



Review

# Fog Computing for Smart Cities' Big Data Management and Analytics: A Review

Elarbi Badidi <sup>1,\*</sup> , Zineb Mahrez <sup>2</sup> and Essaid Sabir <sup>2,3</sup>

<sup>1</sup> Department of Computer Science and Software Engineering, College of Information Technology, UAE University, AL-AIN P.O. Box. 15551, UAE

<sup>2</sup> NEST Research Group, LRI. Lab, ENSEM, Hassan II University of Casablanca, Casablanca 9167, Morocco; zineb.mahrez@ensem.ac.ma (Z.M.); e.sabir@ensem.ac.ma (E.S.)

<sup>3</sup> Computer Science Department, University of Quebec at Montreal (UQAM), Montreal, QC H2L 2C4, Canada

\* Correspondence: ebadidi@uaeu.ac.ae; Tel.: +971-3-713-5552

Received: 18 September 2020; Accepted: 29 October 2020; Published: 31 October 2020



**Abstract:** Demographic growth in urban areas means that modern cities face challenges in ensuring a steady supply of water and electricity, smart transport, livable space, better health services, and citizens' safety. Advances in sensing, communication, and digital technologies promise to mitigate these challenges. Hence, many smart cities have taken a new step in moving away from internal information technology (IT) infrastructure to utility-supplied IT delivered over the Internet. The benefit of this move is to manage the vast amounts of data generated by the various city systems, including water and electricity systems, the waste management system, transportation system, public space management systems, health and education systems, and many more. Furthermore, many smart city applications are time-sensitive and need to quickly analyze data to react promptly to the various events occurring in a city. The new and emerging paradigms of edge and fog computing promise to address big data storage and analysis in the field of smart cities. Here, we review existing service delivery models in smart cities and present our perspective on adopting these two emerging paradigms. We specifically describe the design of a fog-based data pipeline to address the issues of latency and network bandwidth required by time-sensitive smart city applications.

**Keywords:** fog computing; cloud computing; smart cities; smart grids; smart transportation; smart healthcare; big data; data storage; data analytics; data pipeline; data processing

## 1. Introduction

Many smart city facilities use advanced Internet of Things (IoT) solutions to implement smart services and applications that use real-time data from various devices such as sensors and meters. Examples of exciting applications include smart transportation services, smart parking solutions, smart waste management solutions, smart grid applications, and smart healthcare solutions. IoT initiatives include ways to reduce waiting time in public transportation facilities/stops and in emergency rooms, track various city assets, and track people who use public transportation to attend different city events. These initiatives also include ways to provide proactive alerts about devices that may fail and prevent threats and unauthorized entry and exit from city facilities using surveillance cameras and electronic identification-enabled security doors. These advances are undoubtedly impressive, but managing the massive amounts of data they generate in near real-time is a challenge. The potential benefits of IoT solutions in smart cities face challenges in storing, distributing, and processing these immense IoT data. Many smart city applications use the cloud to store and process data to take advantage of its elastic computing power and unlimited storage capacities [1–3]. However, many other applications

are very time sensitive and cannot support data stream transmission to cloud servers for processing due to excessively high latencies and high network bandwidth usage requirements.

In transportation applications, large amounts of data are generated by vehicles on the road as well as by roadside units and many other devices. In the healthcare sector, devices worn by patients, medical equipment and the various sensors deployed in healthcare facilities generate enormous amounts of data. The rapid processing of data in these applications would help improve the transport system, improve patient care by optimizing care processes, and reduce the overall cost per patient. Many smart city applications are already using the power of cloud computing for their data processing needs. Time-sensitive or data-intensive applications would have a problem using cloud computing as the backend servers could be far away from edge devices. The data themselves could clutter the network. For example, applications such as electrocardiogram (ECG) signal processing could have real-time requirements, while computer-aided tomography processing could have a big data transportation problem. Healthcare facilities and providers need to stream and process data at the edge in real time. Instead of conveying data to cloud servers for processing and storage, sensors and edge devices should transmit their data to edge gateways or servers to be aggregated, processed, or analyzed. This operation would allow a reduction in the volume of data to send and store in cloud servers, minimizing costs, reducing latency, and controlling the network bandwidth usage [4].

We review relevant research efforts on adopting these two emerging distributed computing technologies in smart cities in this work. More specifically, we review fog node deployment models that the National Institute of Standards and Technology (NIST) prescribes and then discuss some fog-based service delivery models for smart cities. Afterwards, we describe some use cases of fog computing in smart cities, mainly intelligent transportation, smart healthcare, and smart grids. Then, we present a conceptual design of a fog computing-based data pipeline for smart cities and discuss the challenges of fog computing-based solutions in smart cities.

This paper is organized as follows. Section 2 describes the emerging paradigms of edge and fog computing and the benefits they offer to users. Section 3 outlines the fog-based service delivery models, which NIST prescribes. Section 4 presents our view regarding how data as a service (DaaS) could be an additional edge and fog service delivery model. Section 5 describes the possible deployment models of fog nodes in a smart city, the potential techniques to transfer large volumes of data from the IoT infrastructure to fog nodes and cloud servers, and the various data management operations in the edge–cloud continuum. Transportation, healthcare, and smart grids are three application domains of smart cities that have witnessed growing interest over the past few years in adopting fog computing to cope with many of their challenges. Section 6 reviews several research efforts and implementations, which assess the benefits of using fog computing-based solutions in these application domains. Section 7 describes our proposed conceptual framework in which several recent technologies for data management and processing could be used to build a fog-based big data pipeline for smart cities. Section 8 discusses the challenges that smart cities must overcome to take advantage of edge and fog computing. Finally, Section 9 concludes the paper and highlights the main open research issues that need to be addressed.

## 2. Background

### 2.1. Edge and Fog Computing

Gartner defined Edge computing in their glossary as follows: “Edge computing is part of a distributed computing topology where information processing is located close to the edge, where things and people produce or consume that information” [5]. Instead of relying on a data center that can be thousands of miles away for storage and processing, edge computing brings data storage and processing closer to the devices that generate and collect the data. Proximity is essential for applications that require real-time data for their operations and very low latency. Additionally, by processing data locally, organizations can save money by reducing the data volume to be transferred and processed in

cloud servers [6]. Companies that have adopted cloud-based solutions to implement many business processes may have realized that the required bandwidth results in higher costs than expected. A typical example is a video camera connected to the Internet that sends live photographs from a public square in the city or an app that monitors health equipment in a health facility using several sensors and IoT devices. A single video camera can very quickly transmit the generated data over a network. However, when many video cameras simultaneously transmit live recordings, many problems can arise. Not only does latency affect quality, but bandwidth costs can be huge.

Edge computing is proliferating due to the rapid and growing deployment of IoT devices, which generate vast amounts of data during their operations. These devices connect to the Internet directly or through gateways to transmit data to the cloud or receive instructions from the cloud server. According to IDC’s forecast for 2019, “at least 40% of the data generated by IoT devices would be stored, processed, and analyzed at the edge of the network.” [7]. The inefficiency of cloud-based data processing solutions for latency-sensitive edge applications has led to the design of some solutions deployed at the network’s edge. These solutions include micro data centers, cloudlets and fog computing-based solutions [6,8].

Cisco Systems defined Fog computing as follows: “Fog Computing is a highly virtualized platform that provides compute, storage, and networking services between end devices and traditional Cloud Computing Data Centers, typically, but not exclusively located at the edge of the network” [9]. The primary benefit of both edge computing and fog computing paradigms is the ability to store and process data faster, making real-time applications critical to business operations more efficient. Before the advent of edge and fog computing, an application using a surveillance camera had to invoke a cloud-based service to perform facial recognition, which results in high latency. With an edge or fog computing-based solution, an edge server or gateway would run the service locally. Applications such as self-driving cars, virtual and augmented reality, and intelligent transportation systems require rapid processing and response. Figure 1 illustrates the position of fog computing in a fog–cloud smart city environment and examples of application areas that would benefit from it.

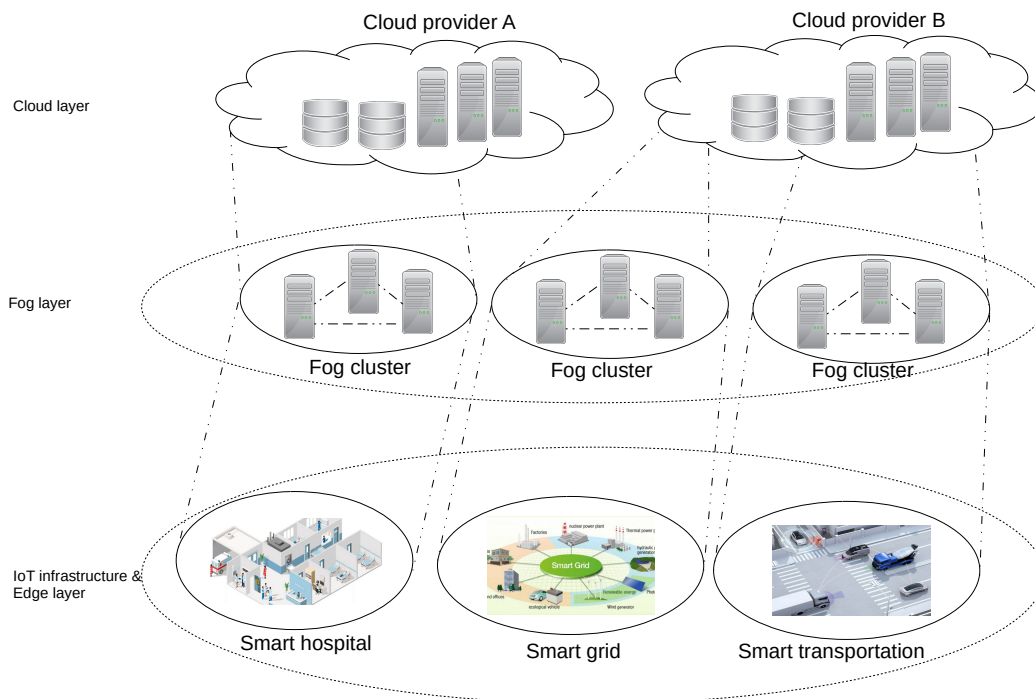


Figure 1. Smart city fog and cloud infrastructure.

The earlier literature on fog computing and edge computing described the two paradigms interchangeably. For both technologies, IoT data processing is done close to the data sources before being transferred to a cloud server. With edge computing, data are typically processed in edge devices and then forwarded to fog nodes or cloud servers using edge gateways, which have more powerful communication capabilities than edge devices. With fog computing, the data are analyzed and processed in a fog node at the local network's edge.

The OpenFog Consortium ([openfogconsortium.org](http://openfogconsortium.org)), which is now part of the Industrial Internet Consortium (IIC), reveals that edge computing is often mistakenly called fog computing. Its reference architecture for fog computing, released in February 2017, affirms that: (i) fog nodes are organized hierarchically, with a three-tier architecture, although n-tiers can be used for specific scenarios, and (ii) fog computing provides computation, networking, storage, and control anywhere from cloud to IoT and sensing devices, while edge computing provides limited computing resources at the edge [10]. Its first tier, near the edge of the network, typically involves getting data from edge devices, normalizing data, and controlling sensors and actuators. The second tier involves the filtering, compression, and transformation of data. The third level is where the aggregation of data and the transformation of those data into knowledge occur.

Fog nodes represent the backbone of the fog computing infrastructure. They can be physical devices such as gateways, switches, routers, servers, or software components like virtual machines and cloudlets connected with edge devices and access networks to provide them with data storage and computing resources. For example, Cisco network devices such as routers, switches, and Unified Computing System (UCS) servers could serve as fog nodes. These servers include computing hardware, virtualization support, switching fabric, and management software. Organizations can develop and enhance their IoT applications in the cloud to finally deploy them in fog nodes or cloud servers [11]. The fog computing layer lies between the edge layer, where the end devices reside, and the cloud computing layer, where unlimited computing power and storage are available. Fog nodes can operate in a stand-alone mode or can be configured to work in a cluster to provide the service to their users [12]. "Fog nodes... can be deployed anywhere with a network connection: on a factory floor, on top of a power pole, alongside a railway track, in a vehicle or on an oil rig", according to Cisco. A fog node implementation is described in [13] and a fog architectural design and implementation are discussed in [14]. Eva Marín Tordera et al. [15] focused on identifying the basic functionality of a fog node and the associated opportunities and challenges. They summarized and compared the concepts and the lessons learned from the implementation of advanced fog computing technologies. Moreover, they showed how a unified definition of fog nodes could emerge from a conceptual framework. Additionally, they discussed some of the open issues and challenges that arise when fog nodes expose abstract and virtualized views of their physical resources to the upper layers in a fog–cloud scenario.

There is almost unanimity in the literature that fog computing is not meant to replace cloud computing, but rather it is an extension. Many cloud computing technologies, such as virtualization, also apply to fog computing [16]. Furthermore, data processing can be done on a single fog node or multiple nodes working together. When more processing power is needed, additional fog nodes might be deployed, improving the scalability and providing the elasticity required by multiple enterprise applications.

## 2.2. Characteristics of Fog Computing

The research efforts described in Section 6 as well as other works in other fields demonstrate that reduced latency, reduced traffic and bandwidth optimization, and enhanced privacy and security represent the essential advantages that edge and fog computing have over cloud computing.

**Reduced latency:** With cloud-based IoT solutions, whenever an application initiates an action, for example, to query the status of a medical device over a period of time, the system must understand the command, send a request to a cloud server for processing, then wait for the response before presenting it graphically. With solutions based on edge computing, IoT data streams are partially



stored and processed locally, allowing businesses to get faster responses and reduce network traffic. Reduced latency is essential for many verticals, such as supply chain management and logistics, smart transportation, and digital healthcare.

**Reduced traffic and bandwidth optimization:** According to several forecasts, the number of IoT devices will exceed 30 billion over the next few years. These devices would generate nearly 2.5 billion bytes of data per day. To reduce the volume of data crossing the network and optimize bandwidth usage, organizations need to analyze IoT data at the edge, while state data can be aggregated, compiled, and sent daily to a central cloud repository. Data analysis at the edge can be supported by machine learning and artificial intelligence algorithms. Many modern devices, such as consumer electronics, delivery drones with computer vision capabilities, and autonomous industrial robots, already support artificial intelligence algorithms.

**Enhanced privacy and security:** Edge and fog computing allow for the analysis of sensitive data at an on-premises gateway or fog node instead of a data center outside of the organization's control, which dramatically improves data privacy. Vulnerabilities in the Internet of Things come from the limited computing power of IoT devices, as well as hardware design flaws and firmware update constraints. These inconsistencies can provide hackers with an easy entry point into a network. Data transfer between devices in an IoT solution and the cloud infrastructure can also pose security issues. Therefore, the security of IoT solutions should be enhanced at the network level. The network edge might be the place to insert security patches between vulnerable devices and the other components of the network.

For many organizations, data overload and the high cost of cloud analytics is the biggest barrier to capturing and analyzing sensor data. The above benefits of edge and fog computing can lead many companies to change their minds by adopting smart IoT solutions in which edge and fog computing could help combat the data deluge by making it easier to collect, aggregate, and process their IoT data. A balance between processing and storing data at the edge/fog level and its processing and storage at the cloud level is necessary because the cloud will always have more storage and processing power than edge devices. Fog computing is a substitute for pure edge computing in cases where edge devices cannot perform data analysis due to their limited resources.

### 2.3. What Value Does 5G Bring to Fog Computing?

Fifth-generation technology promises to elevate the cellular network and not only to interconnect people, but also to interconnect and control machines, things, and devices. The main characteristics of 5G networks are high speed data transfer, zero latency, and ubiquitous connectivity, which should support a wide range of high-performance applications and services [17]. Fifth-generation technology will provide new levels of performance and efficiency that will allow new user experiences and new industries to connect. Moreover, 5G will offer multi-Gbps data rates, ultra-low latency, huge capacity, and a more consistent user experience. For instance, Yu et al. [18] used fog computing to support the 5G-enabled Internet of Vehicles (IoV) and thus provide accurate location service and address the problems associated with the failure of the Global Navigation Satellite System (GNSS). They proposed a topology that relies on the GNSS-free emergency location service in the 5G-enabled IoV, which contains fog clusters and fog nodes to collect traffic data and process requests rapidly. The combination of fog computing and 5G-enabled IoV helped to achieve high precision in estimating the vehicles' location. The significant value that 5G brings to fog computing is the large number of simultaneous devices supported by 5G cells compared to 4G. The utilization of 5G in smart city scenarios means that individual components of the smart city can be connected autonomously, sharing data, aggregating it, and exploiting it in real time.

### 3. Service Delivery Models in Fog Computing

Fog computing in smart cities involves a wide variety of resources ranging from computation, data storage, networks, data acquisition to data analysis. NIST has defined three types of service

models that fog computing solutions could implement. These models are like the delivery models prescribed for cloud computing-based solutions [12,19].

### 3.1. Infrastructure as a Service (IaaS)

In the infrastructure as a service (IaaS) delivery model, computing infrastructure is delivered on an outsourced basis to a fog service consumer to support its operations by leveraging the infrastructure of the fog nodes, forming a federated cluster. Typically, a fog computing provider hosts the infrastructure components, including servers, storage, networking hardware, and the virtualization or hypervisor layer. Like the cloud IaaS services, the fog IaaS services relieve the user from managing and controlling the fog nodes' resources, but they do allow the user to control the operating systems, the amount of storage to use and the applications to deploy. As IaaS applications often require different resources, customization and flexibility are essential for IaaS users. The maturity of virtualization technologies makes it possible for applications to run in virtual machines (VMs) isolated from the underlying infrastructure. VMs provide the flexibility to meet personalized user needs.

### 3.2. Platform as a Service (PaaS)

The platform as a service (PaaS) delivery model in fog computing is similar to the PaaS delivery model in cloud computing. It makes it easy to deploy customer created or acquired applications to federated fog nodes that form a cluster without incurring the cost and complexity of purchasing and managing the underlying fog infrastructure, which includes network, servers, operating systems, and storage hardware. The PaaS service provider provides all the facilities, such as programming languages, libraries, and tools, necessary to support the full lifecycle of building and delivering applications and services. The PaaS customer has control over the deployed applications and configuration settings of the hosting environment. The PaaS solution of Hivecell [20] is an example of PaaS for edge and fog computing, allowing users to install rich distributed frameworks on a cluster of Hivecell devices very quickly with the click of a button. Hivecell's solution provides support for machine learning at the edge. It supports several distributed frameworks and machine learning solutions such as Kafka [21,22], Kubernetes [23], and Tensorflow [24]. Each Hivecell device has a 256-core Compute Unified Device Architecture (CUDA) Graphical Processing Unit (GPU), which provides the necessary resources for training and testing models where needed.

### 3.3. Software as a Service (SaaS)

This type of service is similar to software as a service (SaaS) in cloud computing and means that the consumer of the service accesses the applications offered by the fog node through a thin client or simple interface. The fog provider applications run on a group of federated nodes managed by the fog provider. The consumer of SaaS fog services is exempt from the management and control of the underlying fog infrastructure, including the network, servers, operating systems, storage, or even individual application functionality, except configuring user-specific application settings.

Smart cities use various software tools in their operations and for data analysis. The SaaS delivery model is becoming attractive for smart cities as it provides them with online software services and various software tools deployed at the fog level and facilitates remote access to them over a network or the Internet. It does not require installing fog-based services locally and makes it easier to access the latest version of the services.

## 4. Fog Data as a Service Delivery Model

Because of the importance of data analysis in fog computing, we believe that data as a service (DaaS) needs to be considered as the fourth delivery model in fog-based environments. With the advent of the service-oriented architecture (SOA) and the proliferation of cloud-based data management and access solutions, data as a service (DaaS) has emerged as a data delivery model that continues to grow in popularity. DaaS relies on the same concepts as the other cloud delivery models (IaaS,

PaaS, and SaaS). With the DaaS model, data are delivered to consumers on-demand regardless of their location [19,25]. DaaS considers that the platform where the data resides is not essential to the consumer. The consumer application could access the data where they live. Figure 2 shows a DaaS-based system's components, including service-oriented architecture as the underlying platform, data ingestion and processing, microservices, applications, and visualization dashboards. The usage of this delivery model in fog computing is growing. For example, Plebani et al. [26] proposed a DaaS-based solution to support data delivery in a fog computing environment. The solution permits efficient data transfer between data stores owned by data providers and data consumers.

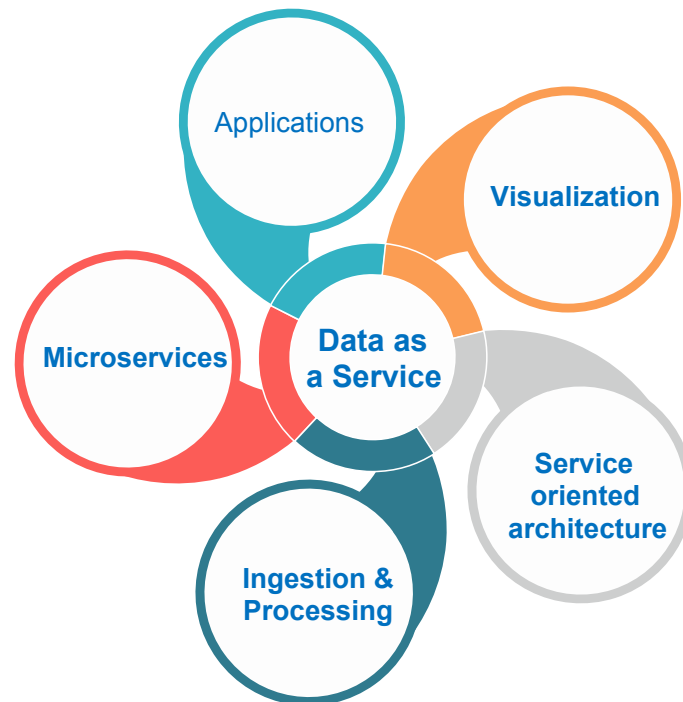


Figure 2. Data as a service.

DaaS provides the following benefits:

- **Agility:** Since DaaS relies on a Service-Oriented Architecture (SOA), access to critical data through a cloud or fog service powered by DaaS provides great flexibility. Access to data is fast, because the architecture in which they exist is quite simplistic. Moreover, when the data structure needs to be changed or geographic needs arise, changes to the data are easy to implement.
- **High-quality data:** Implementing a rigorous process of data management and processing (acquisition, cleansing, aggregation, and enrichment) by the DaaS provider guarantees consumer access to high-quality data.
- **Easy access:** The DaaS model permits easy access to data using various devices like desktops, laptops, tablets, and smartphones anywhere and anytime.
- **DaaS provider lock-in avoidance:** The DaaS model allows data to be transferred quickly from one platform to another.

In smart cities, data are fundamentally important for downstream analysis and decision-making. With the phenomenal growth of smart city data, providing access to data as a service (DaaS) over the Internet or through a network is of utmost importance. The DaaS service enables the consumer to have dynamic access to data on demand and provides up-to-date data that could be used by a wide range of smart city applications. Similar to other cloud and fog delivery models, DaaS adoption raises several concerns with regard to privacy, security, data governance, and ownership issues.

## 5. Toward Fog-Based Smart City Data Management and Analytics

Although relatively new, fog computing holds great promise for effectively solving data analysis and management problems in smart cities, using one of the following fog node deployment models, as prescribed by NIST [12].

- Private fog cluster (or node): A private fog cluster is typically designed to be used exclusively by a single organization with multiple consumers. The organization could own the cluster, operate it, and manage it or delegate its management fully (or partially) to a commercial third party.
- Community fog cluster (or node): A community fog cluster is generally designed for exclusive use and exploitation by a specific community of consumers belonging to many organizations that share common concerns. One or more community organizations can own, operate, and manage it or delegate its management to a third party.
- Public fog cluster (or node): A public fog cluster is designed to be open to the general public. A business, government agency, university, or a combination of these three entities can own, operate, and manage the cluster.
- Hybrid fog cluster (or node): A hybrid fog cluster consists of at least two different fog nodes (private, community, or public), which operate independently. The portability of data and applications between nodes (e.g., fog bursting for load balancing between fog nodes) is ensured using proprietary or standardized technology.

### 5.1. Deploying Data and Software in Fog Nodes and Cloud Servers

The traditional method for smart city applications is often to install software tools locally and perform in-house analysis. However, with recent advances in cloud and fog computing, the growing trend is to store data and deploy software in fog/cloud to make them available as a service (i.e., DaaS for data and SaaS for software). Depending on the kind of data analysis required by the application, data can be deployed in the fog or the cloud. Time-sensitive analysis, which requires low latency, would use data stored in the fog. However, non-time-sensitive analysis would use data deployed in the cloud to gain in-depth insights.

Currently, most smart cities store only a small amount of their data in the cloud. The bulk of their data is still in traditional databases. As an increasing number of projects, such as healthcare and transport, deploy IoT solutions to enhance their services, they generate massive data volumes. Data storage, sharing, and analysis would be required at the fog/cloud level. Transferring massive amounts of data from smart cities to cloud servers is a significant bottleneck. At present, it is not uncommon to physically ship large-capacity hard drives, such as Amazon Web Services (AWS) Snowball devices, to the cloud data center. It is recognized that when data to be moved to the cloud are on a terabyte scale and beyond, it is better to ship them to the cloud provider rather than uploading them. Direct uploading of large volumes of data to the cloud may require an unacceptable amount of time, even on a hundred Mbps Internet connections and faster. The technique to choose would depend on many factors, including the size of the data to be transferred, the speed of Internet connection supported between the source and destination servers, the sustained speeds of copy-in/copy-out supported by the storage device and the source and destination drives, the monetary cost of data transfer and, to a lesser extent, the cost of shipping and transit time. Currently, a promising technique for transferring large volumes of data to fog/cloud servers is using innovative new transfer technologies such as IBM Aspera's high-speed file transfer technology (Aspera FASP). This technology significantly speeds up file transfers and outperforms traditional file transfer technologies such as File Transfer Protocol (FTP) and Hypertext Transfer Protocol (HTTP). Several large companies rely on this technology for the high-speed transfer of large volumes of data. For example, BT Sport uses Aspera technology to power its intense file-based production workflows. Furthermore, other technologies, such as data compression, are available to facilitate the transfer of large amounts of data [27,28].

Answering the smart city’s most important and most complex questions often involves the use of multiple tools. Most of the software tools used by different city departments are designed for workstations (rather than the cloud) and therefore are not delivered as cloud-based web services, making it impossible to run complex analysis tasks. As a result, many smart cities’ continued efforts have only used a small portion of the available cloud-based tools. To store, share and analyze big data with lower cost and increased efficiency in smart cities, it is essential to make a large volume of smart city data available to the public, as well as a wide variety of software applications in the fog/cloud, provided as services via the internet.

### 5.2. Fog-Based Data Management and Analytics

As smart cities embark on new IoT initiatives and strive to extract more information from continuously generated data in large quantities to improve or deliver new smart services and respond to emergencies, new approaches and data management techniques are needed. Traditional databases and business intelligence architectures will always be crucial for smart cities, but IoT solutions require specific capabilities to manage various data continuously from various sources in different formats.

Data management in the IoT era is becoming a broad discipline, which entails techniques, tools and platforms for storage, preprocessing, batch and stream processing. It encompasses other data-related disciplines, including data integration (propagation, consolidation, and federation), data quality management, data provisioning, and data governance. Therefore, managing IoT data in organizations, such as smart cities, is a complex process, especially when business intelligence and decision making must use large amounts of data from multiple heterogeneous sources. These organizations deploy algorithms to filter data from multiple sources, providing different data quality levels before they reach a centralized data store. Additionally, they use automated data aggregation and classification tools at the edge and in the fog to accelerate the generation of insights from data streams and protect data stores against massive data volumes and high data velocity. Figure 3 depicts the aforementioned data management functions in the continuum from IoT devices to the fog/cloud. These data management operations are detailed in Section 7, which describes a conceptual framework and the necessary tools for building a fog-based data pipeline.

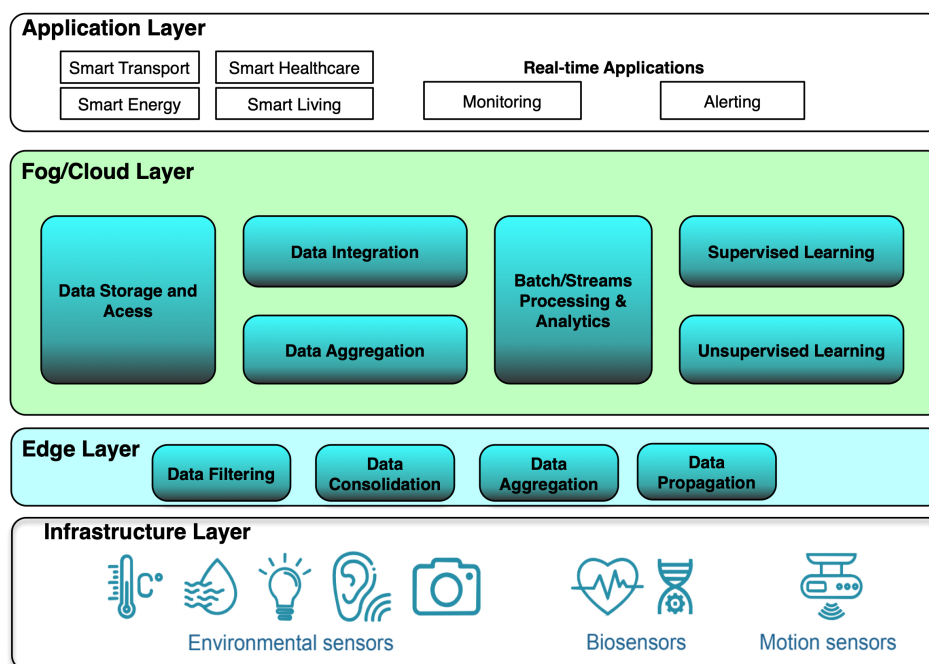


Figure 3. Data management operations at the edge–cloud continuum.



Several recent research efforts studied the issue of big data management and analytics in smart cities [1,29–31]. Our focus in this work is on the efforts that use fog-based solutions in the continuum from edge to cloud. Here, we review some of these works and the next section describes some works in three application domains: transportation, healthcare, and smart grids. Tang et al. [32] proposed a hierarchical fog-based architecture for data management in smart cities. They described the data management tasks performed by edge devices (edge gateways) and fog nodes. Each edge device must identify potential threat patterns from sensor data streams and send control signals to sensors when a threat is detected. The authors state that various supervised machine learning algorithms could be used to identify threat patterns and that unsupervised machine learning algorithms could be used to detect data anomalies. Additionally, edge devices could perform feature extraction and report the results to fog nodes for further analysis. Fog nodes typically receive data from multiple edge devices and combine their features into a single vector. In their pipeline monitoring use case, they apply the Hidden Markov Model (HMM) to model each event's spatiotemporal association in a probabilistic manner. For smart homes management, Yassine et al. [33] proposed a fog/cloud-based platform to implement IoT big data analytics. One of the main components of their platform is an "IoT Management Broker" which is in charge of handling various requests from multiple smart homes. Data management operations in fog nodes include: data preprocessing, pattern mining, classification, prediction, and visualization. The authors in [34] discussed how some data management issues in the medical field are addressed using fog computing. Data fusion from large heterogeneous datasets includes data combination, data integration, and data aggregation. They also discussed the issue of data migration from legacy systems into new databases.

Car parking occupancy in cities is one the concerns that all cities strive to solve. The authors in [35] proposed an IoT-based solution to monitor car park occupancy in a city by carrying out data analytics at the fog level. Edge gateways perform data aggregation and preprocessing. The proposed solution uses Hadoop and MapReduce [36,37] for data processing and runs on a cluster of commodity computers at each fog node. This work is limited to batch processing as data are stored in log files.

The authors in [38] proposed a big data architecture that distributes data processing at the three layers—edge, fog, and cloud—while taking advantage of each layer's capabilities. The edge layer is responsible for managing devices and actuators; the fog layer ensures data ingestion and performs data aggregation; the cloud is responsible for carrying out massive analytics that require more computational resources. The architecture's components are comparable to the ones we suggested in Section 7 that describes our proposed fog-based data management pipeline. These components include tools for data ingestion, data stream processing and analytics, and data visualization. The next section describes three use cases of fog computing-based solutions proposed to cope with many smart cities' challenging issues—in particular data management. The use cases are on (1) smart transportation and vehicular fog computing, (2) smart healthcare, and (3) smart grids.

## 6. Fog Computing and Data Management Use Cases in Smart Cities

### 6.1. Intelligent Transportation Systems and Vehicular Fog Computing

Transport systems in cities worldwide are adopting digital technologies to deal with the challenges of this vital domain in any city. Intelligent Transportation Systems (ITS) are part of this development and are also part of a broader vehicle automation process. These systems use emerging technologies that allow road vehicles to communicate with other vehicles (V2V) or road users and roadside infrastructure. ITS systems can improve traffic efficiency and road safety in cities and reduce energy consumption and transport emissions. These goals cannot be achieved without improving the quality and reliability of information and ensuring data protection and cybersecurity. Figure 4 illustrates a fog-based intelligent transportation system (ITS). Fog computing provides several benefits to all users of the intelligent transportation system. The vehicles are informed of traffic status and congested roads and are redirected to other routes. They are informed of road accidents and emergency evacuation

routes, as well as services such as finding appropriate parking spaces. Traffic lights and traffic signs can be adjusted according to traffic conditions. They can recognize emergency vehicles such as ambulances and police cars and assign them a specific lane. Pedestrians are informed of the best routes to their destinations based on the traffic conditions on the roads and the condition of the sidewalks they have to cross. The Internet of Vehicles (IoV) refers to a network of these various entities, which include vehicles, roadside units, pedestrians, traffic lights, and parking lots. This network provides real-time communication between these entities. To solve the parking problem, Tang et al. [39] proposed a parking service that relies on fog computing to provide drivers looking for parking spaces with real-time information on vacant parking spaces. The service aims to improve the forecast of available parking spaces. Data on the number of vehicles looking for a parking space and the number of currently vacant parking slots are collected by the fog nodes, making predictions and broadcasting the information to the vehicles. Fog computing can also improve transit services by providing up-to-date information on the arrival and departure of public transit services such as buses and trams [40].

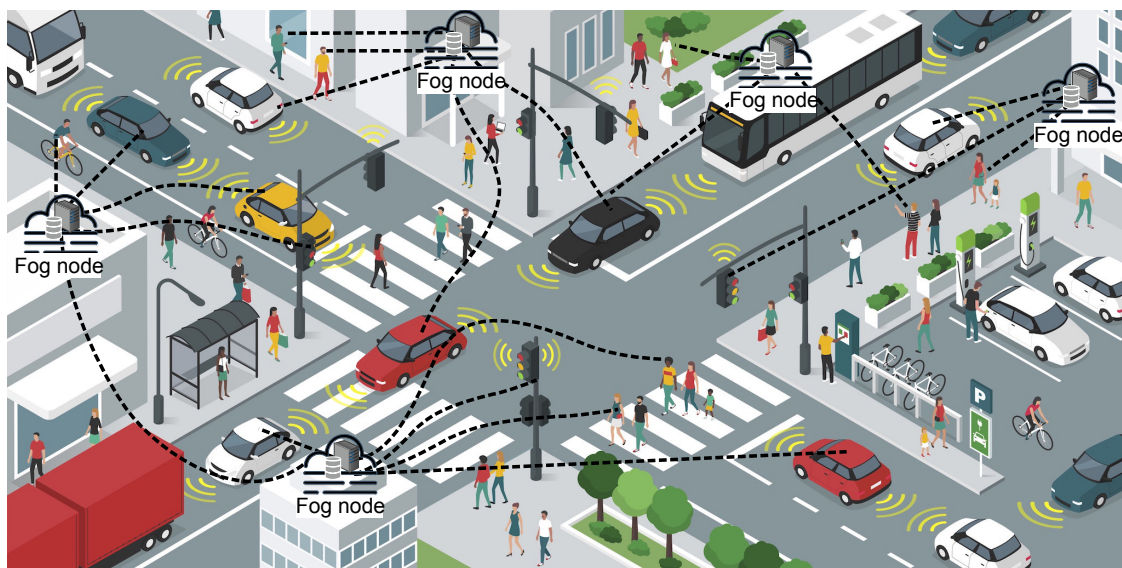


Figure 4. Fog computing in Intelligent Transportation System.

In the context of IoV, Chun et al. [41] proposed an architecture that relies on fog computing to share semantic IoV knowledge among fog nodes, using a publish/subscribe model and their proposed ontology, consistently and explicitly. Fog nodes continually collect data from vehicles and smart traffic lights. A publisher fog node is responsible for defining an event (topic) to which other fog nodes can subscribe. The distribution of fog nodes at different localities, such as at a road intersection, helps meet the demand for low-latency service.

Urban vehicle networks are an essential element in future intelligent transport systems. They provide support for various mobile services, including information delivery services and content sharing services. However, without effective communication and IT support, these services cannot be put into practice in everyday life and remain in the design phase. Hou et al. [42] proposed a new system, which expands the available resources and increases the achievable capacities using vehicles as a communication and computing platform. They referred to this system as Vehicle Fog Computing (VFC). They described four application scenarios in which moving vehicles and parked vehicles are used as infrastructure. They presented an overview of the VFC paradigm's potential capabilities, which uses vehicles as infrastructure, and the open problems associated with it. Their study demonstrated the benefits of VFC in increasing computational speeds and reducing delays for applications with intensive computing requirements. In another study, Ning et al.'s proposed architecture uses vehicular fog computing for distributed traffic management in real time [43]. At the center of each region of the city,

a cloudlet component manages the messages uploaded by vehicles. Data received from the vehicles are processed at the cloudlet layer, then delivered to the cloud layer. Vehicles and roadside unit (RSU) devices in the wireless communication range of vehicles form the fog layer. Moving vehicles and parked vehicles near RSUs form fog nodes. Data sensed by vehicles can be uploaded to RSUs. By balancing the load between the cloudlet and fog nodes, the architecture objective is to minimize the response time of the events collected and reported by vehicles.

Moreover, Xiao et al. [44] proposed a VFC model, which involves deploying fog nodes on some connected vehicles, such as taxis and self-driving buses. These mobile vehicular fog nodes provide computation and communication capabilities to applications that require these resources on the go. Static fog nodes at the edge of the network and mobile vehicular fog nodes provide on-demand fog computing capabilities. Vehicular fog nodes locally process data (such as real-time video) sensed by vehicles, and typically host applications with stringent latency and privacy requirements. If they are overloaded or vehicles are not within the V2V communication range, their workload may be offloaded to nearby cellular or vehicular fog nodes. Du et al. [45] highlighted the importance of IoV in an autonomous platoon scenario, where sensing data are highly time sensitive and massively shared. They proposed a new VFC architecture for autonomous driving vehicles that cooperate using their sensing capabilities. The architecture simulates platoon vehicles as a vehicular fog where the lead vehicle, with powerful sensing abilities, is the server, and the following vehicles are the fog nodes. The authors stated that their simulation results showed that their proposed VFC architecture helped achieve a 90% sensing coverage ratio in the platoon scenario.

In the context of electric vehicles, Belakaria et al. [46] addressed the computational and charging problems that hinder the successful implementation of an “autonomous electric mobility on demand” service (AEMoD). Their solution used a fog-based architecture to improve local management operations, minimizing the expected response time (elapsed time from customer request to vehicle dispatching) and ensuring an efficient charging strategy (finding a nearby charging station). Moreover, Sun et al. [47] used fog computing to develop an energy trading system for Plug-in Hybrid Electric Vehicles (PHEVs). Their solution promotes the balancing of the energy market based on a local V2V energy trading architecture to reduce time costs and energy losses. Darwish et al. [48] described the IoV environment, big data characteristics in ITS, and some real-time big data analytics solutions. Furthermore, they discussed the opportunities and challenges in the IoV environment concerning the implementation of fog-based solutions and real-time big data analysis. In addition, the authors in [49] proposed a fog computing system that combines the computing capacity of congested local vehicles and remote data centers to better meet the demand for vehicles in terms of computational capabilities. Using an immigration–death model, they predicted a vehicular fog’s potential computability and observed the relationship between computational capacity and vehicular fog radius. In addition, they studied the statistical characteristics of computing capacity generated by vehicular fogs across the city and observed their spatial distribution using visualization. Table 1 summarizes these aforementioned fog-based applications in transportation.

## 6.2. Fog Computing in Smart Healthcare

With the advent of cloud computing, many researchers investigated the use of cloud computing to support the computational needs, data management, storage, and integration of healthcare systems. Minh Dang et al. [50] studied how cloud computing, big data, ambient assisted living, and wearable technologies could contribute to cloud-based health solutions’ sustainable development. Moreover, they examined the issues of privacy and security in IoT, including potential threats, types of attacks, and security configurations from a healthcare perspective, and highlighted the opportunities and challenges for IoT-based healthcare solutions. Goli-Malekabadi et al. [2] studied the storage of healthcare big data in the cloud and its retrieval. Their proposed data storage model relies on Document-based Non-Structured Query Language (NoSQL) databases. They assessed their model’s effectiveness against the relational database model using some metrics such as data

preparation, query time, flexibility, and extensibility. Their results revealed that the proposed model showed similar performance as SQL Server for “read” queries, and it outperformed it for “write” queries in terms of data preparation, flexibility, and extensibility. M. Elhoseny et al. [51] proposed a model for health service applications based on IoT cloud. The model aims to optimize the selection of cloud virtual machines for use in data analysis. It implements three algorithms using Particle Swarm Optimization (PSO), Parallel Particle Swarm Optimization (PPSO), and Genetic Algorithm (GA) optimizers. The experiments they conducted with these optimizers revealed that the system efficiency, in terms of real-time data retrieval, improved significantly by a factor of 5.2%. Moreover, several authors surveyed the existing research works on cloud-based healthcare applications. Calabrese et al. [52] reviewed a number of cloud-based applications in healthcare biomedicine and bioinformatics. They highlighted the issues associated with using such applications for storing and analyzing patient data with regard to privacy, security, confidentiality, and data integrity. Kundella et al. [53] surveyed several works on cloud computing-based big data analytics for healthcare. They investigated the different techniques and algorithms used in these works with regard to privacy and security, as well as the different metrics used in big data analytics.

**Table 1.** Fog computing based transportation applications.

Application	Computing Model (Fog/Cloud)	Fog Nodes’ Role	Benefits
Parking service [39]	Fog	Collect data on the number of vehicles looking for a parking space	Providing drivers with real-time information
		Collect data on available parking slots	Enhancing the prediction of vacant parking lots
Transit services [40]	Fog	Provide up-to-date information on the arrival and departure of public transit services such as buses and trams.	
Sharing of IoV semantic knowledge [41]	Fog with Publish/subscribe	Collect data from vehicles and smart traffic lights	Providing low-latency service
Vehicles as infrastructure for computation and communication (VFC) [42–45] Autonomous platooning vehicles [45]	Fog/cloud 3-layer architecture [43]	Moving and parked vehicles act as fog nodes	Increasing computing speeds
Autonomous platooning vehicles [45]		Sense road events and upload data to RSUs	Decreasing delays for applications with intensive computations needs
		Support local processing of sensed data	Minimizing response time
Autonomous driving vehicles		Host applications with privacy requirements and strict latency Real-time processing of traffic videos	
Autonomous electric mobility on demand [46]	Fog	Improve local management operations	Minimizing response time
			Ensuring an efficient charging strategy



Most surveys suggest that cloud computing may not be suitable for time-sensitive healthcare applications such as remote patient monitoring applications, which require immediate actions from physicians for unexpected situations. Transferring data to cloud servers for analysis and processing and waiting for the results could not be tolerated in such circumstances. With the growing data management capabilities of fog nodes, advanced analytics, including event processing and machine learning, could be performed at the edge, improving network conditions and end-to-end latency. Furthermore, by using fog services, health systems could have a dashboard providing real-time information on medical equipment and patient conditions. Thus, remedial actions could be taken before accidents or adverse conditions occur.

The potential benefits of edge and fog computing and the advancement made in artificial intelligence and data analytics spurred many efforts to harness IoT, fog computing, cloud computing, and data analytics in healthcare systems. Kraemer et al. [54] studied several health applications, which they categorized by use case class and deployment scenario. Based on this categorization, they concluded that several applications could benefit from the deployment of fog computing solutions in health facilities. The majority of ubiquitous healthcare applications usually need to be running somewhere between the physical infrastructure and the cloud. The authors provided an inventory of application tasks with a description of where they could run in a fog/cloud architecture. Aazam et al. [55] proposed Emergency Help Alert Mobile Cloud (E-HAMC), an alert and emergency management architecture that uses fog and cloud computing. The proposed infrastructure aims to address different types of emergencies in a very simple and effective way. It is designed to optimize the emergency notification process. Emergency data are communicated to fog nodes, which alert the emergency response services and the victim's family members. Fog computing is integrated into the system to allow for the offloading of resource-intensive tasks and data pre-processing to fog nodes. The system's evaluation in a particular scenario showed that the overall delay obtained with fog computing was about six times shorter than when the end node conveyed data directly to the cloud.

Fog-based solutions have been proposed for the classification of patients with infectious diseases. For example, a fog-based mobile health system was proposed by Sareen et al. [56]. This system stores personal Zika virus symptoms and risk zone information on fog servers distributed at different locations in a risk area. Zika is one of the fastest spreading infectious viruses posing new threats to public health around the world. The authors implemented a fuzzy k-nearest neighbor (FKNN)-based classification system to classify users as uninfected or infected based on their Zika virus symptoms. The cloud storage and processing component consists of different modules: data collection, information protection, FKNN-based classification, geographic positioning system (GPS)-based risk assessment, and health communication. The fog server is constantly capturing user data and mosquito sensor data, so that any newly infected user or risk site is automatically identified. Singh et al. [57] proposed similar work to classify dengue patients into three classes (infected, uninfected, and severely infected). Dengue fever is a common viral disease transmitted by mosquitoes. The proposed architecture is a three-layer architecture that relies on fog computing as the intermediate layer between the IoT infrastructure and the cloud. The fog layer allows large amounts of dengue data to be stored after pre-processing. The authors' study found that the use of fog infrastructure resulted in reduced system latency and improved response and execution times for patient classification without affecting accuracy.

As we mentioned earlier, fog computing is not a substitute for cloud computing. Several efforts investigated the interoperability between cloud and fog computing in the context of healthcare systems [58–60]. Abdelmoneem et al. [60] proposed a healthcare fog–cloud architecture to reduce applications' delays and costs and meet their time constraints. They stated that using their proposed algorithms allows for dynamically distributing health tasks between fog nodes and cloud servers. Fog nodes are in charge of executing computation tasks such as data analysis and context management. The scheduling module formulates a mapping between tasks to be executed on fog nodes, represented as a bipartite graph, and cloud servers. The task dispatcher module assigns scheduled tasks to fog and cloud nodes. Paul et al. [58] presented a three-tier architecture for monitoring patient health using fog



computing and cloud computing. The architecture consists of the sensor infrastructure, fog computing resources, and cloud computing servers. The fog tier aggregates data received from edge devices. Then, it assigns processing and data analysis tasks to fog nodes and edge devices using a task-scheduling algorithm. Kumari et al. [59] also proposed a fog computing-based three-tier architecture to help caregivers, clinics, and hospitals offer smart health services to their patients. The architecture consists of a medical devices layer, a fog computing layer, and a cloud computing layer. It allows for the implementation of different operations in healthcare providers' data pipeline. These operations include data collection from medical devices, data processing and analysis in fog resources, and big data analytics on cloud servers.

Table 2 summarizes the above fog-based applications in healthcare and Figure 5 shows the typical components of our proposed fog–cloud-based architecture for healthcare. Several fog nodes can be deployed at various health facilities in a city. They can even be deployed in ambulances to handle emergency situations.

**Table 2.** Fog computing-based healthcare applications.

Application	Computing Model (Fog/Cloud)	Fog Nodes' Role	Benefits
Alert and emergency management architecture [55]	Fog/cloud	Optimize the emergency notification process	Overall delay reduced six times compared to a cloud-only solution
		Alert emergency services and the victim's family members	
		Offload resource-intensive tasks	
Mobile health system [56,57]	Fog/Cloud	Capture user and mosquito sensor data	Quick identification of any newly infected user or risk site.
	Fuzzy k-nearest neighbor (FKNN) based classification model [57]	Provide data storage and pre-processing	Reduced system latency. Improved response and execution times.
Monitoring of patients suffering from chronic diseases and other health service [58,59]	Fog/cloud	Aggregate and analyze data collected by edge devices	Increasing the efficiency of the entire system
		Distribute processing tasks to edge devices	
		Manage data pipeline from data acquisition to data analytics on the cloud	
Dynamic distribution and scheduling of health tasks [60]	Fog/cloud	Perform computations tasks (data analysis, context management, critical control)	Reduce application delays and costs and meet their time constraints

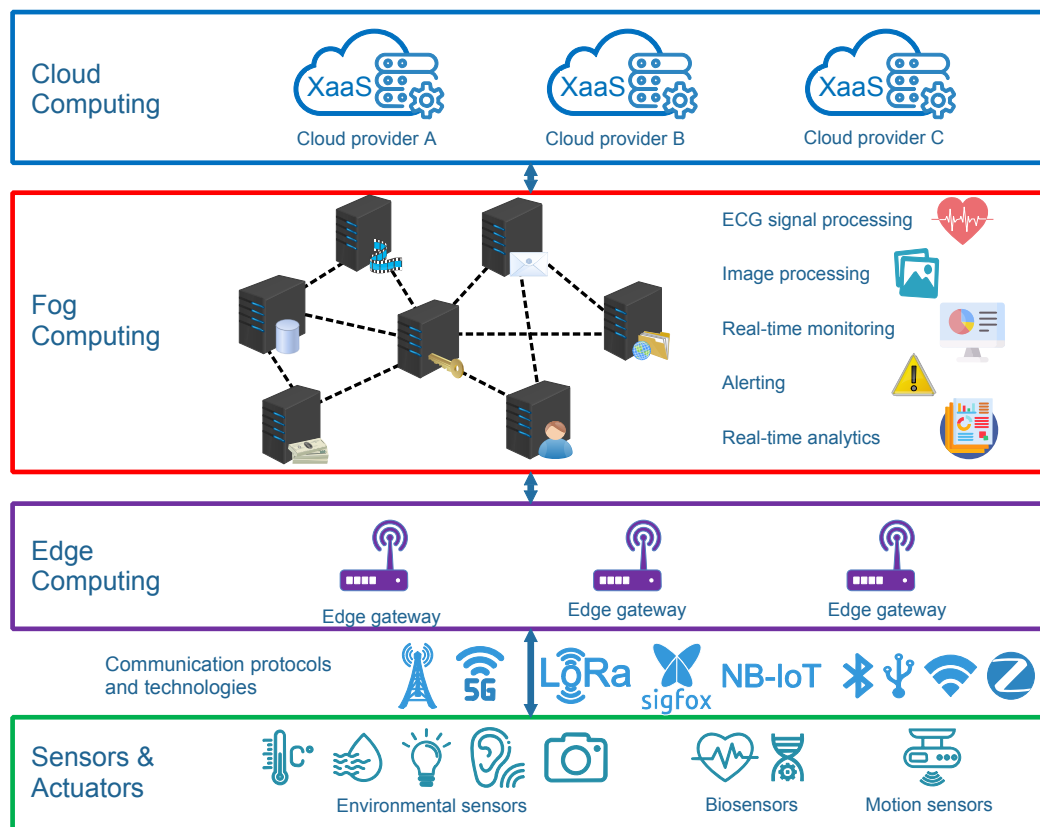


Figure 5. Fog–cloud-based architecture for healthcare.

### 6.3. Fog Computing in Smart Grid Architectures

The transformation of cities into smart cities depends mostly on their ability to modernize their power grids through the deployment of smart grids and their efficient management, which will allow for the integration of renewable energies and the production of clean energy close to needs. The smart grid represents a promising technology for integrating green energy resources into the power distribution system, controlling energy use and balancing the energy load. Smart grids rely on smart meters and devices to ensure bi-directional information flows in the power grid to manage and monitor electricity consumption. These devices produce massive amounts of data that smart grids need to harness to enable utilities to offer new pricing schemes that can increase energy efficiency and generate a more reliable power supply through improved power failure management. Improvements to the city power grid would also enable better integration of new technologies such as electric vehicles, which could serve as energy storage in emergencies, creating many opportunities for urban areas. Future prospects would include zero-emission transportation throughout the city.

Many scholars investigated the use of cloud computing to support the computational needs, information management, storage, and integration of smart grids [61–67]. S. Bera et al. [61] surveyed existing smart grids’ cloud-based applications, dealing with the three issues of energy management, information management, and security. They discussed the benefits of cloud computing applications to cope with these issues, and they provided insights into future opportunities for smart grid development. They also highlighted challenges that conventional power grids face that cloud-based solutions can solve. Furthermore, they overviewed the current state of research on smart grids and identified current challenges in cloud-based energy management, smart grid security, and information management. Fang et al. [62] described the benefits and opportunities of smart grid data management in the cloud. Moreover, they proposed a model linking the smart grid domain to the cloud computing domain and presented motivating applications.

Real-time data management and processing is one of the pressing issues for smart grids. A number of efforts investigated this concern in the context of the cloud [66–68]. Birman et al. [67] studied the requirements for deploying smart grid applications on cloud servers. They found that many promising energy management applications require scalability of the kind only cloud computing can provide. However, these applications also include additional requirements such as support for scalable real-time services, support for consistent and fault-tolerant services, and privacy protection. Cloud computing would not currently support these requirements. The authors in [66] proposed a cloud-based model for smart grid data management to meet the near real-time information retrieval needs of various energy market players. The distributed data management and parallel processing schemes are highly specialized in time series, which is the typical type of data generated by a smart grid. Therefore, a smart grid DaaS provider would provide its data management and processing services to any party with a legitimate interest in providing reusable services such as data collection, validation and cleansing, analysis, and data archiving. Simmhan et al. [68] investigated the issue of smart grid big data analytics in the cloud. They described a cloud-based platform for data-driven analytics to respond to dynamic consumer demand (D2R optimization), detecting the supply–demand mismatch, and preemptively correcting it. The platform ingests real-time data and uses a semantic data integration pipeline and scalable machine-learning models trained over massive historical datasets to predict the energy demand. The data pipeline relies on a private cloud infrastructure to ensure on-demand resource elasticity and to visualize current and historical energy consumption patterns. The authors used public cloud platforms, which provide powerful computing resources for analytics and reliable data hosting solutions suitable for distributed access. Private clouds offer the advantage of providing physical data security, but hardware management remains a concern. They are essential when data transfer latency is a concern. As they are expected to be generated by smart city grids, multi-terabyte data sets would favor private clouds due to the high costs of transferring and storing data in public clouds.

These studies show the need for fog–cloud-based platforms for smart grids, which are highly reliable and can ensure the efficient management of resources to support consumer and business operations. The last few years witnessed a growing trend in using fog computing in smart grids to address several of the concerns involved when the smart grid relies only on cloud servers for data storage and analysis [69–74]. Okay et al. [74] examined the cloud-based smart grid architectures' current state and highlighted the motivation to adopt fog computing as a technology catalyst for real-time smart grid analysis. They proposed a three-tier fog-based solution for a smart grid and described a use case scenario for the proposed model. Hussain et al. [72] also proposed a three-layer fog computing based smart grid framework. They characterized its features with regard to the integration of a massive number of IoT devices. Their evaluation of the framework on real-world parameters showed that for a network with approximately 50% time-critical applications, the overall service latency obtained using fog computing is nearly half that obtained when using a cloud-only solution. They also found that fog computing reduces the generic cloud computing model's aggregated power consumption by more than 44%. Zahoor et al. [71] investigated resource management in smart grids and proposed a fog–cloud-based model. They used five load-balancing techniques to evaluate its performance enhancement over a cloud computing model. These five algorithms are Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), Artificial Bee Colony (ABC), Round-Robin, and Throttled. Furthermore, the authors proposed a hybrid approach, which combines ABC and ACO. The results of their simulations revealed that their technique outperformed the other techniques.

Figure 6 depicts the typical components of a fog–cloud-based platform for smart grids. Several fog nodes can be deployed at various city facilities, including power generation facilities, transmission and distribution stations, and near power consumers (smart buildings and offices, smart homes, and manufacturing factories).

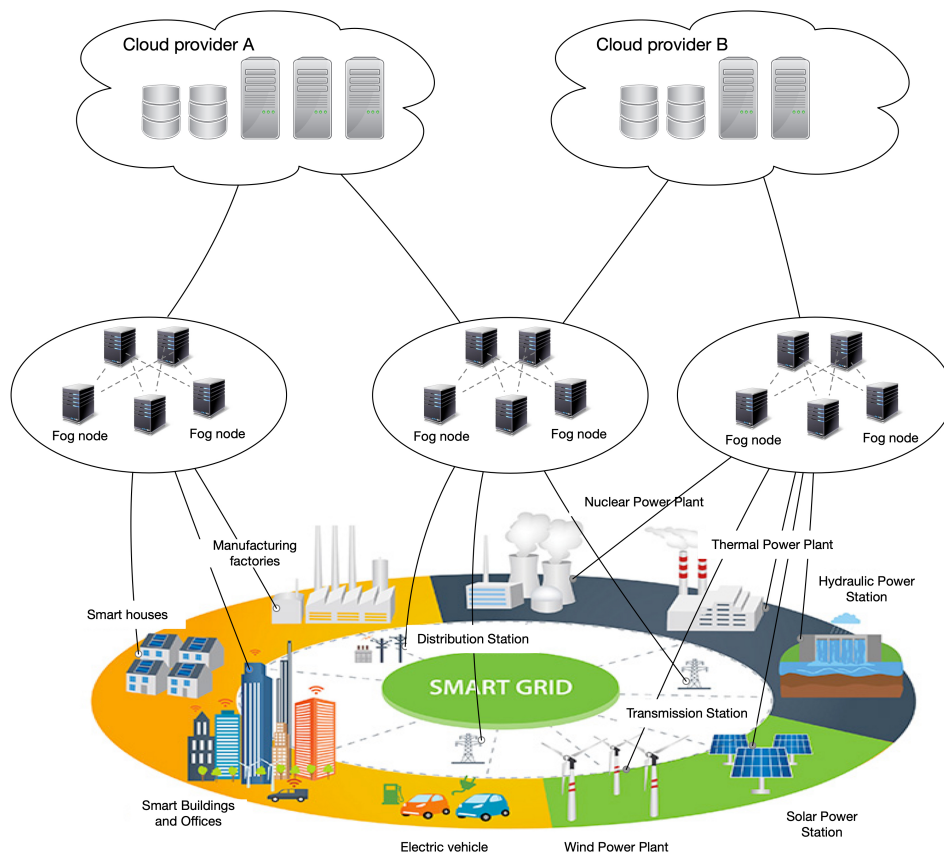


Figure 6. A fog–cloud-based architecture for smart grids.

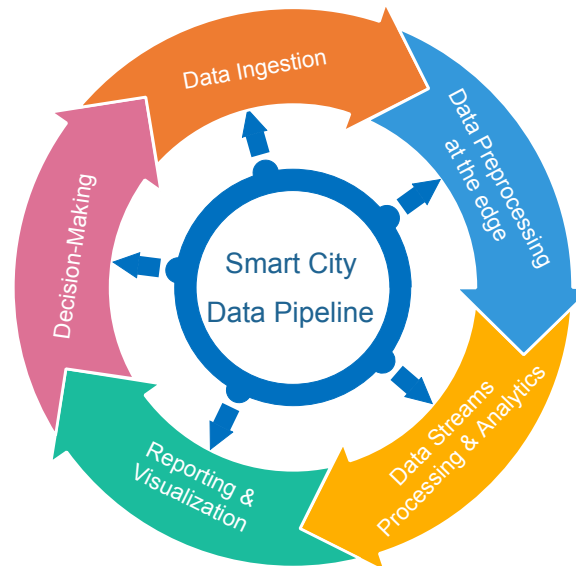
## 7. Toward a Fog-Based Real-Time Big Data Pipeline

### 7.1. Building a Smart City Data Pipeline

Smart city analyses typically implement a data pipeline by connecting one task’s output to another one’s input. Large-scale real-time data analysis would require the deployment of a fog-based data pipeline. Currently, Hadoop [36,37] is the cloud-based big data analytics platform widely used by the big data community. With Hadoop, computation-intensive and data-intensive analytics dispatch tasks to multiple nodes for execution. A real-time big data pipeline for smart cities should have essential characteristics to meet city stakeholders’ diverse demands. Therefore, a fog-based big data pipeline system for smart cities must have the following features:

- Scalable messaging system: The pipeline system should have a robust and scalable distributed messaging system such as Apache Kafka [21,22] to cope with the vast volume and high velocity of IoT data streams originating from the city systems.
- Data storage: The system should have enough storage capacity to allow performing data analytics using a robust big data platform like Apache Hadoop.
- Machine learning libraries: The system should support various machine learning libraries such as Tensorflow [24], Keras [75], Apache Spark MLlib [76,77] to perform predictive analytics.
- Backend store: The analytics output should be stored in some flexible database. A NoSQL database would be preferred.
- Dashboard and visualization tools: The system should have supporting reporting and visualization tools.

Building a fog-based data pipeline for a smart city is a process with five phases: data ingestion, data pre-processing, data processing and analytics, visualization and reporting, and decision making (see Figure 7).



**Figure 7.** Phases of a smart city data pipeline.

#### 7.1.1. Data Ingestion

This phase concerns the acquisition of data streams from various sensors and devices deployed in smart city facilities. These data streams can be in various data formats such as Extensible Markup Language (XML), JavaScript Object Notation (JSON), and text and can be stored in traditional relational database management system (RDBMS) data repositories or NoSQL databases. Additionally, to enable data persistence and consumption by multiple applications, publish–subscribe messaging systems like Apache Kafka must be supported.

#### 7.1.2. Data Preprocessing at the Edge

In this phase, the fusion of data received from various devices and sensors in smart city facilities could use well-known data integration approaches, including data propagation, data consolidation, data federation, and semantic data integration [78]. Edge gateways typically carry out this data fusion task. Data propagation is the transmission of collected data from data sources to target servers. Data consolidation is the process of integrating data received from multiple data sources and persistently storing it in a single data store. Semantic data integration involves controlled vocabularies, which require standardized terminologies to represent data elements in data repositories. Examples of controlled vocabularies are the various ontologies developed for smart cities [79–81]. Data federation uses specialized software to give users a single logical view that allows them to access and query data stored in one or more data stores.

#### 7.1.3. Data Streams Processing and Analytics

The processing of data streams would primarily use streaming processing engines like Apache Flume [82], Apache Flink [83,84], Apache Storm [85], Amazon Kinesis [86], and Apache Spark Streaming [87]. These streaming engines typically support many machine learning libraries like Tensorflow [24], Keras [75], PyTorch [88], Scikit-learn [89], and many more to make sense of unstructured data. The results of the data processing phase would significantly impact the smart city’s decisions and actions. Fei et al. [90] studied the different machine learning techniques used by several cyber–physical systems to analyze their data streams. The study, which focused more specifically on



the temporal complexity perspective, provided information and recommendations on how cloud and fog-based architectures should use machine learning techniques.

#### 7.1.4. Reporting and Visualization

This phase summarizes the results obtained from the processing and analysis of the data and presents them in a format accessible to users using visualization techniques. One of the regularly used presentation interfaces is the visual dashboard. It aggregates information from multiple sources and displays it to the user in graphical form. Business visualization tools include Public Tableau [91] and Qlik Sense Desktop [92], among others.

#### 7.1.5. Decision Making

The provision of smart services in smart cities is a process of continuous change. City stakeholders need to take an iterative approach to applying new technologies such as IoT. The approach is to continually assess the needs of cities, use appropriate technologies, collect data and analyze them to assess the impacts of technology applications, and make appropriate and well-informed decisions.

### 7.2. Implementation Scenario

As we have discussed earlier, the fog node's data processing components would typically include a data ingestion platform, a scalable messaging system, a data stream processing engine, and sufficient storage capacity. Apache Nifi [93], which automates the transfer of data between disparate systems, would allow the ingestion of data from different sources. It provides real-time control that makes it easy to manage data movement between all sources and destinations. It is data source agnostic and supports disparate and distributed sources of diverse formats, schemas, and protocols. Over the last decade, many publish–subscribe messaging systems have been developed. ActiveMQ [94], RabbitMQ [95], and Kafka are the most popular ones to date. They support several protocols including Advanced Message Queuing Protocol (AMQP), Message Queuing Telemetry Transport (MQTT) to exchange data with edge gateways. Apache Kafka's popularity is growing due to its advanced features for handling multiple data streams efficiently and scalably. Several real-time applications, such as social media analytics, telemetry from connected devices, network monitoring, clickstream analysis, and financial analytics and alerts, have successfully used Kafka. Many data processing engines like Apache Storm, Apache Flink, Apache Spark, Google Cloud Dataflow [96], and AWS IoT [97] can integrate well with Apache Kafka.

Figure 8 illustrates the software components required to implement data stream processing in a fog node effectively. They include Apache Nifi for data ingestion, a Kafka cluster with multiple topics, a Storm topology for data processing, and a NoSQL database like Cassandra [98] or MongoDB [99] and query tools Apache Drill [100], and visualization and monitoring dashboards. Apache Kafka is typically deployed and executed on a cluster of one or several brokers (also called servers). It can immutably store messages received from multiple data sources, called producers, in topics or queues. The topics are organized into multiple partitions replicated to all clusters' brokers to guarantee fault tolerance. Each partition stores incoming events or messages, with an index and a timestamp. Consumers, which can be different applications, can then query the messages stored in Kafka partitions. Apache Storm is an open-source real-time data processing engine designed to accept tons of high-speed incoming data, possibly from multiple sources, for analysis and for publishing real-time updates to a user interface or application without storing actual data. Apache Storm relies on Apache Zookeeper, which is an open-source platform for cluster state management. A Storm application for real-time data processing is typically designed and implemented as a directed acyclic graph workflow with spouts and bolts forming the graph's vertices. A storm workflow is called a topology. Topology edges, called streams, carry data from one vertex of the workflow to another. Thus, a data stream retrieved by a spout from a data source is routed to various bolts where the data are filtered, aggregated, analyzed, and then sent to a user interface such as a dashboard or other application. Apache Storm is highly

scalable and offers job processing guarantees. It can process more than a million tuples per second per node. Furthermore, it integrates very well with Apache Kafka to read and write data from Kafka topics using KafkaSpout and KafkaBolt.

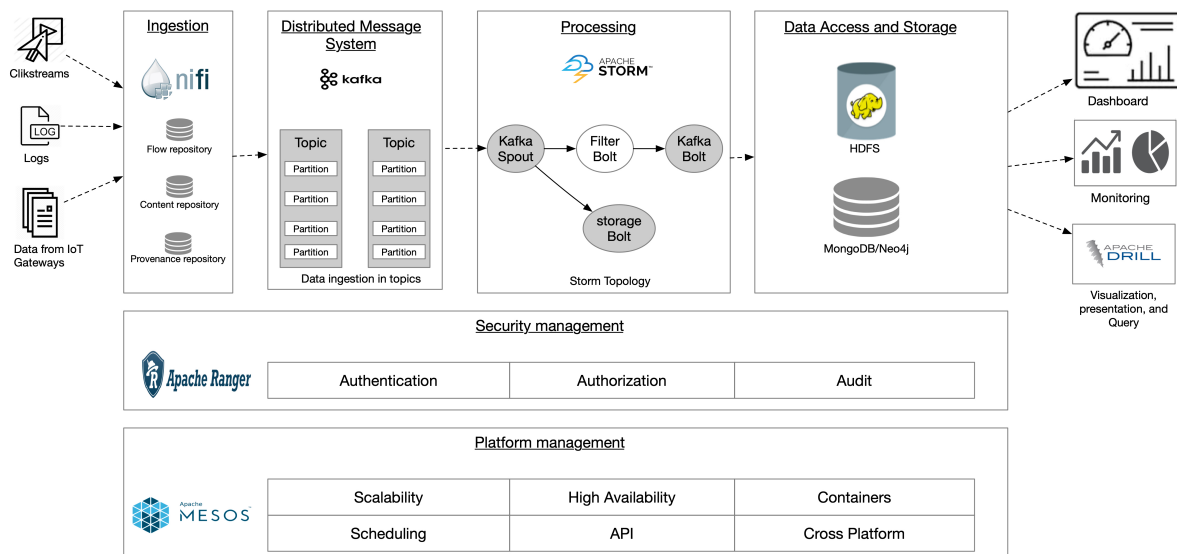


Figure 8. Data processing at the fog.

These software tools are not specific to fog computing solutions, but they are deployed by most public cloud providers such as Amazon AWS, Google Cloud, and Microsoft Azure to support large-scale data processing. They have also been used or suggested in many fog-based solutions [101–103].

We are at the early phases of implementing a fog node prototype to carry out data analytics at the edge using ThingsBoard IoT Gateway [104], an open-source IoT platform for the rapid ingestion of IoT data, a Kafka cluster and Apache Storm for data analytics. ThingsBoard provides a Kafka plugin to send messages to the Kafka cluster triggered by specific rules. Its web interface allows the user to describe organization assets, such as buildings or equipment, and their IoT devices, and its rule-chain interface allows for the specification of actions to be taken for input events such as telemetry measurements. ThingsBoard provides a generator to emulate temperature readings from a thermometer, which are then conveyed to the Kafka plugin.

## 8. Challenges and Open Research Issues

Today, smart cities use an array of smart sensors, edge devices, edge gateways, and fog nodes to collect urban real-time data for real-time decision making and insights. As in other distributed computing paradigms, several challenges must be overcome in the edge and fog computing ecosystem to enable smart city end users, service providers, and infrastructure providers to leverage the services provided by edge and fog servers. Four of the significant challenges and open research issues are security, privacy, interoperability, and characterizing smart city applications.

### 8.1. Security and Privacy

As we mentioned earlier in Section 2.2, edge computing and fog computing dramatically reduce the transfer of data to data centers outside the control of the organization. Analyzing sensitive data on a local gateway or fog node can deeply improve the privacy of the data. However, vulnerabilities come from the limited computing power of IoT devices, as well as hardware design flaws and firmware update constraints. Furthermore, edge and fog systems employ various network components to interconnect IoT devices, storage and computing devices, making them potential targets for various kinds of attacks. Network monitoring solutions can help detect anomalies and security

vulnerabilities [105]. These inconsistencies can provide hackers with an easy entry point into an edge or fog network. Therefore, the security of fog-based solutions should be enhanced at the network level. The network edge might be the place to insert security patches between vulnerable devices and the other components of the network.

Security challenges in fog computing can manifest themselves at three levels: edge and fog network infrastructures, fog nodes, and device level. An increasing number of efforts are aimed at solving security and privacy concerns in edge- and fog-based systems. For example, Roman et al. [106] provided a detailed analysis of security threats that can hamper the network infrastructure, user devices, edge data center, and virtualization infrastructure. They listed eight potential security challenges and described how existing solutions could address those challenges. Ni et al. [107] analyzed the security and privacy challenges in fog computing. They reviewed existing solutions and approaches and described general perspectives on security and privacy issues in fog-assisted systems. Tariq et al. [108] discussed the security requirements of fog-assisted systems in smart healthcare and smart grids. They described fog computing's security challenges and the trust and privacy issues surrounding big data in these fog-assisted systems. Furthermore, they discussed the potential of blockchain technology to address many of these security concerns. In [109], the authors proposed a solution for reliable smart city service delivery at the edge, which utilizes collaboration between edge servers and privacy mediation nodes and an intrusion detection system to improve the security, reliability, and availability of smart city applications.

## 8.2. Interoperability

Interoperability in fog-assisted systems is another major challenge that these systems need to overcome. Several concerns arise from the absence of interoperability between deployed equipment and devices. These concerns are:

- Difficulty integrating and deploying devices and equipment made by different manufacturers, having different types of connectors, using different data formats and supporting different communication protocols.
- A lack of common monitoring platforms to monitor these devices,
- A lack of common interfaces to pull and push information from/to these devices,
- A lack of common techniques and approaches for testing the Application Programming Interfaces (APIs) of these devices,
- Difficulty in using security software offered by third parties to secure devices.

Currently, non-interoperability in IoT is addressed with the deployment of intermediary components, supporting multiple protocols, to resolve the above issues. An intermediary component, such as An IoT gateway, typically performs several critical functions from translating protocols to aggregating, filtering, encrypting, processing, and managing data. Several efforts investigated the issue of IoT interoperability and proposed intelligent edge gateways [110–113]. Morabito et al. [112] presented a lightweight edge gateway for IoT (LEGIOT) architecture, which uses microservices and thin virtualization technologies to ensure an extensible and flexible solution. By using the socket-proxy framework and container-based virtualization, their solution aims to make edge gateways suitable for various IoT protocols/applications. It also aims to allow optimized management of resources while taking into account their needs in terms of energy efficiency, multi-location, and interoperability. Tuli et al. [113] proposed a blockchain-based framework for edge and fog computing called FogBus, where the main components are Fog Gateway Nodes (FGNs). FogBus provides front-end interfaces to users to perform several operations such as accessing back-end services, managing IoT devices, and requesting resources through FGNs, which filter data from different sources, aggregate them, and organize them into a standard format. They forward data to other computation instances in the environment in the case of large processing requirements and exchange data with fog nodes using the Constrained Application Protocol (CoAP) or the Simple Network Management Protocol (SNMP).

### 8.3. Characterizing and Mapping Smart City Applications

Applications have diverse requirements. Some of them have low data rates and/or low latency requirements, while others like medical imaging or video surveillance have very high data rates. Therefore, IoT applications need to be characterized to determine the key issues in mapping them to an edge device, fog, and cloud hierarchy; then, allocation and scheduling algorithms need to be created to map workloads on top of the edge, fog, and cloud systems [114]. Furthermore, user and vehicle mobility are scenarios that need to be characterized to allocate required resources in the edge–cloud continuum [115,116]. Mobility may involve moving the state of an application from one service to another or moving an entire application and its dependencies [117].

Smart city services are typically composed and executed as dependent task workflows. These composite services often experience delays if they rely on cloud solutions. In [118], the authors proposed personalized multimedia services delivery by composing service-specific overlays (SSOs) to take advantage of mobile edge computing's recent advancements. They presented a workflow network-based approach to mobile edge nodes' cooperation in a fog–cloud hierarchy to form guaranteed SSOs.

## 9. Conclusions

Over the past few years, many smart cities have slowly moved from in-house IT infrastructure to utility-delivered computing over the Internet. Here, we reviewed edge and fog computing and described how smart cities could benefit from them. Edge and fog computing hold great promise in effectively solving big data storage problems and rapid data analysis to react quickly to various events in a city that require immediate decision making and action. We reviewed the existing literature on fog computing-based solutions in three crucial smart city areas: Intelligent Transportation Systems, healthcare, and smart grids. A smart city data pipeline should use high-speed data transfer technologies and integrate data streams and software processing tools to help smart cities develop time-sensitive applications. We propose a fog-based data pipeline for IoT data management and processing in a smart city. Data processing and analytics need to rely on robust and highly scalable messaging systems, powerful software engines for data stream processing, and scalable data storage solutions. Security, privacy, interoperability, and characterizing and mapping of smart city applications to devices in the edge/cloud continuum remain the main open research issues and challenges that need to be addressed to deploy fog-based solutions in smart cities successfully. We highlighted some of the efforts towards addressing these challenges.

**Author Contributions:** E.B.: methodology, investigation, writing—original draft, review and editing. E.S.: drawing figures, writing smart grids use case subsection, review and editing. Z.M.: writing transport use case subsection, review and editing. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Acknowledgments:** This work is supported by the UAEU Program for Advanced Research Grant N. G00003443.

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

## References

1. Anjum, A.; Soomro, K.; Tahir, M.A. Towards cloud based big data analytics for smart future cities. *J. Cloud Comput.* **2015**, *4*, 49–11.
2. Goli-Malekabadi, Z.; Sargolzaei-Javan, M.; Akbari, M.K. An effective model for store and retrieve big health data in cloud computing. *Comput. Methods Programs Biomed.* **2016**, *132*, 75–82. [[CrossRef](#)]
3. Ji, Z.; Ganchev, I.; O'Droma, M.; Zhao, L.; Zhang, X. A cloud-based car parking middleware for IoT-based smart cities: Design and implementation. *Sensors* **2014**, *14*, 22372–22393. [[CrossRef](#)] [[PubMed](#)]
4. Mutlag, A.A.; Ghani, M.K.A.; Arunkumar, N.; Mohammed, M.A.; Mohd, O. Enabling technologies for fog computing in healthcare IoT systems. *Future Gener. Comput. Syst.* **2019**, *90*, 62–78. [[CrossRef](#)]

5. Gartner. Gartner Glossary. Available online: <https://www.gartner.com/en/information-technology/glossary/edge-computing> (accessed on 15 September 2020).
6. Satyanarayanan, M.; Simoens, P.; Xiao, Y.; Pillai, P.; Chen, Z.; Ha, K.; Hu, W.; Amos, B. Edge Analytics in the Internet of Things. *IEEE Pervasive Comput.* **2015**, *14*, 24–31. [[CrossRef](#)]
7. IDC.com. IDC FutureScape: Worldwide Internet of Things 2017 Predictions. 2016. Available online: <https://www.idc.com/research/viewtoc.jsp?containerId=US40755816> (accessed on 2 May 2020).
8. Satyanarayanan, M.; Bahl, P.; Cáceres, R.; Davies, N. The case for vm-based cloudlets in mobile computing. *Pervasive Comput. IEEE* **2009**, *8*, 14–23. [[CrossRef](#)]
9. Bonomi, F.; Milito, R.; Zhu, J.; Addepalli, S. Fog Computing and Its Role in the Internet of Things. In Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing, MCC '12, Helsinki, Finland, 13–17 August 2012; ACM: New York, NY, USA, 2012; pp. 13–16. [[CrossRef](#)]
10. OpenFog Consortium Architecture Working Group. OpenFog Reference Architecture for Fog Computing. Available online: [https://www.iiconsortium.org/pdf/OpenFog\\_Reference\\_Architecture\\_2\\_09\\_17.pdf](https://www.iiconsortium.org/pdf/OpenFog_Reference_Architecture_2_09_17.pdf) (accessed on 16 July 2020).
11. Cisco.com. Cisco Fog Computing Solutions: Unleash the Power of the Internet of Things. 2015. Available online: [https://www.cisco.com/c/dam/en\\_us/solutions/trends/iot/docs/computing-solutions.pdf](https://www.cisco.com/c/dam/en_us/solutions/trends/iot/docs/computing-solutions.pdf) (accessed on 1 July 2020).
12. Iorga, M.; Feldman, L.; Barton, R.; Martin, M.J.; Goren, N.S.; Mahmoudi, C. Fog Computing Conceptual Model. Available online: <https://www.nist.gov/publications/fog-computing-conceptual-model> (accessed on 16 May 2019).
13. Yannuzzi, M.; van Lingen, F.; Jain, A.; Parellada, O.L.; Flores, M.M.; Carrera, D.; Perez, J.L.; Montero, D.; Chacin, P.; Corsaro, A.; et al. A new era for cities with fog computing. *IEEE Internet Comput.* **2017**, *21*, 54–67. [[CrossRef](#)]
14. Anawar, M.R.; Wang, S.; Zia, M.A.; Jadoon, A.K.; Akram, U.; Raza, S. Fog Computing—An Overview of Big IoT Data Analytics. *Wirel. Commun. Mob. Comput.* **2018**, *2018*, 1–22. [[CrossRef](#)]
15. Tordera, E.M.; Masip-Bruin, X.; Garcia-Alminana, J.; Jukan, A.; Ren, G.J.; Zhu, J.; Farre, J. What Is a Fog Node A Tutorial on Current Concepts towards a Common Definition. *arXiv* **2016**, arXiv:1611.09193.
16. Roca, D.; Quiroga, J.V.; Valero, M.; Nemirovsky, M. Fog Function Virtualization—A flexible solution for IoT applications. In Proceedings of the 2017 Second International Conference on Fog and Mobile Edge Computing (FMEC), Valencia, Spain, 8–11 May 2017; pp. 74–80.
17. Panwar, N.; Singh, A.K. A survey on 5G—The next generation of mobile communication. *Phys. Commun.* **2016**, *18*, 64–84. [[CrossRef](#)]
18. Yu, S.; Li, J.; Wu, J. Emergent LBS: If GNSS Fails, How Can 5G-enabled Vehicles Get Locations Using Fogs? In Proceedings of the 15th International Wireless Communications & Mobile Computing Conference (IWCMC), Tangier, Morocco, 24–28 June 2019; pp. 597–602.
19. Mell, P.; Grance, T. The NIST Definition of Cloud Computing. Available online: <http://faculty.winthrop.edu/domanm/csci411/Handouts/NIST.pdf> (accessed on 16 May 2019).
20. Hivcell Is the Premiere Platform as a Service for Edge Computing. Available online: <https://hivcell.com> (accessed on 16 July 2020).
21. Apache Kafka. Available online: <https://kafka.apache.org/> (accessed on 15 September 2020).
22. Kreps, J.; Narkhede, N.; Rao, J. Kafka: A distributed messaging system for log processing. In Proceedings of the Sixth International Workshop on Networking Meets Databases Workshop, Athens, Greece, 12 June 2011; pp. 1–7.
23. Kubernetes—Production-Grade Container Orchestration, Automated container Deployment, Scaling, and Management. Available online: <https://kubernetes.io/> (accessed on 15 September 2020).
24. Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M.; et al. TensorFlow: A system for large-scale machine learning. In Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI '16), Savannah, GA, USA, 2–4 November 2016.
25. Olson, J.A. Data as a Service: Are We in the Clouds? *J. Map Geogr. Libr.* **2009**, *6*, 76–78.
26. Plebani, P.; Salnitri, M.; Vitali, M. Fog Computing and Data as a Service: A Goal-Based Modeling Approach to Enable Effective Data Movements. In *International Conference on Advanced Information Systems Engineering CAiSE 2018, Advanced Information Systems Engineering*; Springer: Cham, Switzerland, 2018; pp. 203–219.



27. Zhao, H.; Ma, L. Big Data Compression of Smart Distribution Systems Based on Tensor Tucker Decomposition. *Chin. Soc. Electr. Eng.* **2019**, *39*, 4744–4752.
28. Salvador-Meneses, J.; Ruiz-Chavez, Z.; Garcia-Rodriguez, J. Low level big data compression. In Proceedings of the 10th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management, Seville, Spain, 18–20 September 2018.
29. He, X.; Wang, K.; Huang, H.; Liu, B. QoE-Driven Big Data Architecture for Smart City. *IEEE Commun. Mag.* **2018**, *56*, 88–93.
30. Rathore, M.M.; Paul, A.; Hong, W.H.; Seo, H.; Awan, I.; Saeed, S. Exploiting IoT and big data analytics Defining Smart Digital City using real-time urban data. *Sustain. Cities Soc.* **2018**, *40*, 600–610.
31. Bellini, P.; Nesi, P.; Paolucci, M.; Zaza, I. Smart City Architecture for Data Ingestion and Analytics: Processes and Solutions. In Proceedings of the 2018 IEEE Fourth International Conference on Big Data Computing Service and Applications (BigDataService), Bamberg, Germany, 26–29 March 2018; pp. 137–144.
32. Tang, B.; Chen, Z.; Hefferman, G.; Pei, S.; Wei, T.; He, H.; Yang, Q. Incorporating Intelligence in Fog Computing for Big Data Analysis in Smart Cities. *IEEE Trans. Ind. Inf.* **2017**, *13*, 2140–2150.
33. Yassine, A.; Singh, S.; Hossain, M.S.; Muhammad, G. IoT big data analytics for smart homes with fog and cloud computing. *Future Gener. Comput. Syst.* **2019**, *91*, 563–573.
34. Qin, B.; Tang, H.; Chen, H.; Cui, L.; Liu, J.; Yu, X. Review on big data application of medical system based on fog computing and IoT technology. *J. Phys.* **2019**, 1423, 012030.
35. Nguyen, S.; Salcic, Z.; Zhang, X. Big Data Processing in Fog—Smart Parking Case Study. In Proceedings of the IEEE International Conference on Parallel & Distributed Processing with Applications, Ubiquitous Computing & Communications, Big Data & Cloud Computing, Social Computing & Networking, Sustainable Computing & Communications (ISPA/IUCC/BDCLOUD/SocialCom/SustainCom), Melbourne, Australia, 11–13 December 2019; pp. 127–134.
36. Apache Hadoop. Available online: <http://hadoop.apache.org> (accessed on 15 September 2020).
37. Shvachko, K.; Kuang, H.; Radia, S.; Chansler, R. The Hadoop distributed file system. In Proceedings of the IEEE 26th Symposium on Mass Storage Systems and Technologies, MSST2010, Incline Village, NV, USA, 3–7 May 2010; pp. 1–10.
38. Díaz-De-Arcaya, J.; Miñón, R.; Torre-Bastida, A.I. Towards an architecture for big data analytics leveraging edge/fog paradigms. In *ACM International Conference Proceeding Series*; TECNALIA: Donostia-San Sebastian, Spain, 2019; pp. 173–176.
39. Tang, C.; Wei, X.; Zhu, C.; Chen, W.; Rodrigues, J.J. Towards smart parking based on fog computing. *IEEE Access* **2018**, *6*, 70172–70185.
40. Munir, A.; Kansakar, P.; Khan, S.U. IFCIoT - Integrated Fog Cloud IoT—A novel architectural paradigm for the future Internet of Things. *IEEE Consum. Electron. Mag.* **2017**, *6*, 74–82.
41. Chun, S.; Shin, S.; Seo, S.; Eom, S.; Jung, J.; Lee, K.H. A Pub/Sub-Based Fog Computing Architecture for Internet-of-Vehicles. In Proceedings of the 2016 IEEE International Conference on Cloud Computing Technology and Science (CloudCom), Luxembourg, 12–15 December 2016; pp. 90–93.
42. Hou, X.; Li, Y.; Chen, M.; Wu, D.; Jin, D.; Chen, S. Vehicular Fog Computing—A Viewpoint of Vehicles as the Infrastructures. *IEEE Trans. Veh. Technol.* **2016**, *65*, 3860–3873.
43. 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](#)]
44. Xiao, Y.; Zhu, C. Vehicular fog computing: Vision and challenges. In Proceedings of the 2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), Kona, HI, USA, 13–17 March 2017; pp. 6–9.
45. Du, H.; Leng, S.; Wu, F.; Chen, X.; Mao, S. A New Vehicular Fog Computing Architecture for Cooperative Sensing of Autonomous Driving. *IEEE Access* **2020**, *8*, 10997–11006. [[CrossRef](#)]
46. Belakaria, S.; Ammous, M.; Sorour, S.; Abdel-Rahim, A. Fog-based multi-class dispatching and charging for autonomous electric mobility on-demand. *IEEE Trans. Intell. Transport. Syst.* **2019**, *21*, 762–776. [[CrossRef](#)]
47. Sun, G.; Zhang, F.; Liao, D.; Yu, H.; Du, X.; Guizani, M. Optimal energy trading for plug-in hybrid electric vehicles based on fog computing. *IEEE Internet Things J.* **2019**, *6*, 2309–2324. [[CrossRef](#)]
48. Darwish, T.S.J.; Abu Bakar, K. 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](#)]

49. Xiao, X.; Hou, X.; Chen, X.; Liu, C.; Li, Y. Quantitative analysis for capabilities of vehicular fog computing. *Inf. Sci.* **2019**, *501*, 742–760. [[CrossRef](#)]
50. Minh Dang, L.; Piran, M.J.; Han, D.; Min, K.; Moon, H. A survey on internet of things and cloud computing for healthcare. *Electronics* **2019**, *8*, 768. [[CrossRef](#)]
51. Elhoseny, M.; Abdelaziz, A.; Salama, A.S.; Riad, A.M.; Muhammad, K.; Sangaiah, A.K. A hybrid model of Internet of Things and cloud computing to manage big data in health services applications. *Future Gener. Comput. Syst.* **2018**, *86*, 1383–1394. [[CrossRef](#)]
52. Calabrese, B.; Cannataro, M. Cloud Computing in Healthcare and Biomedicine. *Scalable Comput. Pract. Exp.* **2015**, *16*, 1–18. [[CrossRef](#)]
53. Kundella, S.; Gobinath, R. A survey on big data analytics in medical and healthcare using cloud computing. *Int. J. Sci. Technol. Res.* **2019**, *8*, 1061–1065.
54. Kraemer, F.A.; Braten, A.E.; Tamkittikhun, N.; Palma, D. Fog Computing in Healthcare—A Review and Discussion. *IEEE Access* **2017**, *5*, 9206–9222. [[CrossRef](#)]
55. Aazam, M.; Huh, E.N. E-HAMC: Leveraging Fog computing for emergency alert service. In Proceedings of the 2015 IEEE International Conference on Pervasive Computing and Communication Workshops, PerCom Workshops, Kyung Hee University, Seoul, Korea, 23–27 March 2015; pp. 518–523.
56. Sareen, S.; Gupta, S.K.; Sood, S.K. An intelligent and secure system for predicting and preventing Zika virus outbreak using Fog computing. *Enterp. Inf. Syst.* **2017**, *11*, 1436–1456. [[CrossRef](#)]
57. Singh, S.; Bansal, A.; Sandhu, R.; Sidhu, J. Fog computing and IoT based healthcare support service for dengue fever. *Int. J. Pervasive Comput. Commun.* **2018**, *14*, 197–207. [[CrossRef](#)]
58. Paul, A.; Pinjari, H.; Hong, W.H.; Seo, H.C.; Rho, S. Fog Computing-Based IoT for Health Monitoring System. *J. Sens.* **2018**, *2018*, 1386470. [[CrossRef](#)]
59. Kumari, A.; Tanwar, S.; Tyagi, S.; Kumar, N. Fog computing for Healthcare 4.0 environment: Opportunities and challenges. *Comput. Electr. Eng.* **2018**, *72*, 1–13. [[CrossRef](#)]
60. Abdelmoneem, R.M.; Benslimane, A.; Shaaban, E.; Abdelhamid, S.; Ghoneim, S. A Cloud-Fog Based Architecture for IoT Applications Dedicated to Healthcare. In Proceedings of the ICC 2019—2019 IEEE International Conference on Communications (ICC), Shanghai, China, 20–24 May 2019; pp. 1–6.
61. Bera, S.; Misra, S.; Joel, J.P.C. Rodrigues. Cloud Computing Applications for Smart Grid—A Survey. *IEEE Trans. Parallel Distrib. Syst.* **2015**, *26*, 1477–1494. [[CrossRef](#)]
62. Fang, X.; Misra, S.; Xue, G.; Yang, D. Managing smart grid information in the cloud: Opportunities, model, and applications. *IEEE Netw.* **2012**, *26*, 32–38. [[CrossRef](#)]
63. Yigit, M.; Gungor, V.C.; Baktir, S. Cloud Computing for Smart Grid applications. *Comput. Netw.* **2014**, *70*, 312–329. [[CrossRef](#)]
64. Baek, J.; Vu, Q.H.; Liu, J.K.; Huang, X.; Xiang, Y. A secure cloud computing based framework for big data information management of smart grid. *IEEE Trans. Cloud Comput.* **2015**, *3*, 233–244. [[CrossRef](#)]
65. Bitzer, B.; Gebretsadik, E. Security issues in cloud-based smart grid applications. *Renew. Energy Power Qual. J.* **2016**, *1*, 126–132. [[CrossRef](#)]
66. Rusitschka, S.; Eger, K.; Gerdes, C. Smart Grid Data Cloud: A Model for Utilizing Cloud Computing in the Smart Grid Domain. In Proceedings of the 1st IEEE International Conference on Smart Grid Communications (SmartGridComm), Gaithersburg, MD, USA, 4–6 October 2010; pp. 483–488.
67. Birman, K.P.; Ganesh, L.; van Renesse, R. Running Smart Grid Control Software on Cloud Computing Architectures. In Proceedings of the DOE Workshop on Computational Needs for the Next Generation Electric Grid, Ithaca, NY, USA, 19–20 April 2011.
68. Simmhan, Y.; Aman, S.; Kumbhare, A.G.; Liu, R.; Stevens, S.; Zhou, Q.; Prasanna, V.K. Cloud-Based Software Platform for Big Data Analytics in Smart Grids. *Comput. Sci. Eng.* **2013**, *15*, 38–47. [[CrossRef](#)]
69. Barik, R.K.; Gudey, S.K.; Reddy, G.G.; Pant, M.; Dubey, H.; Mankodiya, K.; Kumar, V. FogGrid: Leveraging Fog Computing for Enhanced Smart Grid Network. In Proceedings of the 14th IEEE India Council International Conference (INDICON), Roorkee, India, 15–17 December 2017; pp. 1–6.
70. Jalali, F.; Vishwanath, A.; De Hoog, J.; Suits, F. Interconnecting Fog computing and microgrids for greening IoT. In Proceedings of the IEEE Innovative Smart Grid Technologies—Asia (ISGT-Asia), Roorkee, India, 15–17 December 2016; pp. 693–698.
71. Zahoor, S.; Javaid, S.; Javaid, N.; Ashraf, M.; Ishmanov, F.; Afzal, M. Cloud–Fog–Based Smart Grid Model for Efficient Resource Management. *Sustainability* **2018**, *10*, 2079. [[CrossRef](#)]

72. Hussain, M.M.; Beg, M.M.S. Fog Computing for Internet of Things (IoT)-Aided Smart Grid Architectures. *Big Data Cogn. Comput.* **2019**, *3*, 8. [CrossRef]
73. Barros, E.B.C.; Filho, D.M.L.; Batista, B.G.; Kuehne, B.T.; Peixoto, M.L.M. Fog Computing Model to Orchestrate the Consumption and Production of Energy in Microgrids. *Sensors* **2019**, *19*, 2642. [CrossRef]
74. Okay, F.Y.; Ozdemir, S. A fog computing based smart grid model. In Proceedings of the 2016 International Symposium on Networks, Computers and Communications (ISNCC), Yasmine Hammamet, Tunisia, 11–13 May 2016.
75. Chollet, F. Keras. 2016. Available online: <https://keras.io/> (accessed on 15 September 2020).
76. Apache Spark MLlib. Available online: <https://spark.apache.org/mllib> (accessed on 15 September 2020).
77. Zaharia, M.; Chowdhury, M.; Franklin, M.J.; Shenker, S.; Stoica, I. Spark—Cluster Computing with Working Sets. In Proceedings of the 2nd USENIX Conference on Hot Topics in Cloud Computing, Boston, MA, USA, 22 June 2010.
78. Badidi, E.; Muthucumar, M. Towards a Platform for Urban Data Management, Integration and Processing. In Proceedings of the 3rd International Conference on Internet of Things, Big data, and Security (IoTBDs 2018), Madeira, Portugal, 19–21 March 2018.
79. Pierfrancesco, B.; Monica, B.; Riccardo, B.; Paolo, N.; Nadia, R. Km4City ontology building vs data harvesting and cleaning for smart-city services. *J. Vis. Lang. Comput.* **2014**, *25*, 827–839.
80. Compton, M.; Barnaghi, P.M.; Bermudez, L.; Garcia-Castro, R.; Corcho, Ó.; Cox, S.J.D.; Graybeal, J.; Hauswirth, M.; Henson, C.A.; Herzog, A.; et al. The SSN ontology of the W3C semantic sensor network incubator group. *J. Web Semant.* **2012**, *17*, 25–32. [CrossRef]
81. Nemirovski, G.; Nolle, A.; Sicilia, Á.; Ballarini, I.; Corado, V. Data integration driven ontology design, case study smart city. In Proceedings of the 3rd International Conference on Web Intelligence, Mining and Semantics, Madrid, Spain, 12–14 June 2016; ACM Press: New York, NY, USA, 2013.
82. Apache Flume. 2012. Available online: <https://flume.apache.org/> (accessed on 15 September 2020).
83. Carbone, P.; Katsifodimos, A.; Ewen, S.; Markl, V.; Haridi, S.; Tzoumas, K. Apache Flink™—Stream and Batch Processing in a Single Engine. *IEEE Data Eng. Bull.* **2015**, *36*.
84. Apache Flink—Stateful Computations over Data Streams. Available online: <https://flink.apache.org/> (accessed on 15 September 2020).
85. Apache Storm. Available online: <https://storm.apache.org/> (accessed on 15 September 2020).
86. AWS. Amazon Kinesis, Easily Collect, Process, and Analyze Video and Data Streams in Real Time. Available online: <https://aws.amazon.com/kinesis/> (accessed on 15 September 2020).
87. Apache Spark Lightning-Fast Unified Analytics Engine. Available online: <https://spark.apache.org/> (accessed on 15 September 2020).
88. PyTorch An Open Source Machine Learning Framework That Accelerates the Path from Research Prototyping to Production Deployment. Available online: <https://pytorch.org/> (accessed on 15 September 2020).
89. Scikit-Learn Machine Learning in Python. Available online: <https://scikit-learn.org/stable/> (accessed on 15 September 2020).
90. 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]
91. Tableau Public. Available online: <http://public.tableau.com/> (accessed on 15 September 2020).
92. Qlik Sense Data Analytics Platform. Available online: <https://www.qlik.com/us/products/qlik-sense> (accessed on 15 September 2020).
93. Apache Nifi an Easy to Use, Powerful, and Reliable System to Process and Distribute Data. Available online: <https://nifi.apache.org> (accessed on 15 September 2020).
94. ActiveMQ Flexible and Powerful Open Source Multi-Protocol Messaging. Available online: <http://activemq.apache.org/> (accessed on 15 September 2020).
95. RabbitMQ. Available online: <https://www.rabbitmq.com/> (accessed on 15 September 2020).
96. Google Cloud Dataflow. Available online: <https://cloud.google.com/dataflow> (accessed on 15 September 2020).
97. AWS IoT. Available online: <https://aws.amazon.com/iot/> (accessed on 15 September 2020).
98. Cassandra—Manage Massive Amounts of Data, Fast, without Losing Sleep. Available online: <https://cassandra.apache.org> (accessed on 15 September 2020).

99. MongoDB—The Database for Modern Applications. Available online: <https://www.mongodb.com> (accessed on 15 September 2020).
100. Apache Drill—Schema-free SQL Query Engine for Hadoop, NoSQL and Cloud Storage. Available online: <https://drill.apache.org/> (accessed on 15 September 2020).
101. Yang, S. IoT Stream Processing and Analytics in the Fog. *IEEE Commun. Mag.* **2017**, *55*, 21–27. [[CrossRef](#)]
102. Zhang, Q.; Zhang, Q.; Shi, W.; Zhong, H. Firework: Data Processing and Sharing for Hybrid Cloud-Edge Analytics. *IEEE Trans. Parallel Distrib. Syst.* **2018**, *29*, 2004–2017. [[CrossRef](#)]
103. Battulga, D.; Miorandi, D.; Tedeschi, C. FogGuru—A Fog Computing platform based on Apache Flink. In Proceedings of the 23rd Conference on Innovation in Clouds, Internet and Networks (ICIN 2020), Paris, France, 24–27 February 2020.
104. ThingsBoard Open-Source IoT Platform. Available online: <https://thingsboard.io/> (accessed on 15 October 2020).
105. Miller, K.B.; Brandon, E.R. Improving Network Monitoring and Security via Visualization. *arXiv* **2015**, arXiv:1511.08795v1.
106. Roman, R.; Lopez, J.; Mambo, M. Mobile edge computing, Fog et al.: A survey and analysis of security threats and challenges. *Future Gener. Comput. Syst.* **2018**, *78*, 680–698. [[CrossRef](#)]
107. Ni, J.; Zhang, K.; Lin, X.; Shen, X.S. Securing Fog Computing for Internet of Things Applications: Challenges and Solutions. *IEEE Commun. Surv. Tutor.* **2018**, *20*, 601–628. [[CrossRef](#)]
108. Tariq, N.; Asim, M.; Al-Obeidat, F.; Farooqi, M.Z.; Baker, T.; Hammoudeh, M.; Ghafir, I. The Security of Big Data in Fog-Enabled IoT Applications Including Blockchain—A Survey. *Sensors* **2019**, *19*, 1788. [[CrossRef](#)]
109. Jararweh, Y.; Otoum, S.; Cities, A.R.S.; Society, I. Trustworthy and sustainable smart city services at the edge. *Sustain. Cities Soc.* **2020**, *62*, 102394. [[CrossRef](#)]
110. Bimschas, D.; Hellbrück, H.; Mietz, R.; Pfisterer, D.; Römer, K.; Teubler, T. Middleware for smart gateways connecting sensornets to the internet. In Proceedings of the 5th International Workshop on Middleware Tools, Services and Run-Time Support for Sensor Networks, Bangalore, India, 29 November–3 December 2010; ACM Press: New York, NY, USA, 2010; pp. 8–14.
111. Datta, S.K.; Bonnet, C.; Nikaiein, N. An IoT gateway centric architecture to provide novel M2M services. In Proceedings of the 2014 IEEE World Forum on Internet of Things (WF-IoT), Seoul, Korea, 6–8 March 2014.
112. Morabito, R.; Petrolo, R.; Loscrì, V.; Mitton, N. LEGIoT—A Lightweight Edge Gateway for the Internet of Things. *Future Gener. Comp. Syst.* **2018**, *81*, 1–15. [[CrossRef](#)]
113. 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](#)]
114. Cheng, B.; Solmaz, G.; Cirillo, F.; Kovacs, E.; Terasawa, K.; Kitazawa, A. FogFlow: Easy Programming of IoT Services Over Cloud and Edges for Smart Cities. *IEEE Internet Things J.* **2018**, *5*, 696–707. [[CrossRef](#)]
115. User mobility aware task assignment for Mobile Edge Computing. *Future Gener. Comput. Syst.* **2018**, *85*, 1–8. [[CrossRef](#)]
116. Liao, S.; Dong, M.; Ota, K.; Wu, J.; Li, J.; Ye, T. Vehicle Mobility-Based Geographical Migration of Fog Resource for Satellite-Enabled Smart Cities. In Proceedings of the 2018 IEEE Global Communications Conference, GLOBECOM 2018, Abu Dhabi, UAE, 9–13 December 2018.
117. Varshney, P.; Simmhan, Y. Demystifying Fog Computing: Characterizing Architectures, Applications and Abstractions. In Proceedings of the 2017 IEEE 1st International Conference on Fog and Edge Computing, Madrid, Spain, 14–15 May 2017; pp. 116–124.
118. Al Ridhawi, I.; Kotb, Y.; Al Ridhawi, Y. Workflow-Net Based Service Composition Using Mobile Edge Nodes. *IEEE Access* **2017**, *5*, 23719–23735. [[CrossRef](#)]

**Publisher’s Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



© 2020 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 (<http://creativecommons.org/licenses/by/4.0/>).

Reproduced with permission of copyright owner. Further reproduction prohibited without permission.