh, S.K.; Buyya, R. iFogSim: A toolkit for modeling and simulation of resource management techniques in the Internet of Things, Edge and Fog computing environments. *Softw. Pract. Exp.* **2017**, *47*, 1275–1296. [[CrossRef](#)]
77. Zhu, J.; Chan, D.; Prabhu, M.; Natarajan, P.; Hu, H.; Bonomi, F. Improving web sites performance using edge servers in fog computing architecture. In Proceedings of the 2013 IEEE Seventh International Symposium On Service-Oriented System Engineering, San Francisco, CA, USA, 25–28 March 2013; pp. 320–323.

78. Hu, W.; Gao, Y.; Ha, K.; Wang, J.; Amos, B.; Chen, Z.; Pillai, P.; Satyanarayanan, M. Quantifying the impact of edge computing on mobile applications. In Proceedings of the 7th ACM SIGOPS Asia-Pacific Workshop on Systems, Hong Kong, 4–5 August 2016; pp. 1–8.
79. Hao, Y.; Helo, P. The role of wearable devices in meeting the needs of cloud manufacturing: A case study. *Robot. Comput.-Integr. Manuf.* **2017**, *45*, 168–179. [[CrossRef](#)]
80. Mubeen, S.; Nikolaidis, P.; Didic, A.; Pei-Breivold, H.; Sandström, K.; Behnam, M. Delay mitigation in offloaded cloud controllers in industrial IoT. *IEEE Access* **2017**, *5*, 4418–4430. [[CrossRef](#)]
81. Shih, Y.; Chung, W.; Pang, A.; Chiu, T.; Wei, H. Enabling low-latency applications in fog-radio access networks. *IEEE Netw.* **2016**, *31*, 52–58. [[CrossRef](#)]
82. Flores, H.; Sharma, R.; Ferreira, D.; Luo, C.; Kostakos, V.; Tarkoma, S.; Hui, P.; Li, Y. Social-aware device-to-device communication: a contribution for edge and fog computing? In Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct, Heidelberg, Germany, 12–16 September 2016; pp. 1466–1471.
83. Fan, J.; Wei, X.; Wang, T.; Lan, T.; Subramaniam, S. Deadline-aware task scheduling in a tiered IoT infrastructure. In Proceedings of the GLOBECOM 2017—2017 IEEE Global Communications Conference, Singapore, 4–8 December 2017; pp. 1–7.
84. Ye, D.; Wu, M.; Tang, S.; Yu, R. Scalable fog computing with service offloading in bus networks. In Proceedings of the 2016 IEEE 3rd International Conference on Cyber Security and Cloud Computing (CSCloud), Beijing, China, 25–27 June 2016; pp. 247–251.
85. Tseng, F.H.; Tsai, M.S.; Tseng, C.W.; Yang, Y.T.; Liu, C.C.; Chou, L.D. A lightweight autoscaling mechanism for fog computing in industrial applications. *IEEE Trans. Ind. Inform.* **2018**, *14*, 4529–4537. [[CrossRef](#)]
86. Manzoor, A.; Shah, M.A.; Khattak, H.A.; Din, I.U.; Khan, M.K. Multi-tier authentication schemes for fog computing: Architecture, security perspective, and challenges. *Int. J. Commun. Syst.* **2019**, *35*, e4033. [[CrossRef](#)]
87. Yang, Y.; Wu, L.; Yin, G.; Li, L.; Zhao, H. A survey on security and privacy issues in Internet-of-Things. *IEEE Internet Things J.* **2017**, *4*, 1250–1258. [[CrossRef](#)]
88. Kakanakov, N.; Shopov, M. Adaptive models for security and data protection in IoT with Cloud technologies. In Proceedings of the 2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, Croatia, 22–26 May 2017; pp. 1001–1004.
89. Alharbi, S.; Rodriguez, P.; Maharaja, R.; Iyer, P.; Subaschandrabose, N.; Ye, Z. Secure the internet of things with challenge response authentication in fog computing. In Proceedings of the 2017 IEEE 36th International Performance Computing and Communications Conference (IPCCC), San Diego, CA, USA, 10–12 December 2017; pp. 1–2.
90. Singh, A.; Chatterjee, K. A secure multi-tier authentication scheme in cloud computing environment. In Proceedings of the 2015 International Conference on Circuits, Power and Computing Technologies [ICCPCT-2015], Nagercoil, India, 19–20 March 2015; pp. 1–7.
91. Muzammal, S.M.; Shah, M.A.; Khattak, H.A.; Jabbar, S.; Ahmed, G.; Khalid, S.; Hussain, S.; Han, K. Counter measuring conceivable security threats on smart healthcare devices. *IEEE Access* **2018**, *6*, 20722–20733. [[CrossRef](#)]
92. Champakamala, S.; Anjan, P.; Karunakar, P. Privacy Preserving Multifactor authentication schemes in a Distributed Environment: An Overview. *Int. J. Innov. Res. Stud.* **2013**, *2*, 123–131.
93. Banyal, R.; Jain, V.; Jain, P. Data Management System to Improve Security and Availability in Cloud Storage. In Proceedings of the 2015 International Conference On Computational Intelligence And Networks, Odisha, India, 12–13 January 2015; pp. 124–129.
94. Emam, A. Additional authentication and authorization using registered email-ID for cloud computing. *Int. J. Soft Comput. Eng.* **2013**, *3*, 110–113.
95. Kumar, S.; Ganpati, A. Multi-authentication for cloud security: A framework. *Int. J. Comput. Sci. Eng. Technol.* **2014**, *5*, 295–303
96. Abdurrahman, U.; Kaiiali, M.; Muhammad, J. A new mobile-based multi-factor authentication scheme using pre-shared number, GPS location and time stamp. In Proceedings of the 2013 International Conference On Electronics, Computer And Computation (ICECCO), Ankara, Turkey, 7–9 November 2013; pp. 293–296.
97. Liu, S.; Hu, S.; Weng, J.; Zhu, S.; Chen, Z. A novel asymmetric three-party based authentication scheme in wearable devices environment. *J. Netw. Comput. Appl.* **2016**, *60*, 144–154. [[CrossRef](#)]
98. Ahmad, S.; Ali, B.; Adnan, W. Technical issues and challenges of biometric applications as access control tools of information security. *Int. J. Innov. Comput. Inf. Control* **2012**, *8*, 7983–7999.
99. Soni, A.; Pachouri, R.; Jain, A. Pake protocol with OTSP and image based password authentication. *Int. J. Adv. Res. Comput. Sci.* **2018**, *9*.
100. Dhamija, A.; Dhamija, D. SSQS and OTP based user authentication mechanism in cloud computing. *Inf. Res.* **2016**, *3*, 2431–2437.
101. Memon, S.; Maheswaran, M. Using machine learning for handover optimization in vehicular fog computing. In Proceedings of the Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing, Limassol, Cyprus, 8–12 April 2019; pp. 182–190.
102. Zeydan, E.; Bastug, E.; Bennis, M.; Kader, M.A.; Karatepe, I.A.; Er, A.S.; Debbah, M. Big data caching for networking: Moving from cloud to edge. *IEEE Commun. Mag.* **2016**, *54*, 36–42. [[CrossRef](#)]
103. Bastug, E.; Bennis, M.; Debbah, M. Living on the edge: The role of proactive caching in 5G wireless networks. *IEEE Commun. Mag.* **2014**, *52*, 82–89. [[CrossRef](#)]
104. Jiang, Y.; Ma, M.; Bennis, M.; Zheng, F.; You, X. A novel caching policy with content popularity prediction and user preference learning in Fog-RAN. In Proceedings of the 2017 IEEE Globecom Workshops (GC Wkshps), Singapore, 4–8 December 2017; pp. 1–6.

105. Mann, Z.Á. Optimization problems in fog and edge computing. In *Fog and Edge Computing: Principles and Paradigms*; Wiley: Hoboken, NJ, USA, 2019.
106. Gao, X.; Huang, X.; Bian, S.; Shao, Z.; Yang, Y. Pora: Predictive offloading and resource allocation in dynamic fog computing systems. *IEEE Internet Things J.* **2019**, *7*, 72–87. [[CrossRef](#)]
107. Velasquez, K.; Abreu, D.P.; Gonçalves, D.; Bittencourt, L.; Curado, M.; Monteiro, E.; Madeira, E. Service orchestration in fog environments. In Proceedings of the 2017 IEEE 5th International Conference on Future Internet of Things and Cloud (FiCloud), Prague, Czech Republic, 21–23 August 2017; pp. 329–336.
108. Mayoral, A.; Vilalta, R.; Muñoz, R.; Casellas, R.; Martínez, R. Experimental seamless virtual machine migration using an integrated SDN IT and network orchestrator. In Proceedings of the 2015 Optical Fiber Communications Conference and Exhibition (OFC), Los Angeles, CA, USA, 22–26 March 2015; pp. 1–3.
109. Hoque, S.; de Brito, M.S.; Willner, A.; Keil, O.; Magedanz, T. Towards container orchestration in fog computing infrastructures. In Proceedings of the 2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC), Turin, Italy, 4–8 July 2017; Volume 2, pp. 294–299.
110. Jaeger, B. Security orchestrator: Introducing a security orchestrator in the context of the etsi nfv reference architecture. In Proceedings of the 2015 IEEE Trustcom/BigDataSE/ISPA, Helsinki, Finland, 20–22 August 2015; Volume 1, pp. 1255–1260.
111. Zaalouk, A.; Khondoker, R.; Marx, R.; Bayarou, K. OrchSec: An orchestrator-based architecture for enhancing network-security using Network Monitoring and SDN Control functions. In Proceedings of the 2014 IEEE Network Operations And Management Symposium (NOMS), Krakow, Poland, 5–9 May 2014; pp. 1–9.
112. Vilalta, R.; Mayoral, A.; Baranda, J.; Nunez, J.; Casellas, R.; Martinez, R.; Mangués-Bafalluy, J.; Munoz, R. Hierarchical SDN orchestration of wireless and optical networks with E2E provisioning and recovery for future 5G networks. In Proceedings of the 2016 Optical Fiber Communications Conference and Exhibition (OFC), Anaheim, CA, USA, 20–24 March 2016; pp. 459–463.
113. Furtado, T.; Francesquini, E.; Lago, N.; Kon, F. A Middleware for Reflective Web Service Choreographies on the Cloud, (2014,12). In Proceedings of the 13th Workshop on Adaptive and Reflective Middleware, Bordeaux, France, 8–12 December 2014; Volume 9, pp. 1–9. [[CrossRef](#)]
114. Cherrier, S.; Ghamri-Doudane, Y.; Lohier, S.; Roussel, G. Services Collaboration in Wireless Sensor and Actuator Networks: Orchestration versus Choreography. In Proceedings of the International Symposium on Computers and Communications, Cappadocia, Turkey, 1–4 July 2012; Volume 8.
115. Ghobaei-Arani, M.; Souri, A.; Rahmani, A.A. Resource management approaches in fog computing: A comprehensive review. *J. Grid Comput.* **2019**, *18*, 1–42. [[CrossRef](#)]
116. Lin, C.C.; Yang, J.W. Cost-efficient deployment of fog computing systems at logistics centers in industry 4.0. *IEEE Trans. Ind. Inform.* **2018**, *14*, 4603–4611. [[CrossRef](#)]
117. Sun, Y.; Lin, F.; Xu, H. Multi-objective optimization of resource scheduling in fog computing using an improved nsga-ii. *Wirel. Pers. Commun.* **2018**, *102*, 1369–1385. [[CrossRef](#)]
118. Bitam, S.; Zeadally, S.; Mellouk, A. Fog computing job scheduling optimization based on bees swarm. *Enterp. Inf. Syst.* **2018**, *12*, 373–397. [[CrossRef](#)]
119. Cardellini, V.; Grassi, V.; Presti, F.L.; Nardelli, M. On QoS-aware scheduling of data stream applications over fog computing infrastructures. In Proceedings of the 2015 IEEE Symposium on Computers and Communication (ISCC), Larnaca, Cyprus, 6–9 July 2015; pp. 271–276.
120. Souza, V.; Masip-Bruin, X.; Marín-Tordera, E.; Sánchez-López, S.; Garcia, J.; Ren, G.J.; Jukan, A.; Ferrer, A.J. Towards a proper service placement in combined Fog-to-Cloud (F2C) architectures. *Future Gener. Comput. Syst.* **2018**, *87*, 1–15. [[CrossRef](#)]
121. Ghobaei-Arani, M.; Khorsand, R.; Ramezani, M. An autonomous resource provisioning framework for massively multiplayer online games in cloud environment. *J. Netw. Comput. Appl.* **2019**, *142*, 76–97. [[CrossRef](#)]
122. Zeng, D.; Gu, L.; Guo, S.; Cheng, Z.; Yu, S. Joint optimization of task scheduling and image placement in fog computing supported software-defined embedded system. *IEEE Trans. Comput.* **2016**, *65*, 3702–3712. [[CrossRef](#)]
123. Rahbari, D.; Nickray, M. Scheduling of fog networks with optimized knapsack by symbiotic organisms search. In Proceedings of the 2017 21st Conference of Open Innovations Association (FRUCT), Helsinki, Finland, 6–10 November 2017; pp. 278–283.
124. Pham, X.Q.; Huh, E.N. Towards task scheduling in a cloud-fog computing system. In Proceedings of the 2016 18th Asia-Pacific Network Operations and Management Symposium (APNOMS), Kanazawa, Japan, 5–7 October 2016; pp. 1–4.
125. Kabirzadeh, S.; Rahbari, D.; Nickray, M. A hyper heuristic algorithm for scheduling of fog networks. In Proceedings of the 2017 21st Conference of Open Innovations Association (FRUCT), Helsinki, Finland, 6–10 November 2017; pp. 148–155.
126. Urgaonkar, R.; Ciftcioglu, E.N.; Yener, A.; Neely, M. Quality of Information aware scheduling in task processing networks. In Proceedings of the 2011 International Symposium of Modeling and Optimization of Mobile, Ad Hoc, and Wireless Networks, Princeton, NJ, USA, 9–13 May 2011; pp. 401–406.
127. De Benedetti, M.; Messina, F.; Pappalardo, G.; Santoro, C. JarvSis: A distributed scheduler for IoT applications. *Cluster Comput.* **2017**, *20*, 1775–1790. [[CrossRef](#)]
128. Hong, C.H.; Varghese, B. Resource management in fog/edge computing: A survey. *arXiv* **2018**, arXiv:1810.00305.
129. Neto, E.C.P.; Callou, G.; Aires, F. An algorithm to optimise the load distribution of fog environments. In Proceedings of the 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Banff, AB, Canada, 5–8 October 2017; pp. 1292–1297.



130. Ni, L.; Zhang, J.; Jiang, C.; Yan, C.; Yu, K. Resource allocation strategy in fog computing based on priced timed petri nets. *IEEE Internet Things J.* **2017**, *4*, 1216–1228. [[CrossRef](#)]
131. Aazam, M.; St-Hilaire, M.; Lung, C.H.; Lambadaris, I.; Huh, E.N. IoT resource estimation challenges and modeling in fog. In *Fog Computing in the Internet of Things*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 17–31.
132. Sood, S.K.; Singh, K.D. SNA based resource optimization in optical network using fog and cloud computing. *Opt. Switch. Netw.* **2019**, *33*, 114–121. [[CrossRef](#)]
133. Kapsalis, A.; Kasnesis, P.; Venieris, I.S.; Kaklamani, D.I.; Patrikakis, C.Z. A cooperative fog approach for effective workload balancing. *IEEE Cloud Comput.* **2017**, *4*, 36–45. [[CrossRef](#)]
134. Verma, S.; Yadav, A.K.; Motwani, D.; Raw, R.; Singh, H.K. An efficient data replication and load balancing technique for fog computing environment. In Proceedings of the 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 16–18 March 2016; pp. 2888–2895.
135. Gu, L.; Zeng, D.; Guo, S.; Barnawi, A.; Xiang, Y. Cost efficient resource management in fog computing supported medical cyber-physical system. *IEEE Trans. Emerg. Top. Comput.* **2015**, *5*, 108–119. [[CrossRef](#)]
136. Xu, X.; Fu, S.; Cai, Q.; Tian, W.; Liu, W.; Dou, W.; Sun, X.; Liu, A.X. Dynamic resource allocation for load balancing in fog environment. *Wirel. Commun. Mob. Comput.* **2018**, *2018*, 6421607. [[CrossRef](#)]
137. Zhang, H.; Xiao, Y.; Bu, S.; Niyato, D.; Yu, F.R.; Han, Z. Computing resource allocation in three-tier IoT fog networks: A joint optimization approach combining Stackelberg game and matching. *IEEE Internet Things J.* **2017**, *4*, 1204–1215. [[CrossRef](#)]
138. Alsaffar, A.A.; Pham, H.P.; Hong, C.S.; Huh, E.N.; Aazam, M. An architecture of IoT service delegation and resource allocation based on collaboration between fog and cloud computing. *Mob. Inf. Syst.* **2016**, 6123234. [[CrossRef](#)]
139. Zhang, Y.; Zhang, H.; Long, K.; Xie, X.; Leung, V.C. Resource allocation in software defined fog vehicular networks. In Proceedings of the 6th ACM Symposium on Development and Analysis of Intelligent Vehicular Networks and Applications, Miami, FL, USA, 21–25 November 2017; pp. 71–76.
140. Xu, J.; Palanisamy, B.; Ludwig, H.; Wang, Q. Zenith: Utility-aware resource allocation for edge computing. In Proceedings of the 2017 IEEE international conference on edge computing (EDGE), Honolulu, HI, USA, 25–30 June 2017; pp. 47–54.
141. Tselios, C.; Politis, I.; Amaxilatis, D.; Akrivopoulos, O.; Chatzigiannakis, I.; Panagiotakis, S.; Markakis, E.K. Melding Fog Computing and IoT for Deploying Secure, Response-Capable Healthcare Services in 5G and Beyond. *Sensors* **2022**, *22*, 3375. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.