A Survey on Trustworthiness for the Internet of Things

Article *in* IEEE Access · March 2021

DOI: 10.1109/ACCESS.2021.3066457

CITATIONS
19

READS
882

2 authors:

Franklin M. Ribeiro Junior
Instituto Federal do Maranhão (IFMA)
16 PUBLICATIONS **60** CITATIONS
SEE PROFILE

Carlos Alberto Kamienski
Universidade Federal do ABC (UFABC)
209 PUBLICATIONS **1,832** CITATIONS
SEE PROFILE

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 29, 2021, accepted March 9, 2021, date of publication March 17, 2021, date of current version March 23, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3066457

A Survey on Trustworthiness for the Internet of Things

FRANKLIN MAGALHÃES RIBEIRO JUNIOR[1,2], AND CARLOS ALBERTO KAMIENSKI[2], (Senior Member, IEEE)

[1]Federal Institute of Maranhão (IFMA), São Luís 65075-441, Brazil
[2]Federal University of ABC (UFABC), Santo André 09210-580, Brazil

Corresponding authors: Franklin Magalhães Ribeiro Junior (franklin.ribeirojunior@ifma.edu.br) and Carlos Alberto Kamienski (carlos.kamienski@ufabc.edu.br)

This work was supported in part by the MCTIC/RNP in Brazil, through the EU-Brazil Joint Call under Grant H2020-EUB-2017, and in part by the Federal Institute of Education, Science, and Technology of Maranhão (IFMA).

ABSTRACT IoT systems use sensors to collect data from smart environments and manage resources through data analysis. An IoT system deals with many connected devices with different network and hardware constraints in a real-world scenario. An IoT system needs to handle low-latency data analysis, security threats, internal vulnerabilities, and network disconnections, which cause data loss and incorrect decisions. Trustworthiness (also known as dependability) provides various features for an IoT end-to-end data flow, such as resilience, security, availability, reliability, scalability, maintainability, heterogeneity, hardware resources management, fault management policies, and data quality. This paper presents a survey on trustworthiness and dependability in IoT systems and proposes the Trustworthiness for IoT Framework (TW-IoT) to provide trustworthiness at the data level for mist and fog-based IoT systems. The TW-IoT framework provides data trustworthiness to ensure a continuous and uninterrupted operation of IoT data flow. We also discuss challenges and trade-offs related to data trustworthiness in IoT.

INDEX TERMS Dependability, fog computing, Internet of Things, mist computing, trustworthiness.

I. INTRODUCTION

The Internet of Things (IoT) is a networked system with billions of connected physical devices (sensors and actuators) transmitting and receiving data in a given context application [1]–[3]. IoT systems can monitor water distribution and irrigation in agriculture, reduces logistics costs, monitor patient health in a hospital, or optimize vehicle traffic through smart traffic lights [4]–[7]. Therefore, an IoT system grants services to smart scenarios in different contexts, efficiently managing hardware, software, and communication resources to reduce costs in specific domains.

In IoT, dealing with vulnerabilities of a vast number of heterogeneous and hardware-constrained devices is a challenge [4], [8]. Trustworthiness enhances IoT system features to handle several system challenges. However, there is no consensus in the literature about the definition of trustworthiness, as some research lines limit this concept to security scope handling only malicious attacks [9], [10].

The associate editor coordinating the review of this manuscript and approving it for publication was Eyhab Al-Masri.

Risks and threats for IoT systems involve malicious attacks by external agents and system threats such as faults, vulnerabilities, or unexpected system behaviors. Trustworthiness and dependability [11], [12] are similar concepts. They include requirements of system availability, reliability, scalability, maintainability, heterogeneity, data quality, hardware resources, security, agility in the response time, and system and network resilience [13].

A way to enforce trustworthiness requirements in IoT systems is using fog computing because it allows real-time data analysis at the edge [14]–[17]. Fog computing does not limit data analysis to a centralized cloud server. Fog enables local data analysis at the network edge [18], [19], reducing the network throughput and the need for data processing in a cloud [20], as well as allowing faster decision-making at the edge [19], [21]–[23]. However, even a fog-based IoT system has trustworthiness challenges, considering mechanisms to manage the IoT devices' data flow and the network, memory, and energy consumption constraints [8], [24], [25].

This paper presents a research overview of IoT trustworthiness to identify problems, challenges, approaches, solutions,

and technologies. We identify gaps and challenges about trustworthiness in IoT systems and also propose the *Trustworthiness for IoT Framework* (TW-IoT). Based on mist [26] and fog computing, TW-IoT provides trustworthiness from a data flow perspective throughout different IoT stages (thing, mist, fog, and cloud).

The main contributions of this paper are (i) to clarify concepts, characteristics, and gaps about trustworthiness in IoT systems, (ii) to propose the TW-IoT framework that contains a set of techniques and mechanisms for ensuring trustworthiness in the development of an IoT system based on mist and fog computing, (iii) to propose a data flow for each IoT stage (thing, mist, fog, and cloud), using the TW-IoT framework, (iv) to expose some trade-offs among trustworthiness mechanisms for the IoT data flow and finally, (v) to present the challenges related to trustworthiness in IoT systems.

The remainder of this paper is organized as follows: Section II defines a fog-based IoT system architecture, Section III contextualizes concepts of trustworthiness, dependability, and data trustworthiness, Section IV presents this paper adopted terminologies, and Section V describes the trustworthiness related studies. Section VI introduces the TW-IoT framework for data trustworthiness in IoT systems based on mist and fog computing. Section VII exemplifies TW-IoT mechanisms for IoT data flow stages (thing, mist, fog, and cloud), and Section VIII explains trade-offs between these mechanisms. Section IX discusses the lessons learned and challenges, and finally, Section X presents conclusions and future work.

II. THING-MIST-FOG-CLOUD IOT COMPUTING CONTINUUM

The Internet of Things (IoT) connects billions of embedded devices (sensors and actuators), transmitting and receiving data in a network [1], [27]. Through Internet, sensors send data to be processed and analyzed in a cloud server, whose results can generate instructions to actuators. Fig. 1 describes the underlying IoT Computing Continuum [28], providing the rationale for analyzing the trustworthiness concept, based on four processing stages: thing, mist, fog, and cloud.

A. THING STAGE

The Thing stage contains the physical devices: sensors and actuators. Sensors gather data from a specific environment, sending them to the Mist stage and beyond. After it, the IoT system uses data fusion or different models (e.g., machine/deep learning) according to a specific application [29]. As a result, the model generates a decision and sends it back to actuators in the Thing stage as commands to change the environment accordingly (e.g., turn on/ off some equipment).

B. MIST STAGE

The mist processing stage is closer to sensors [30], playing the role of a specific fog node deployed in the field, providing

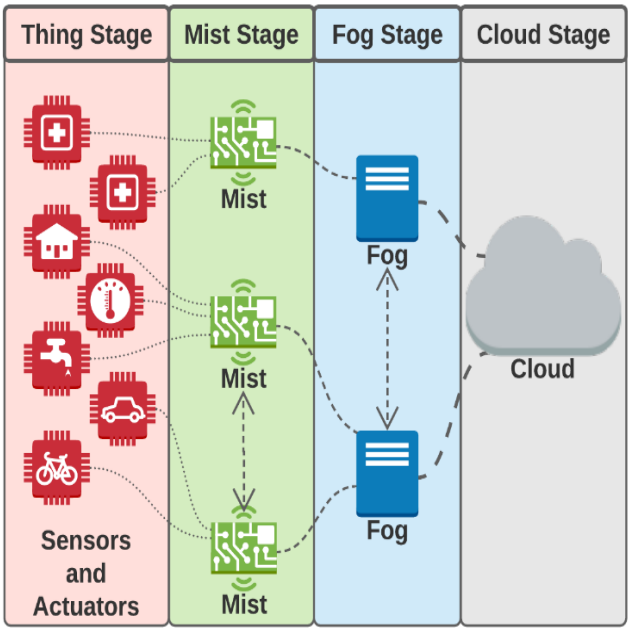

FIGURE 1. Thing-Mist-Fog-Cloud IoT computing continuum.

direct support for the communication of sensors and actuators with the Internet. Mist computing is a subset of fog computing, running on constrained resource equipment, such as single-board computers [31], [32]. A mist node significant function is behaving as a radio gateway for sensors and actuators, as defined by RFC 8376 [33]. Also, it can store, preprocess, and analyze data in a distributed fashion over multiple nodes. For that reason, one of the mist computing benefits is to improve the system scalability, as it may help increase the autonomy of devices closer to the edge [26].

C. FOG STAGE

Fog computing addresses new challenges related to the massive amount of data generated by the increasing use of IoT systems [16], [18], [20]. Fog computing supports a virtualized computing platform that offers processing, storage, and communication services between devices, users, and the cloud datacenter [34]. The main goals of fog computing are (i) decreasing latency for real-time services negatively affected by the long physical distance between devices and cloud data centers, (ii) enabling system load-balance at the edge and reducing processing in the cloud, and (iii) decreasing data traffic between the edge and the network core as the system does not need to send all data to the cloud.

Similarly to a cloud service, a fog node can store data and process models such as machine learning or data fusion algorithms to forecast certain behaviors or make decisions. However, there are significant differences between cloud and fog, such as the limited availability of computing resources, the security policies, and the hardware performance (memory and processing) [16], [35].

Fog is not a mandatory stage in IoT systems. However, its potential benefits may be worth it, such as reduced response

times at the edge and more robust network resilience [36]. Local data processing also prevents sending irrelevant data to the cloud [22] by analyzing sensor data locally [20].

D. CLOUD STAGE

Cloud computing has data processing high performance because it provides hardware resources in a scalable approach through cloud services virtualization [37]. A cloud service customer can execute multiple processes on a large scale since the cloud provides distributed hardware resources on demand for these processes [38]. The cloud plays a crucial role in any IoT system. Theoretically, the cloud stage is not needed because everything can be processed at an edge, e.g., in a farm office for smart agriculture. However, in practice, the cloud's resources, robustness, and reliability, either public or private, are unavoidable characteristics.

III. TRUSTWORTHINESS AND RELATED PARADIGMS

In this section, we explain the concepts of trustworthiness, dependability, and data trustworthiness.

A. TRUSTWORTHINESS AND DEPENDABILITY

Trustworthiness allows the uninterrupted continuity of system services [39]. In other words, a trustworthy (dependable) system should adapt and tolerate vulnerabilities throughout its life cycle [40]. The trustworthiness definition consists of system requirements concerning multiple aspects, such as security, resilience, availability, reliability, scalability, maintainability, heterogeneity, data quality, hardware resources, and fault management policies [13].

Some authors limit the trustworthiness concept as dealing with system security only [9], [41]. However, throughout this paper, trustworthiness and dependability are equivalent concepts [11], [12]. Trustworthiness is not only associated with failures in the system security level (by attacks) but also with general system vulnerabilities [13], [42]–[44]. Therefore, we consider the concepts of trustworthiness and dependability as synonyms, even though there is no consensus in the community [11], [12].

IoT critical systems need to deal with privacy, heterogeneity, and data analysis from billions of devices [45] in real-time, as well as system failures and real-time decision-making [45], [46]. Consequently, trustworthiness is essential in these IoT environments [47]–[50].

B. DATA TRUSTWORTHINESS

Trustworthiness is not limited only to enhance the system at the software/hardware level but also to improve the system at the data level [44], [51]. An IoT system that collects data in an environment must ensure the data trustworthiness, verifying if that data contains relevant information to the application domain.

Data trustworthiness needs to ensure data veracity, according to the IoT system context [52], [53]. For example, supposing the IoT system collects data from tropical weather. In that case, the IoT system must verify the data behavior, referring to these climate conditions. Otherwise, in future data analysis, the IoT system may not respond correctly, making wrong decisions, causing a vulnerability at the data level.

In an IoT system, ensuring data trustworthiness also means ensuring resilient data flow. Therefore, the data trustworthiness consists of maintaining data flow continuity and trusted system decisions [44], [53].

IV. ADOPTED TERMINOLOGIES

We adopt terminologies in this paper as the IoT system concepts, IoT smart service, IoT smart application, context, and a mechanism in our proposal called the integration of smart everything (explained in Section VII). We refer to the system context as circumstance or condition related to external or internal variations in an IoT system environment [17], [54], [55], as the weather temperature, energy consumption, soil moisture, or wind velocity measurement variations.

An IoT smart service contains smart applications. For example, a smart service refers to smart farming, smart mobility, or smart hospital service. At the same time, a smart application corresponds to applications of these smart services. Thus, a smart farming service contains smart applications like smart irrigation, smart water management, smart pest control, and a smart monitoring crop growth application.

The IoT system represents the entire hardware and software system, which contains IoT smart services and, consequently, IoT smart applications. An ideal IoT system has multiple smart services connected by the mechanism that we call smart everything.

V. TRUSTWORTHINESS RELATED STUDIES

We searched studies that explicitly use concepts and definitions of trustworthiness and dependability with the following combination of terms: dependability and IoT, trustworthiness and IoT, trustworthy IoT. We searched papers in IEEE Xplore, ACM Digital Library, Google Scholar, Science Direct, and Springer Link. We also examined the abstracts of the articles.

For the survey scope, we found a total of 57 studies related to dependability and trustworthiness. We perceived that 56 investigations refer to IoT, fog, or edge computing. We also observed that one study only focusing mainly on physical devices (sensors and actuators). We identified 24 papers referring only to trustworthiness, 27 to dependability, and six mentioned both terms (TABLE 1).

A. RELATED STUDIES CATEGORIES

We classify 57 studies related to dependability and trustworthiness: 18 present conceptual studies, 10 introduce a framework, 10 describe a mathematical investigation, eight present a performance analysis, six report algorithms or techniques, three show an architectural approach, and two focus on hardware related issues (Fig. 2).

1) CONCEPTUAL

Regarding studies that present definitions on dependability and trustworthiness, some papers explore the state of the art

FIGURE 2. Related studies categories.

and definitions [9], [42], [43], [56]–[63]. We also found surveys on trustworthiness [64]–[68] and dependability [69] and a systematic literature review (SLR) on dependability [70].

2) FRAMEWORK DESIGN

Some papers propose a framework for resource management in fog/cloud [71], other propose a framework for trust management for IoT devices [51], [72], [73], and fog nodes [74]. Also, some papers focus on security issues [10], [75]–[77] or fault recovery [78].

3) MATHEMATICAL APPROACH

Among research on IoT dependability, we found five studies that present mathematical approaches to measuring system reliability and availability [79]–[83]. In contrast, two others propose a calculus to estimate a confidence score to sensors [41], [84], and users [85]. We also found a proposal for an optimization approach for some dependability characteristics [86] and a theoretical approach based on Markov models to deal with vulnerabilities in a healthcare system [87].

4) PERFORMANCE ANALYSIS

Among the papers reporting a performance evaluation, some analyze trustworthiness by energy consumption [88]–[90], evaluate protocols [91], and communication interfaces [92]. Two papers compare different machine learning techniques for malicious data detection [93], [94], and another one compares machine learning techniques for data (image) reconstruction [95].

TABLE 1. Topics per paper.

Papers	Trustworthiness	Dependability	%
[9] [10] [41] [51] [56] [64-68] [72-74] [84] [85] [88] [89] [93] [94] [99-102] [104]	X		≈ 42.105
[39] [58-61] [63] [69-71] [76] [78-83] [86] [87] [90-92] [95-98] [103] [105]		X	≈ 47.368
[42] [43] [57] [62] [75] [77]	X	X	≈ 10.526

5) ALGORITHMS/TECHNIQUES

We identified papers that propose techniques or algorithms to improve dependability characteristics [39], [96]–[98], or techniques related to trust management mechanisms [99], [100].

6) ARCHITECTURAL APPROACH

We found three studies that propose architectural approaches: an architectural approach for a gateway that provides security to an IoT system [101]; a conceptual architecture for data provenance [102]; a standards-oriented approach [103].

7) HARDWARE DESIGN

We found two studies that propose solutions at the hardware level, dealing with system cryptography [104], and with limited computational resources in IoT focused on device memory constraints [105].

B. TRUSTWORTHINESS FEATURES

We consider the following set of features describes the trustworthiness concept: security (integrity, confidentiality, availability, and authenticity), system and network resilience, data quality (and semantic integrity), system availability, system reliability, scalability, maintainability, survivability, heterogeneity, IoT constraints (latency, memory, processing power, energy consumption), fault management and redundancy (see TABLE 2).

1) SECURITY

Data security is a feature for ensuring the IoT system trustworthiness [64], [88], [106]. A secure system needs to provide data integrity, authenticity, confidentiality to protect data from malicious attacks or not authorized access [107].

As shown in TABLE 2, we observed that security is the most mentioned feature of an IoT system's trustworthiness. The main security features that we found as follows:

- Integrity: refers to data content preservation, ensuring that data is not corrupted or altered by a malicious user or software. Integrity is strongly associated with IoT system trustworthiness because fraudulent data turns the system no longer reliable [64] since it makes wrong decisions [66].
- Authenticity: IoT data is authentic when it comes from a source that is a trusted part of the system [104], [107]. Unknown sensors can transmit non-authentic data with information that differs from the application context [97]. One way to deal with unreliable sensors in IoT is through the traceability of transmitted data [73] or by verifying the behavior of network sensors' energy consumption [84]. When there is a lack of data authenticity, integrity is not necessarily compromised, as the data is not corrupted or altered but fabricated. Therefore, a trustworthy IoT system must deal with mechanisms to verify data authenticity.
- Confidentiality: ensures that third parties have no access to IoT devices' data by packet interception [65]. Consequently, the IoT system must restrict data access via encryption mechanisms and access validation policies [64].
- Availability: from the security viewpoint, availability deals with interruptions [9], [104] caused by attacks such as denial of service (DoS), for example. However, the availability concept has a broader scope. A system can stop working even without any external attack but due to other vulnerabilities.

2) RESILIENCE

Resilience of a computer system is part of the trustworthiness concept [94]. It includes the ability to deal with system failures. Thus, a resilient system can prevent, tolerate, mitigate, remove, and predict failures [108], [109]. In a system with devices connected to a network, resilience is responsible for maintaining or recovering the communication service between devices, regardless of network failures [110], [111]. The concept of resilience is the system's ability to resist failures [112] using recoverability (survivability), adaptability, and the capacity to manage failures [13].

Network, hardware, or software vulnerabilities can interrupt a system service, so the system must react to these failures when they occur. However, there are computing costs to support system resilience [113]. The system requires resources to recover from a failure, to solve or mitigate a threat, reducing or compromising the system performance.

A resilient IoT system must deal with resource constraints (network, battery, memory, and processing capacity limitations) because IoT devices must react to failures as quickly as possible. After all, operations must occur in real-time [69].

A resilient fog-based IoT system data flow must prevent data losses caused by the network connection failure between mist/fog components. The fog also needs to deal with data loss due to low storage capacity in mist/fog nodes memory [111].

A relevant resilience challenge deals with network disconnection between components of different IoT stages, such as thing, mist, fog, or cloud [16], [114]–[121]. Therefore, it is necessary to establish mechanisms to ensure resilience

TABLE 2. Trustworthiness features in related work.

Paper Reference	Trustworthiness	Dependability	Resilience	Security	Data quality/ Semantic Integrity	Availability	Reliability	Scalability	Maintainability	Survivability	Heterogeneity	Quality of Service/ Bandwidth Constraints	Memory Capacity	Processing Power	Energy Consumption	Fault management	Redundancy
[9]	X			X		X	X										X
[10]	X			X			X										
[39]		X				X	X	X								X	X
[41]	X		X	X												X	
[42]	X	X	X	X		X	X			X							
[43]	X	X	X	X		X	X		X							X	
[51]	X			X	X		X										
[56]	X			X	X		X				X	X	X	X	X		
[57]	X	X	X	X		X	X		X								
[58]		X				X	X	X	X	X	X					X	X
[59]		X		X		X	X	X	X					X	X	X	
[60]		X	X	X		X	X	X			X	X	X	X		X	X
[61]		X		X		X	X	X	X	X						X	
[62]	X	X		X	X	X	X				X				X		
[63]		X		X		X	X	X	X		X	X	X	X		X	X
[64]	X			X		X	X										
[65]	X			X	X	X	X				X	X	X		X		
[66]	X			X		X	X										
[67]	X			X		X	X	X									
[68]	X		X	X	X	X	X	X			X	X	X	X	X		
[69]		X	X	X		X	X	X	X		X				X	X	X
[70]		X		X		X	X	X	X	X		X				X	X
[71]		X		X		X	X					X	X	X	X	X	X
[72]	X			X			X	X			X	X	X	X	X		
[73]	X			X			X									X	X
[74]	X			X		X	X	X				X					
[75]	X	X		X	X	X	X				X						
[76]		X		X		X	X		X		X					X	
[77]	X	X				X						X			X		
[78]		X		X		X	X	X	X							X	X
[79]		X				X	X	X				X				X	X
[80]		X		X		X	X				X					X	X
[81]		X				X	X	X	X		X						X
[82]		X				X	X			X					X	X	X
[83]		X				X	X	X				X				X	X
[84]	X			X	X	X					X				X		
[85]	X		X	X	X		X										
[86]		X		X		X	X			X	X	X				X	

TABLE 2. *(Continued.)* **Trustworthiness features in related work.**

[87]		X		X		X	X									X	X
[88]	X			X		X	X					X	X	X	X		
[89]	X			X		X	X					X			X	X	
[90]		X	X		X		X				X	X	X	X	X		
[91]		X				X	X					X			X		
[92]		X				X	X	X				X				X	X
[93]	X			X		X	X										
[94]	X		X	X		X	X	X			X						
[95]		X		X	X			X				X		X	X		
[96]		X	X	X		X	X									X	X
[97]		X		X	X	X	X				X					X	X
[98]		X		X		X	X	X				X			X	X	
[99]	X			X			X										
[100]	X					X	X					X					
[101]	X			X			X										
[102]	X			X			X						X				
[103]		X		X		X	X				X	X				X	
[104]	X			X		X	X										
[105]		X		X		X	X	X				X	X		X		

to fog-based IoT systems for data transmission [16], [122], such as fault management mechanisms for IoT nodes [16].

A resilient system must provide security and fault management mechanisms. The survivability concept offers the system's continuity through fault recovery techniques, which recovery techniques are one of the fault management features. Moreover, redundancy is a technique that the system uses to recover from a failure.

3) FAULT MANAGEMENT

In computer systems, a failure represents an unexpected behavior. In the IoT data domain, it is possible to exemplify some of these behaviors as data integrity loss due to semantic/syntactic vulnerabilities, packet loss due to network interference or connection loss between fog nodes [70], and data loss caused by network exhaustion or device memory overflow.

The literature categorizes fault management into four main aspects: fault detection, fault prediction, fault recovery (or mitigation), and fault prevention:

- Fault detection: fault detection in IoT is the verification of unexpected behavior using, for example, statistical analysis or machine learning methods [69]. IoT makes it possible to detect sensor failures, for example, by monitoring data values coming, such as the detection of outliers [97].
- Fault prediction: consists of predicting the fault occurrence. In IoT, the most common method for predicting a fault is through probabilistic models or data regression techniques.
- Fault recovery or mitigation: A way of recovering from a failure in IoT is, for example, using load balancing between nodes to mitigate service discontinuation [74]. Some studies also deal with IoT fault recovery by using redundancy techniques [58].
- Fault prevention: When the system predicts a fault (perceive the fault condition before it occurs), the system can perform mechanisms to prevent a failure (which refers to the system's inability to operate). Redundancy is an example of a mechanism for fault prevention by replicating data from a fog node to other fog nodes.

4) SURVIVABILITY

Survivability is part of the concept of resilience and represents a system's ability to survive attacks, failures, or degradation [13], [61]. In the fog computing scope, survivability maintains the data flow continuity between the IoT stages (from the thing stage until the cloud stage).

This feature uses some mechanisms to increase an IoT system's ability to survive vulnerabilities, such as data replication or fault management mechanisms. Also, fog-based IoT systems can rely on load balancing. In the fog node's imminent resource exhaustion, the fog can transfer its workload to other nodes [74].

An open issue related to survivability is data loss [16], which can occur by an eventual disconnection between IoT

stages, such as between fog and cloud. After a disconnection, when the connection returns, the data not sent to the cloud is not necessarily immediately transmitted, causing a data flow gap between fog and cloud. Therefore, IoT stages must store data temporarily during disconnections and transmit it to the next stage when the connection is active again.

Delay-tolerant networking (DTN) is a mechanism that deals with data loss due to network disconnections [123] and can be used together with fog-based IoT systems [124]–[128]. However, these proposals only focus on delay or packet loss, disregarding the solution impact on fog resources. Besides, the DTN design does not deal with IoT trustworthiness requirements.

5) REDUNDANCY

Redundancy increases IoT systems' reliability and availability [58] by replicating data from one node to other nodes. When, for example, a fog node A fails, the system continues to operate if other fog nodes B and C have a data copy from the failed fog node A.

Redundancy is not limited to providing system data replication. It also includes connection replication using different communication technology on the same IoT device [82], [83]. For example, in case of a connection failure in the LPWAN interface, the device may use a Wi-Fi or 4G connection to reestablish communication with other nodes [92].

6) DATA QUALITY/SEMANTIC INTEGRITY

Data quality is related to the IoT system context, and it directly impacts the solution decisions. The system context refers to a circumstance or condition related to external or internal variations in an IoT system environment [17], [54], [55], as the weather temperature, energy consumption, soil moisture, or wind velocity measurement variations.

Assuring data quality involves verifying whether the gathered data represents the actual system context [51], which means that semantic data integrity is essential in assessing data quality [129]. For example, in an environment where the climate is always humid, data with a dry climate's characteristics is likely wrong.

As the IoT system context directly impacts the actuators' actions, maintaining data quality ensures data trustworthiness [51]. The IoT system can verify data trust by the data type, system behavior, and context. Therefore, an IoT system previously needs to know the context and problem domain to provide trustworthy data analysis [129].

An IoT system needs to use meaningful data for decision-making [130], which means that part of the initial data processing must identify and remove data outliers. However, in cases of fake or manipulated data, it is necessary to observe whether the data comes from unreliable sensors [65], [95]. A data filter can solve this problem [97] since fake data can compromise future system actions.

7) SYSTEM AVAILABILITY

Availability ensures that the IoT system continues operating as long as possible. There are methods to keep the system available, like managing hardware redundancy [63], [82], and maintaining a fault management mechanism [58].

Some IoT applications are latency-sensitive and, consequently, require low latency for transmitting packets between IoT stages. Therefore, these systems are always available when it performs all actions in real-time [60].

System availability is a metric that indicates the operation system probability until it begins a failure state (Equation 1) [131]. Therefore, availability is given by the MTTF (mean time to failure) divided by MTTF + MTTR (mean time to repair).

$$Availability = \frac{MTTF}{MTTF + MTTR} \tag{1}$$

8) SYSTEM RELIABILITY

Reliability, as well as availability, is a relevant metric to ensure IoT trustworthiness. However, it is worth empathizing that both concepts differ since availability demands the whole system to be continuously available, while reliability refers to system operations' confidence [63]. Thus, reliability can be measured by the probability that a system will behave as expected for a specific time interval (t).

We measure the reliability by the MTBF (mean time between failures) metric if the system has some fault recovery mechanism. However, we measure the reliability by the MTTF (mean time to failure) if there is no recovery mechanism. For calculation, the first case we consider $\lambda = 1 /$ MTBF and for the second $\lambda = 1 /$ MTTF. Thus, we describe reliability by Equation (2), where R (t) is the reliability function of time t, and λ is the miss rate [132].

$$R(t) = e^{-\lambda t} \tag{2}$$

9) SCALABILITY

In a scenario with thousands or millions of sensors, IoT systems must satisfy the scalability demand to sustain a massive data transmission, storage, and analysis in real-time [15], [19]. A trustworthy system must be scalable because poor system performance and slow data analysis can result in delayed and, consequently, wrong decisions. Besides, with imprecise choices, the system harms data trustworthiness.

The scalability feature implies providing more computing resources to the IoT system [63]. In a fog-based IoT system, nodes can become computationally saturated because, in a real scenario, the fog is responsible for dealing with thousands of devices [60]. Therefore, the scalable deployment of fog computing systems requires the necessary hardware infrastructure.

The IoT system is more trustworthy with thousands of sensors because it stores more information, allowing it to make more precise decisions based on the gathered data. However, a system with many sensors demands additional security, network, and resilience requirements. Maintain a scalable and continuous data flow requires communication resilience between the IoT stages (mist-mist, mist-fog, fog-fog, and fog-cloud) [24], [122].

10) HETEROGENEITY

A real IoT system scenario comprises several heterogeneous communications and device technologies [27]. IoT communication protocols provide interaction between different devices because IoT devices have different hardware technologies and cannot understand each other. Multiple protocols differ in terms of packet formats and communication technologies, and not all protocols can support every IoT system [133]. Furthermore, dealing with heterogeneity is an essential trustworthiness issue.

LPWAN (low power wide area network) technologies provide long transmission ranges, low energy consumption, and low bandwidth. LPWAN is an attractive technology for IoT systems that send a few dozens of bytes every couple of minutes or hours [134], [135]. LoRaWAN, Sigfox, and NB-IoT are LPWAN technologies leading this front [27], [136].

There are other technologies for IoT system communications, as Bluetooth [133], Zigbee [137], and Z-Wave [138]. The IoT developer can also adopt IEEE 802.11 standards (Wi-Fi), transmitting over short-range distances by devices with no battery constraints [133].

IoT system trustworthiness solves the interconnection problems between heterogeneous devices. The massive number of highly heterogenous devices can increase communication faults in IoT infrastructures [69]. Hence, the system must support interoperability [76], allowing intercommunication between applications and devices for every connected node [20].

11) QUALITY OF SERVICE (QOS)

IoT systems, based on fog or not, must process data from thousands of sensors in real-time. Therefore, there is a concern with network latency in sending data from a stage to another since a long packet delay can impair critical decisions' agility and correctness. There is also a data loss problem caused by low-quality network connection [89] or caused by a disconnection between IoT stages.

Some IoT latency-sensitive systems do not accept packet delay or packet loss problems [50]. These circumstances only contribute to generating undesirable behavior, affecting the system's trustworthiness.

As previously mentioned, trustworthiness also deals with the choice of communication technologies. There may be interference between specific transmission protocol frequencies or range restrictions between devices [133], [136]. For example, for the Zigbee technology [137], the transmission distance range is lower than the LoRaWAN range [136].

12) MEMORY CAPACITY

Maintaining all system data stored is a way to ensure IoT data trustworthiness [56]. However, IoT devices frequently suffer from primary memory constraints. Keeping data in an IoT device for an indefinite time has a high cost. If we suppose a fog node that loses the cloud connection but continues to receive data from sensors and stores the data until the Internet connectivity returns, the storage memory can overflow. In this case, data fusion is a way to reduce these costs, joining consecutive stored data [105].

13) ENERGY CONSUMPTION

Most IoT devices have little autonomy due to limited battery [88], [59]. The impact of this constraint can reflect in (i) faults of data gathering whenever a sensor runs out of battery [89], (ii) storage gaps in a data time series, and (iii) wrong data analysis, since the gap in time series, harms the data analysis, and consequently untrusty decisions. Therefore, data trustworthiness needs to consider the device's energy consumption.

In smart farming, energy harvesting is a feasible solution for device battery constraints, such as photovoltaic cells [139]. However, this strategy does not work for all scenarios. For this reason, a relevant research challenge is reducing energy consumption in IoT devices [140], [141].

14) PROCESSING POWER

One critical aspect of IoT systems design and development is the limited devices processing power capacity [59]. Therefore, it is crucial to consider the processing constraints, especially for mist and fog nodes, since they are, to a great extent, responsible for ensuring trustworthiness.

In latency-sensitive IoT systems, some scenarios require real-time data processing [56], and the processing power directly impacts a system execution time (see Equation 3) [131]. Many IoT operations are related to fault detection, data redundancy, or load balancing. Therefore, IoT processing power is a concern to provide trustworthiness to IoT systems.

$$\text{CPU execution time for a program} = \frac{\text{CPU clock cycles for a program}}{\text{Clock rate}} \quad (3)$$

15) MAINTAINABILITY

Maintainability refers to a system's capacity to deal with changes [59], [69], including system evolution and adaptation. In a resilient computer system, adaptability suggests that the system can learn specific attack types to make the most appropriate decisions to protect against an attack [13].

C. V'S OF BIG DATA AND TRUSTWORTHINESS FEATURES IN IOT

Big data systems have several V features, such as variety, veracity, volume, velocity, and value [142], [143], which overlap some data trustworthiness characteristics for IoT. Variety deals with heterogeneity problems such as the treatment of data types. Veracity relates to data quality. Volume reflects the concern of analyzing information on a very large scale. Velocity assumes that computer systems must process and store data in real-time. Finally, value considers the data relevance for an application [142], [143].

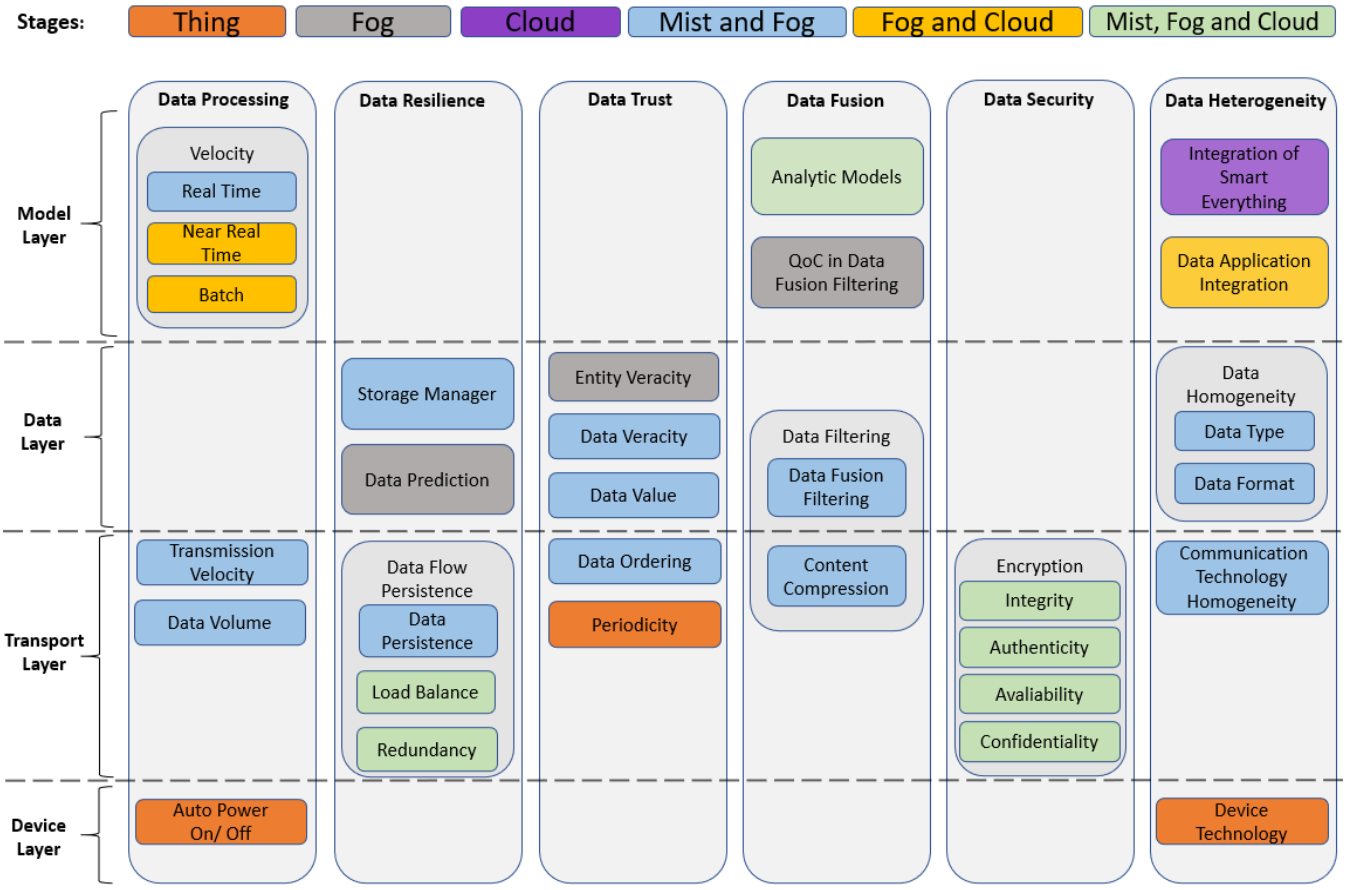

FIGURE 3. TW-IoT framework mechanisms.

VI. TW-IOT FRAMEWORK: A MIST-FOG-BASED FRAMEWORK FOR DATA TRUSTWORTHINESS TO THE INTERNET OF THINGS

In Section V, we presented the related studies on trustworthiness for IoT. However, there is still an unbridged gap in the literature: an architectural framework with concrete roles and mechanisms to ensure data trustworthiness in each stage of a mist-fog computing architecture.

With this challenge in mind, we propose the *Trustworthiness for IoT Framework* (TW-IoT) (Fig. 3), comprised of features, mechanisms, recommendations, and methods to ensure data flow continuity and data trustworthiness for mist-fog-based IoT systems. The proposed framework contains four stages (thing, mist, fog, and cloud), based on the *IoTinuum* architecture [28] (Section II) and four layers (device, transport, data, and model) based on the *IoTecture* architecture [28]:

- Device Layer: comprises different mechanisms to ensure data trustworthiness in the physical devices.
- Transport Layer: contains mechanisms responsible for packet security and trustworthiness in connection, communication, and data transmission over the network.
- Data Layer: contains mechanisms for ensuring data quality and dealing with meaningful information in the data, as data value and veracity.
- Model Layer: this layer is related to trustworthiness in data decision models and system data analysis.

We also propose six data trustworthiness design views: data processing, data resilience, data trust, data security, data heterogeneity, and data fusion (Fig. 3).

A. DATA RESILIENCE VIEW

The IoT system needs to ensure continuous and uninterrupted data flow through the IoT stages. Therefore, we design the data resilience view to endure vulnerabilities and data loss. This view includes data prediction, data flow persistence (data persistence mechanism, load balance, redundancy), and storage management mechanisms.

1) DATA PERSISTENCE MECHANISM

This mechanism focuses on ensuring data persistence when the network between the IoT stages disconnects. The data persistence mechanism improves communication between the

different IoT stages, considering typical limitations of mist and fog nodes.

The data persistence mechanism regularly performs health checks in network connections to avoid data losses between IoT stages. Simultaneously, mist and fog nodes store the data locally because they send the data later when the link returns, after a connection failure event.

The IoT nodes have constrained memory resources, so deploying together the data persistence mechanism and the data filter mechanism (subsection B.1) reduces the amount of data transmitted. This persistence mechanism may cause inconveniences for long disconnections without the filtering mechanism, such as mist memory overflow and long data transmission delays.

2) DATA PREDICTION

Data prediction is a mechanism that maintains data flow persistence when the data from a previous stage does not reach the destination stage. One of the reasons for data loss is the run out of the battery of the sensors or network disconnections. Without the complete time series, the system makes incorrect decisions.

Data prediction using machine learning algorithms can mitigate the data time series gaps and provide more reliable decisions. However, supposing that the system receives the lost data at the future moment, the system needs to replace the original data with the data created by prediction algorithms. In other words, it needs to compose new historical data.

3) STORAGE MANAGER

Billions of sensors can periodically collect and transmit data. The mist/fog stages are responsible for storing this massive amount of data, but these nodes have limited memory capacity. One way to solve this problem is by using data filtering (subsection B.1). The system needs to evaluate how long the mist/fog nodes should preserve old data, thus assessing the impact that deleting old data cause on the system's trustworthiness.

4) LOAD BALANCE

The load balancing mechanism prevents a mist/fog node from exhausting memory and processing resources. When a fog node reaches maximum hardware utilization in an IoT scenario, fog can alert mist nodes to redirecting part of future data to other fog nodes with time-driven sensors. In IoT scenarios with event-based sensors, an overloaded fog node can temporarily redirect data to another fog node.

5) REDUNDANCY

Redundancy is a way to ensure system continuity in the event of IoT node failures. Through redundancy, the system may find a new data path through other mist or fog nodes. A redundancy mechanism presumes that nodes always replicate data to other nodes whenever possible. Therefore, if a node loses connection or fails, the system can use another node to maintain the data flow.

B. DATA FUSION VIEW

This design view is responsible for implementing data filtering mechanisms in the mist and fog stages. It is also responsible for verifying and selecting data models used to make decisions.

1) DATA FILTERING

Data reduction (filtering) mechanisms optimize the data flow between mist, fog, and cloud [22], [144]–[148]. However, data filtering must deal with mist and fog resource constraints regarding the processing and storage capacity for a smart service (like smart farming) or with different smart applications of this service (like smart irrigation, smart water management, or smart crop growth monitoring).

Data filters are particularly relevant at the mist and fog stages, especially in long disconnections, given the need to store data temporarily until the connection returns. Filtered data occupies less memory space, thus avoiding memory overflow. Also, data can be transmitted faster when the connection returns, minimizing latency.

Filtering mechanisms can achieve significant data reduction while dealing with memory and latency constraints. In general terms, IoT data transmitted in short intervals can contain identical or very close values and classified through a data fusion model. This approach divides the data filtering mechanisms into two main methods: (i) the data fusion filtering method (by data sampling or classification [148]) and (ii) the data compression filtering method:

- Data fusion filtering method: fusion-based data filtering can adopt different techniques, considering smart applications and the computational capacity of IoT stages. The mist node has not the same computational power as the fog node. In other words, different stages need to apply different data fusion techniques for various applications. Therefore, it is necessary to evaluate the fusion filtering techniques concerning the constraints of each IoT stage.

  Data fusion can reduce the amount of data stored by statistical sampling or by data classification techniques. However, supposing that the sampling or classification techniques are not precise or accurate, the filtered data may lose details. Consequently, the system can make worse decisions.
- Data compression filtering method: allows the compression of data content and reduces data to a smaller size, thus lowering the resource requirements for transmission and storage. The data compression advantage is maintaining its details when uncompressed because it restores the original content. However, uncompressing generates computational costs because data always returns to the original size, demanding more resources from that particular IoT stage.

2) QUALITY OF CONTEXT (QOC) IN DATA FUSION FILTERING

The consequence of fusion-based filtering is data reduction, which requires the data quality to be maintained. Therefore,

one needs to understand how much reduction can be performed without jeopardizing future analysis in a given context. Ensuring the quality of the context in data fusion means refraining from excessively filtering. Therefore, IoT system designers can use a mechanism to verify the filtered data's quality based on the application context and decision models.

3) ANALYTIC MODELS

An IoT system needs to perform data gathering, monitoring, and analysis to extract information from a given context. Thus, based on environmental conditions or circumstances (context), the system makes decisions based on application rules (decision data models) for each specific smart application [54], [130], [149].

IoT systems can analyze data from individual IoT smart services (as smart farming) and store information from sensors of different smart applications (like smart irrigation, smart water management, or smart crop growth monitoring, for example). Thus, the system can analyze data from more than one smart application to enhance the decisions.

As the decision accuracy is directly related to the data model, this mechanism needs to automatically select a decision model according to the specific smart application.

C. DATA TRUST VIEW

The data trust view provides a data path without data syntactic and semantic integrity losses with meaningful information. This view deals with the features of data value, data veracity, entity veracity, data order, and data periodicity.

1) DATA VALUE

According to the smart application, the data value presumes that sensors must gather the necessary and relevant information for the IoT system. The lack of data or unnecessary excess of data can negatively impact future decisions. Therefore, the data value mechanism is responsible for checking the essential data to the IoT system based on the application and the data decision model.

For example, in a smart farming service, the data value mechanism works as follows: a sensor collects soil moisture measures from three different depths in a field. Supposing that the crop is in an early growth stage, the plant root size only reaches the first depth. Therefore, the IoT system can safely discard soil moisture measurements from deeper depths because they do not convey meaningful information.

2) DATA VERACITY

The data veracity mechanism must verify and discard outliers in a data set for a given context. It also perceives manipulated, corrupted, and fabricated data targeted by attacks or interference.

3) ENTITY VERACITY

The entity veracity is an extension of the data veracity concept to verify whether an entity, like a sensor or mist node, is transmitting unreliable data to fog. The veracity mechanism is responsible for identifies untrusted entities using a trust score. IoT designers can use algorithms to obtain this score [41], [51]. They can form the entity trust score by identifying the number of untrusted packets and their origin entity.

An untrusted entity transmits untrustworthy data caused by malicious attacks or internal hardware/software failures. Therefore, the IoT system needs to discard received data from that entity to ensure the data trustworthiness.

4) PERIODICITY

Data gathering frequency directly impacts the network traffic between the IoT stages and the accuracy of system decisions. The shorter the gathering interval, the greater the number of collected data and perceived details.

Some IoT systems have time-driven sensors gathering data by a fixed time interval. It is essential to find the ideal time interval for data gathering to obtain relevant decisions according to the smart application [134].

5) DATA ORDERING

After a network disconnection between the fog and the cloud, the fog nodes can send past and present data not ordered. Because of it, the IoT system needs to put data in order. An IoT application can analyze a data time series to make a decision. Sometimes the data order matters to guarantee trustworthy choices. Therefore, maintaining the chronological data order impacts the IoT system data trustworthiness.

D. DATA PROCESSING VIEW

A fog-based IoT system needs to execute instructions and handle the massive data transmission in batch, in real-time, or near real-time speed. The lambda architecture addresses the data volume and processing in real-time and batch [150]. However, this architecture works for high-performance systems. The IoT system designers can reuse concepts from this architecture and apply them to fog computing standards.

1) DATA VOLUME

A massive data volume requires an indispensable concern with the system's scalability. The IoT system must calculate and evaluate how much data each node can store and analyze, considering its hardware resource constraints. A node that reaches its hardware limits may use a load balancing mechanism to maintain the data flow under normal operating circumstances.

2) DATA TRANSMISSION VELOCITY

IoT latency-sensitive systems require real-time decisions, so mist/fog needs to analyze data in real-time. Data transfer delays may cause unexpected or delayed decisions. Consequently, the transmission velocity impacts data trustworthiness.

Developers of a mist-fog-based IoT system need to design the IoT stages considering the communication protocols,

the data flow volume, and hardware constraints used for each mist/fog node to increase the data transfer velocity.

3) DATA PROCESSING VELOCITY

The IoT system developers need to consider batch, near real-time, and real-time data processing techniques for different smart applications to ensure desirable speed performance.

4) AUTO POWER ON/OFF

The scope of the data processing view also deals with the execution of data gathering by sensors. The auto power-off mechanism works for IoT systems based on time-driven sensors (i.e., not event-driven sensors). This mechanism turns sensors off to consume less energy. It only turns the sensors on again during the next data gathering period.

This mechanism is feasible in many fog-based IoT systems, such as in the SWAMP project scenario [6], that sensors gather data periodically in 10-minute intervals. However, supposing that the gathering time interval is too short, IoT designers need to conduct a performance evaluation to verify the impact of energy consumption before deploying it.

E. DATA HETEROGENEITY VIEW

The heterogeneity view has mechanisms that deal with different communication protocols, data types, and data formats for various smart applications.

1) DEVICE TECHNOLOGY

In a scenario with billions of sensors from different hardware and software vendors, it is expected that the sensors use other measurement units and scales for the same type of data. For example, temperature sensors may gather data in various scales such as Celsius, Fahrenheit, or Kelvin. Therefore, this mechanism proposes to homogenize different data scales.

2) COMMUNICATION TECHNOLOGY HOMOGENEITY

There are several communication protocols, and each one better meets the demands of specific environments. Supposing that a fog node receives packets from different protocols, such as LoRa, Sigfox, and Zigbee, the fog node needs to understand the protocol formats and extract data from packet payloads.

The communication technology homogeneity mechanism interprets message patterns for different protocols. This mechanism can also convert several protocol messages into only one type of protocol message format.

3) DATA TYPE

The IoT system deals with different data types to make a decision. Smart healthcare data from a hospital, for example, may contain text, audio, video, and image. IoT system designers need to consider all data types, integrating it with the scale conversion mechanism, whether this mechanism did not previously deploy in the thing stage.

4) DATA FORMAT

In the IoT stages, it is necessary to deal with different data types and formats. For example, in smart city applications, images may have other formats such as JPEG, PNG, or BMP. Data values may also have different formats such as binary, octal, hexadecimal, or decimal. This mechanism needs to use a well-known technique to homogenize the data formats into just one, enabling the IoT stages to analyze it.

5) DATA APPLICATION INTEGRATION

A fog-based IoT system handles many smart applications that belong to the same smart service. A smart farming service, for example, can support different smart applications, such as precision irrigation, crop stock management, or pest control.

Considering that data gathering and decisions for different smart applications influence the whole system environment, this mechanism needs to combine data between different smart applications of the same smart service. Additionally, it needs to adequately analyze data by selecting and merging different data analytics models.

6) INTEGRATION OF SMART EVERYTHING

A complex and robust smart city environment includes smart homes, smart healthcare, and smart mobility. In this case, the system needs to manage and integrate different smart services to interconnect the entire system. Therefore, we suggest a mechanism that allows the system to make decisions based on integrated data analysis using the data knowledge of all IoT smart services.

In a hypothetical smart city system, an IoT home service detects a home accident. The home service transmits this information to a smart healthcare service that requests a smart mobility service to check the city's best route and send an ambulance. Therefore, in this case, the heterogeneity view is responsible for integrating all system services by a mechanism called the integration of smart everything. Section VII.D describes the proposal for the operation of this mechanism.

F. DATA SECURITY VIEW

Security is essential for maintaining data trustworthiness in an IoT system. A fog-based IoT system uses data encrypting for every IoT stage to ensure data confidentiality and integrity.

There are various data encryption and decryption algorithms that demand specific execution time and computational resources. The decision of the security policy needs to consider hardware constraints inherent to each IoT stage.

VII. FRAMEWORK DATA FLOW

In Section VI, we presented the TW-IoT framework, according to design views of data resilience, data security, data heterogeneity, data trust, data processing, and data fusion. The TW-IoT provides methods and mechanisms that satisfy data trustworthiness requirements, ensuring data flow continuity for mist and fog-based IoT systems.

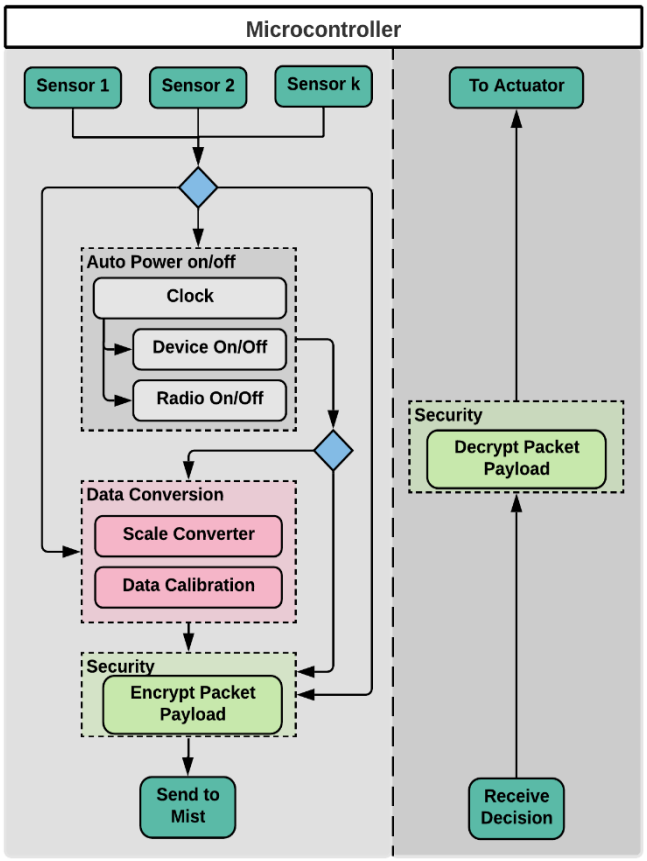

FIGURE 4. Thing data flow.

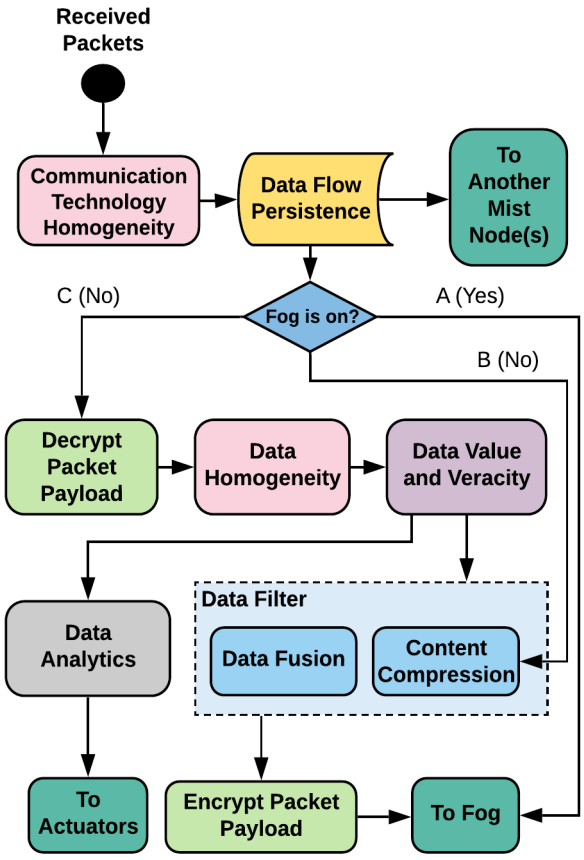

FIGURE 5. Mist data flow.

In this Section, we exemplify the TW-IoT framework data flow for our IoT stages. Our framework allows developers to choose the mechanisms to compose the data flow for each IoT stage.

A. THING DATA FLOW

We designed a data flow for a microcontroller with sensors and actuators in the thing stage (Fig. 4). For the sensor's data flow, a developer chooses to combine the framework mechanisms differently, using optional auto power on/off and data conversion mechanisms. In contrast, the packet encryption mechanism is obligatory for this data flow. Regarding actuators, the microcontroller receives a packet(s) and only needs to decrypt the packet payload to execute the system decision.

The thing stage can use mechanisms to turn the radio on/off, convert the scale value, and calibrate the collected data. The auto power on/off mechanisms save the device battery, and calibration is responsible for converting the collected data in meaningful information into the same scale values. The framework also allows data encryption deployment to encapsulate data into packets and transmit them to the mist. Depending on the data transmission technology, the packets are in LoRa, Sigfox, or Zigbee format.

B. MIST DATA FLOW

The mist stage uses mechanisms to deal with communication technology heterogeneity, data flow persistence, data encryption and decryption, data homogeneity, data value and veracity, data analytics, and data filtering techniques (Fig. 5).

The packets that arrive in the mist have different message formats, such as LoRa, Sigfox, or Zigbee format. Then, in the mist stage, the communication technology homogeneity mechanism is responsible for identifying the packet format and convert it into a single format.

After handling packet heterogeneity, mist uses the data flow persistence mechanism to store data, avoiding data loss even if the fog stage is disconnected. A mist node can send packets to other mist nodes to maintain the data redundancy, avoiding data loss if a mist node fails. Supposing the mist hardware resources (CPU and memory) are close to reaching the capacity limit, mist redirects part of the stored packets to another mist node.

When the IoT system is in normal operating conditions with an active connection between mist and fog, the mist node sends packets to fog by the data flow (A). When the fog connection is not active, the mist can choose two other paths, the first path (B) can compress data, and the second path (C) can analyze and filter data. Being the choice of data flows (B or C) a design option.

Assuming the data flow follows path B (Fig. 5), the mist filter mechanism compresses the packet content, generating a file as output. Then, the mist awaits the fog connection returns to sends the file. In this case, the mist does not need to decrypt the data, making the data flow more secure against attacks. In case of interception, the file only has compressed encrypted packets. However, when a fog node receives this file, it needs to uncompressing the data, causing an additional computational cost to the fog stage.

Assuming the data flow follows path C, the IoT system chooses to follow through the data filter, the data analysis, or both mechanisms but first following by packet payload decryption. Whether the system is handling LoRa packets, it uses LoRaWAN Network Server as ChirpStack [151] to decrypt each packet payload. After decryption, the IoT system converts data according to the data type and format. It checks the data's veracity and value and can follow two paths: data analysis or data filtering.

In data analysis, mist uses an analytic data model to make a decision. Data filtering occurs through two approaches (i) data compression and (ii) data fusion. Additionally, data fusion can use redundant data filtering based on statistical sampling or data characteristics. The choice of data filtering approaches and techniques is a design option.

In our data path, the mist analyzes the data and sends decisions to actuators until the fog connection returns. The mist also filters data during a disconnection. If the connection returns, the system encrypts the filtered data and sends it to fog. Consequently, when new packets arrive at the mist, the older packets do not delay the mist node to send new packets to fog. Then, the system data flow continues to operate uninterruptedly.

C. FOG DATA FLOW

The mist (Fig. 5) sends filtered data or raw data to fog (Fig. 6). Supposing the fog receives raw data from the mist. In that case, fog needs to handle this raw data by mechanisms like data homogeneity, data value, and data veracity. Consequently, the fog stage must have the same mechanisms deployed in the mist stage.

The mist stage transfers packets with raw data or filtered data to the fog stage. When the packets contain raw data, the fog needs to filter and handle data similarly to the mist, but with some differences, passing by more mechanisms through the data path.

In the data flow persistence mechanism, a fog node sends the data to other fog nodes to deal with redundancy and load balancing. Also, the fog node checks its cloud connection and the cloud transmission delay. Supposing the IoT system tolerates the cloud data transfer delay and that the communication channel between fog and cloud is active, the fog node transmits the data directly to the cloud. However, this delay is not acceptable for many IoT systems. In various smart applications, the IoT system must analyze data in real-time. In these situations, the fog node needs to analyze and filter data instead of the cloud.

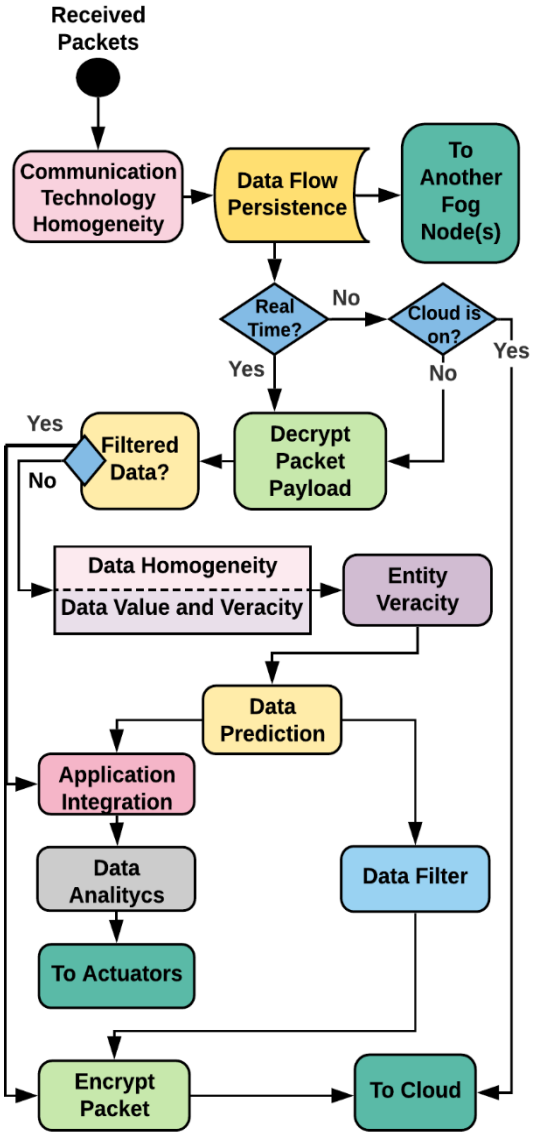

FIGURE 6. Fog data flow.

Upon receiving data, the fog node decrypts the packet payload, which can follow alternative paths: (i) if a packet contains filtered data, the fog node process it to make a decision, but (ii) if fog receives raw data, the flow passes through the data homogeneity step, i.e., through value mechanism (Section VI.C.1) and veracity mechanism (Section VI.C.2), and later on through the entity veracity mechanism (Fig. 6).

The entity veracity mechanism (Section VI.C.3) checks the entity ID, battery level, packet latency, outliers, and possible manipulated data to estimate if the packet comes from a trusted or untrusted entity. Assuming the fog node detects an untrusted packet, the fog node discards the packet. It can also discard all packets from that entity in the future. However, if the entity is trusted, fog stores the data. After these steps, the fog node can use the data prediction mechanism to fill missing data in times series if necessary.

The data flow takes two simultaneous paths in a row: (i) via data filtering, sending filtered data over the cloud,

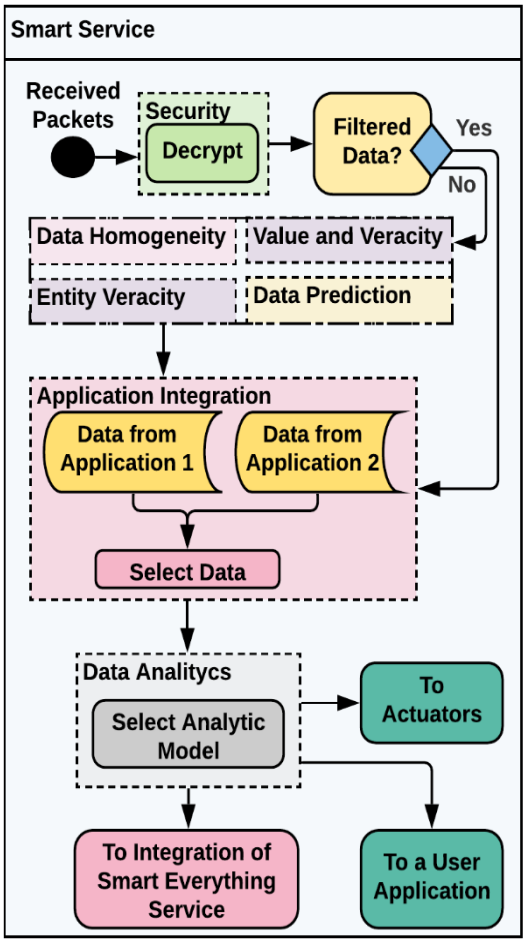

FIGURE 7. Cloud data flow.

and (ii) via data analysis mechanism, where fog chooses an analytic model to analyze data and make decisions. Finally, the fog sends decisions to actuators.

D. CLOUD DATA FLOW

Cloud (Fig. 7) decrypts the packet's payload to verify whether the fog filtered the data. If data is not filtered, the cloud follows data homogeneity, value, veracity, and prediction steps. Also, the cloud selects application analytic models before the data analysis. After processing the data analysis mechanism, the cloud sends data and decisions to the user IoT application, the system actuators, and the integration of smart everything service (Fig. 7), responsible for transmitting relevant data and decisions between services in different smart services (Fig. 8).

The cloud data flow includes an integration service for interconnecting different smart services, like a smart home, smart hospital (health care), and a smart mobility service (Fig. 8). In general, each service includes different smart applications. A smart farming service, for example, covers precision irrigation and pest control applications but not monitoring patient heartbeat, which is an application for smart hospital service.

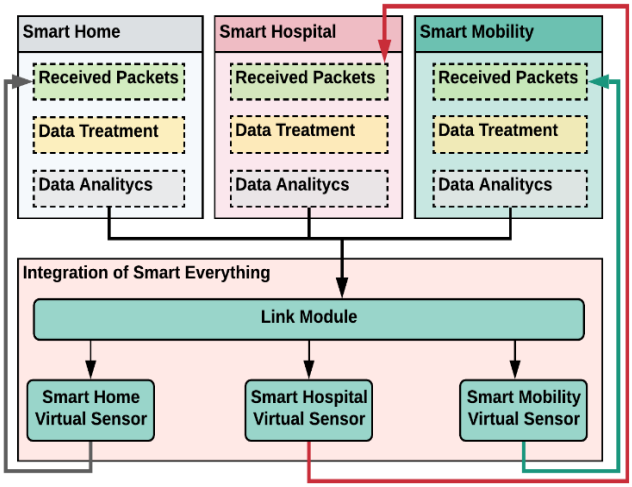

FIGURE 8. Cloud service integration data flow.

In Section VI.E.6, we mentioned the integration of the smart everything mechanism (Fig. 8). We discussed that a smart home service reports a domestic accident to a smart hospital service. Through a smart mobility service, the hospital verifies the best route for an ambulance to rescue the person who suffered the accident.

This integration mechanism contains a module called *Link Module* (Fig. 8), which receives data from every smart service in the system and only sends part of it to virtual sensors. It means that a smart service only receives data related to its smart applications through the virtual sensor.

The integration flow works in 3 steps: (i) cloud sends each smart service's data (and decisions) in the IoT system to the *Link Module*, (ii) this module sends only relevant data for each virtual sensor, and (iii) the virtual sensors send data to the smart services. Therefore, each smart service receives data from other smart services through a virtual sensor (Fig. 8).

Virtual sensors may send non-relevant data to a smart service. However, the TW-IoT framework can use the data value step to verify the data relevance for a specific smart service (Fig. 3 and Fig. 7).

VIII. TRADE-OFFS BETWEEN MIST AND FOG DATA FLOW

The TW-IoT framework mechanisms provide data trustworthiness in a mist-fog-based IoT system. Still, the IoT designers need to consider some trade-offs through the data flow in different IoT stages. The mist and fog data flow needs to consider, for example, computing costs for mechanisms related to data security and data filtering techniques.

A. DATA SECURITY AND DATA FILTERING IN MIST STAGE

Supposing the connection between mist and fog fails, the mist can follow two distinct flows (B or C): decrypt the packet payload and compress the encrypted packet content (Fig. 5). Filtering by data fusion requires decrypting data in the mist. However, decryption makes data more susceptible to malicious attackers.

An attacker can capture the decrypted data from the mist or discover the method used to encrypt data again in the mist. However, by using data filtering by data compression, the mist keeps the data encrypted without additional security service and does not expose data to external attackers. It has a cost for unpacking data in the next IoT stage and possibly a cost for data traffic latency between mist and fog.

B. ADVANTAGES AND DISADVANTAGES OF DATA FILTERING MECHANISMS

There are advantages and disadvantages to different data filtering mechanisms. The mist and fog filtering methods are (i) fusion filtering by statistical sampling, which deals with redundant data, (ii) fusion filtering by data classification, and (iii) filtering by data compression.

The filtering redundant data method reduces the data to a sample of data by statistical sampling. It means that IoT stages can reduce the amount of data, storage costs, and massive data traffic. However, it may cause data details loss and induces the IoT system to non-accurate decisions.

Filtering by data classification can categorize data by machine learning techniques and only transmits each category number to the next IoT stage [148]. It ensures a more significant reduction in the amount of data than the filtering redundant data method. The categories found by the classification should be strictly faithful to the analytic data model for the analysis. This method may lead to wrong decisions. Through this filtering, the loss of details is even more significant.

In data compression filtering, the original data remains intact, preventing possible equivocate system decisions. However, there is a higher cost for storing data, a network delay cost by sending the compressed file, and a computational cost to uncompressing the data in the next IoT stage.

C. DATA ANALYSIS AND CPU USAGE IN THE MIST STAGE

Mist devices have low computational power, but data analysis demands high CPU resources depending on the algorithm or data model. Performing data analysis in the mist stage is a design option. However, a long disconnection between the mist and fog stages can cause the system to wait a long time without analyzing and making decisions. It compromises the system's trustworthiness.

It is essential to emphasize that data analysis in the mist must happen during a network disconnection, but using light algorithms for demanding less processing resources. However, using light algorithms for data analysis may generate untrusted decisions. The IoT designer must balance the data analysis complexity power and the CPU usage, maintaining the decisions' accuracy.

D. DATA PREDICTION AND CPU USAGE IN FOG STAGE

The data prediction techniques generally use regression algorithms that demand high CPU usage. Therefore, the choice of prediction technique should consider the fog nodes' CPU processing power.

E. STORAGE TIME IN IOT STAGES

The IoT system deals with continuous and uninterrupted data flow from billions of sensors. However, the mist and fog nodes have constrained memory capacity. The mist and fog nodes need to keep the data (received by sensors) because they need to analyze data and maintain the data flow persistence.

The TW-IoT framework allows the IoT stages to store 3 data categories: raw data, refined data, and filtered data. According to the application, the IoT designers need to decide the memory usage percentage for each of these data categories and how long they remain stored in memory. They need to resolve it without compromising the data trustworthiness and future decisions.

IX. CHALLENGES

This section presents some challenges of data trustworthiness in a fog-based IoT system, identified throughout this paper.

A. CONNECTION REDUNDANCY

The IoT may use different communication technologies with diverse packet formats and specificities. Some systems deploy IoT nodes with two or more communication protocols [92].

A monitoring mechanism can prevent disconnections and low network performance by automatically switching the IoT node connection technology. In this way, each IoT node receives and transmits data over more than one communication protocol. A node can select a communication protocol with better network performance to transfer data. It is a challenge to deploy, in a real scenario, billions of sensors with connection redundancy mechanisms without increase the hardware costs.

B. DATA FILTER BY DATA KNOWLEDGE

Fog data reduction (filtering) aims to reduce the amount of data stored and transmitted by the mist and fog stages. These techniques can discard redundant data or data with no relevant variations [48], [147]. However, the fog needs to consider the analytic data model and the smart application for data filtering because the filtered data will represent the original data and generate trusted decisions [147].

A relevant challenge is to create a mechanism that precisely and accurately filters data without previously knowing the analytic data model or application context. In other words, the challenge is to design a context-independent data filtering mechanism only based on the data.

C. AGILE FAULT-RECOVERY

A fog-based latency-sensitive IoT system needs to execute fault recovery near real-time since the fog must provide data analysis in real-time. The velocity of fault recovery may impact system availability (Equation 1). It is challenging to offer mechanisms that recover the system of failures in real-time. Agility in fault recovery is a challenge.

D. AUTOMATIC REGISTRATION OF DEVICES

Secure IoT platforms require registering devices (including sensors) on the mist/fog servers. It allows the IoT system to recognize devices as trust devices and enhance the system trustworthiness. In a real scenario, the IoT system has to register billions of devices.

In an ideal scenario, a developer simply needs to connect new sensors to the network, and the system automatically recognizes them. However, new devices without a previous system notification may bring security risks since they can attack the IoT system. Therefore, it is challenging to create secure mechanisms that automatically register new devices to the fog node.

E. REAL-TIME ANALYSIS AND MIST/FOG RESOURCE CONSTRAINS

Ensuring data trustworthiness in a fog-based IoT system means deploying mist and fog nodes responsible for mechanisms that support heterogeneity, resilience, data trust, data filtering, security, and data analysis for billions of sensors. However, these features in the mist/fog nodes need to deal with constrained memory capacity and CPU power. It is challenging to find a balance between these mechanisms' computational costs and the mist/fog resource constraints. Another challenge is maintaining these mechanisms without changing the data analysis velocity, especially real-time data analysis.

X. CONCLUSION AND FUTURE WORK

Trustworthiness (dependability) handles requirements and characteristics that provide system availability, reliability, scalability, maintainability, heterogeneity, data quality, hardware resources, security, response time agility, and network resilience to computer systems. This survey presents state-of-the-art concepts about trustworthiness in fog-based IoT systems, summarizing and discussing literature.

We identified data trustworthiness gaps in dealing with the fog computing data flow. For that reason, we proposed the TW-IoT framework to ensure data trustworthiness for mist-fog-based IoT systems. The TW-IoT framework deploys mechanisms for a mist-fog-based IoT system to ensure trusted decisions and keep the data flow's uninterrupted continuity through all IoT system stages. Also, we identify data trustworthiness trade-offs and challenges for fog-based IoT systems.

As future work, we will evaluate the impact of each design view and mechanism of the TW-IoT framework for a real mist-fog-based IoT system, using pilots of the SWAMP project [6]. We intend to evaluate CPU and RAM usage for each IoT stage using our framework mechanisms. We aim to assess the data filtering mechanism combined with the data resilience view through the packet loss rate and network delay between IoT stages (thing, mist, fog, and cloud) in situations of network availability and unavailability. We plan to assess devices' energy consumption and end-to-end network delay, varying distinct data formats for various packet technologies types (using synthetic and real data) to evaluate the data veracity, value, and homogeneity mechanisms. We will also analyze the data processing view changing the communication channel frequency, channel error, number of sensors, and the packet transmission rate. Finally, we will evaluate the security view measuring energy, processing, and memory usage of different security algorithms, combining them with mechanisms of data resilience, data trust, and heterogeneity views.

REFERENCES

[1] L. Atzori, A. Lera, and G. Morabito, "The Internet of Things: A survey," *Comput. Netw.*, vol. 54, no. 15, pp. 2787–2805, Oct. 2010.

[2] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future Gener. Comput. Syst.*, vol. 29, no. 7, pp. 1645–1660, Sep. 2013.

[3] L. Da Xu, W. He, and S. Li, "Internet of Things in industries: A survey," *IEEE Trans. Ind. Informat.*, vol. 10, no. 4, pp. 2233–2243, Nov. 2014.

[4] E. Park, A. del Pobil, and S. Kwon, "The role of Internet of Things (IoT) in smart cities: Technology roadmap-oriented approaches," *Sustainability*, vol. 10, no. 5, p. 1388, May 2018.

[5] A. P. Plageras, K. E. Psannis, C. Stergiou, H. Wang, and B. B. Gupta, "Efficient IoT-based sensor BIG data collection–processing and analysis in smart buildings," *Future Gener. Comput. Syst.*, vol. 82, pp. 349–357, May 2018.

[6] C. Kamienski, J.-P. Soininen, M. Taumberger, R. Dantas, A. Toscano, T. S. Cinotti, R. F. Maia, and A. T. Neto, "Smart water management platform: IoT-based precision irrigation for agriculture," *Sensors*, vol. 19, no. 2, p. 276, Jan. 2019.

[7] J. Muangprathub, N. Boonnam, S. Kajornkasirat, N. Lekbangpong, A. Wanichsombat, and P. Nillaor, "IoT and agriculture data analysis for smart farm," *Comput. Electron. Agricult.*, vol. 156, pp. 467–474, Jan. 2019.

[8] P. P. Ray, "A survey on Internet of Things architectures," *J. King Saud Univ., Comput. Inf. Sci.*, vol. 30, no. 3, pp. 291–319, 2018.

[9] N. S. Labib, M. R. Brust, G. Danoy, and P. Bouvry, "Trustworthiness in IoT—A standards gap analysis on security, data protection and privacy," in *Proc. IEEE Conf. Standards Commun. Netw. (CSCN)*, Granada, Spain, Oct. 2019, pp. 1–7.

[10] S. Alam, M. M. R. Chowdhury, and J. Noll, "Interoperability of security-enabled Internet of Things," *Wireless Pers. Commun.*, vol. 61, no. 3, pp. 567–586, Dec. 2011.

[11] A. Avizienis, J.-C. Laprie, B. Randell, and C. Landwehr, "Basic concepts and taxonomy of dependable and secure computing," *IEEE Trans. Depend. Sec. Comput.*, vol. 1, no. 1, pp. 11–33, Jan./Mar. 2004.

[12] L. Bukowski, "System of systems dependability—Theoretical models and applications examples," *Rel. Eng. Syst. Saf.*, vol. 151, pp. 76–92, Jul. 2016.

[13] J.-H. Cho, S. Xu, P. M. Hurley, M. Mackay, T. Benjamin, and M. Beaumont, "STRAM: Measuring the trustworthiness of computer-based systems," *ACM Comput. Surv.*, vol. 51, no. 6, pp. 1–47, Feb. 2019.

[14] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the Internet of Things," in *Proc. 1st Ed. MCC Workshop Mobile Cloud Comput. (MCC)*. New York, NY, USA: Association for Computing Machinery, 2012, pp. 13–16.

[15] S. Verma, Y. Kawamoto, Z. M. Fadlullah, H. Nishiyama, and N. Kato, "A survey on network methodologies for real-time analytics of massive IoT data and open research issues," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1457–1477, 3rd Quart., 2017.

[16] A. Yousefpour, C. Fung, T. Nguyen, K. Kadiyala, F. Jalali, A. Niakanlahiji, J. Kong, and J. P. Jue, "All one needs to know about fog computing and related edge computing paradigms: A complete survey," *J. Syst. Archit.*, vol. 98, pp. 289–330, Sep. 2019.

[17] R. Mahmud, R. Kotagiri, and R. Buyya, "Fog computing: A taxonomy, survey and future directions," in *Internet of Everything* (Internet of Things: Technology, Communications and Computing), B. Di Martino, K. C. Li, L. Yang, and A. Esposito, Eds. Singapore: Springer, 2018.

[18] M. Aazam, S. Zeadally, and K. A. Harras, "Fog computing architecture, evaluation, and future research directions," *IEEE Commun. Mag.*, vol. 56, no. 5, pp. 46–52, May 2018.

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

[20] M. Chiang and T. Zhang, "Fog and IoT: An overview of research opportunities," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 854–864, Dec. 2016.

[21] A. Santamaria, P. Raimondo, M. Tropea, F. De Rango, and C. Aiello, "An IoT surveillance system based on a decentralised architecture," *Sensors*, vol. 19, no. 6, p. 1469, Mar. 2019.

[22] T. N. Gia, M. Jiang, A.-M. Rahmani, T. Westerlund, P. Liljeberg, and H. Tenhunen, "Fog computing in healthcare Internet of Things: A case study on ECG feature extraction," in *Proc. IEEE Int. Conf. Comput. Inf. Technol.; Ubiquitous Comput. Commun.; Dependable, Autonomic Secure Comput.; Pervasive Intell. Comput.*, Liverpool, U.K., Oct. 2015, pp. 356–363.

[23] J. Li, J. Jin, D. Yuan, M. Palaniswami, and K. Moessner, "EHOPES: Data-centered fog platform for smart living," in *Proc. Int. Telecommun. Netw. Appl. Conf. (ITNAC)*, Sydney, NSW, Australia, Nov. 2015, pp. 308–313, doi: 10.1109/ATNAC.2015.7366831.

[24] H. Atlam, R. Walters, and G. Wills, "Fog computing and the Internet of Things: A review," *Big Data Cogn. Comput.*, vol. 2, no. 2, p. 10, Apr. 2018.

[25] W. Masri, I. A. Ridhawi, N. Mostafa, and P. Pourghomi, "Minimizing delay in IoT systems through collaborative fog-to-fog (F2F) communication," in *Proc. 9th Int. Conf. Ubiquitous Future Netw. (ICUFN)*, Milan, Italy, Jul. 2017, pp. 1005–1010.

[26] J. S. Preden, K. Tammemäe, A. Jantsch, M. Leier, A. Riid, and E. Calis, "The benefits of self-awareness and attention in fog and mist computing," *Computer*, vol. 48, no. 7, pp. 37–45, Jul. 2015.

[27] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A survey on enabling technologies, protocols, and applications," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 2347–2376, 4th Quart., 2015.

[28] I. Zyrianoff, A. Heideker, D. Silva, J. Kleinschmidt, J.-P. Soininen, T. S. Cinotti, and C. Kamienski, "Architecting and deploying IoT smart applications: A performance–oriented approach," *Sensors*, vol. 20, no. 1, p. 84, Dec. 2019.

[29] P. Bellavista, J. Berrocal, A. Corradi, S. K. Das, L. Foschini, and A. Zanni, "A survey on fog computing for the Internet of Things," *Pervasive Mobile Comput.*, vol. 52, pp. 71–99, Jan. 2019.

[30] M. K. Yogi, K. C. Sekhar, and G. V. Kumar, "Mist computing: Principles, trends and future direction," *Int. J. Comput. Sci. Eng.*, vol. 4, no. 7, pp. 19–21, Jul. 2017.

[31] M. Asif-Ur-Rahman, F. Afsana, M. Mahmud, M. S. Kaiser, M. R. Ahmed, O. Kaiwartya, and A. James-Taylor, "Toward a heterogeneous mist, fog, and cloud-based framework for the Internet of healthcare things," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4049–4062, Jun. 2019.

[32] M. Linaje, J. Berrocal, and A. Galan-Benitez, "Mist and edge storage: Fair storage distribution in sensor networks," *IEEE Access*, vol. 7, pp. 123860–123876, 2019.

[33] S. Farrell, *Low-Power Wide Area Network (LPWAN) Overview*, document RFC 8376, Internet Engineering Task Force, 2018.

[34] M. Mukherjee, L. Shu, and D. Wang, "Survey of fog computing: Fundamental, network applications, and research challenges," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 3, pp. 1826–1857, 3rd Quart., 2018.

[35] M. R. Raza, A. Varol, and N. Varol, "Cloud and fog computing: A survey to the concept and challenges," in *Proc. 8th Int. Symp. Digit. Forensics Secur. (ISDFS)*, Beirut, Lebanon, Jun. 2020, pp. 1–6.

[36] A. Modarresi and J. P. G. Sterbenz, "Toward resilient networks with fog computing," in *Proc. 9th Int. Workshop Resilient Netw. Design Modeling (RNDM)*, Alghero, Italy, Sep. 2017, pp. 1–7.

[37] M. Armbrust, I. Stoica, M. Zaharia, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, and A. Rabkin, "A view of cloud computing," *Commun. ACM*, vol. 53, no. 4, pp. 50–58, 2010.

[38] A. Botta, W. de Donato, V. Persico, and A. Pescapé, "Integration of cloud computing and Internet of Things: A survey," *Future Generat. Comput. Syst.*, vol. 56, pp. 684–700, Mar. 2016.

[39] L. M. C. E. Martins, F. L. de Caldas Filho, R. T. de Sousa Júnior, W. F. Giozza, and J. P. C. L. da Costa, "Increasing the dependability of IoT middleware with cloud computing and microservices," in *Proc. 10th Int. Conf. Utility Cloud Comput. (UCC Companion)*. New York, NY, USA: Association for Computing Machinery, Dec. 2017, pp. 203–208.

[40] D. Jackson, "A direct path to dependable software," *Commun. ACM*, vol. 52, no. 4, pp. 78–88, Apr. 2009.

[41] M. Culler and K. Davis, "Toward a sensor trustworthiness measure for grid-connected IoT-enabled smart cities," in *Proc. IEEE Green Technol. Conf. (GreenTech)*, Austin, TX, USA, Apr. 2018, pp. 168–171.

[42] K. Periyasamy, V. Alagar, and K. Wan, "Dependable design for elderly health care," in *Proc. Federated Conf. Comput. Sci. Inf. Syst. (FedCSIS)*, Prague, Czech Republic, Sep. 2017, pp. 803–806.

[43] F. Coallier, "A system of systems engineering perspective on IoT trustworthiness," in *Proc. 13th Annu. Conf. Syst. Syst. Eng. (SoSE)*, Paris, France, Jun. 2018, pp. 89–91.

[44] H. U. Rahman, G. Wang, M. Z. A. Bhuiyan, and J. Chen, "Trustworthy data collection for cyber systems: A taxonomy and future directions," in *Smart City and Informatization* (Communications in Computer and Information Science), vol. 1122, G. Wang, A. El Saddik, X. Lai, G. M. Perez, and K. K. Choo, Eds. Singapore: Springer, 2019.

[45] G. Javadzadeh and A. M. Rahmani, "Fog computing applications in smart cities: A systematic survey," *Wireless Netw.*, vol. 26, no. 2, pp. 1433–1457, Feb. 2020.

[46] F. A. Kraemer, A. E. Braten, N. Tamkittikhun, and D. Palma, "Fog computing in healthcare—A review and discussion," *IEEE Access*, vol. 5, pp. 9206–9222, 2017.

[47] C. Kotronis, G. Minou, G. Dimitrakopoulos, M. Nikolaidou, D. Anagnostopoulos, A. Amira, F. Bensaali, H. Baali, and H. Djelouat, "Managing criticalities of e-Health IoT systems," in *Proc. IEEE 17th Int. Conf. Ubiquitous Wireless Broadband (ICUWB)*, Salamanca, Spain, Sep. 2017, pp. 1–5.

[48] I. Azimi, A. Anzanpour, A. M. Rahmani, T. Pahikkala, M. Levorato, P. Liljeberg, and N. Dutt, "HiCH: Hierarchical fog-assisted computing architecture for healthcare IoT," *ACM Trans. Embedded Comput. Syst.*, vol. 16, no. 5s, pp. 1–20, Oct. 2017.

[49] C. Kamienski, M. Jentsch, M. Eisenhauer, J. Kiljander, E. Ferrera, P. Rosengren, J. Thestrup, E. Souto, W. S. Andrade, and D. Sadok, "Application development for the Internet of Things: A context-aware mixed criticality systems development platform," *Comput. Commun.*, vol. 104, pp. 1–16, May 2017.

[50] B. Farahani, F. Firouzi, V. Chang, M. Badaroglu, N. Constant, and K. Mankodiya, "Towards fog-driven IoT eHealth: Promises and challenges of IoT in medicine and healthcare," *Future Gener. Comput. Syst.*, vol. 78, pp. 659–676, Jan. 2018.

[51] E. Bertino, "Data trustworthiness—Approaches and research challenges," in *Data Privacy Management, Autonomous Spontaneous Security, and Security Assurance* (Lecture Notes in Computer Science), vol. 8872, J. Garcia-Alfaro *et al.*, Eds. Cham, Switzerland: Springer, 2015.

[52] C. Dai, D. Lin, E. Bertino, and M. Kantarcioglu, "An approach to evaluate data trustworthiness based on data provenance," in *Secure Data Management* (Lecture Notes in Computer Science), vol. 5159, W. Jonker and M. Petković, Eds. Berlin, Germany: Springer, 2008.

[53] F. Azzedin and M. Ghaleb, "Internet-of-Things and information fusion: Trust perspective survey," *Sensors*, vol. 19, no. 8, p. 1929, Apr. 2019.

[54] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Context aware computing for the Internet of Things: A survey," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 1, pp. 414–454, 1st Quart., 2014.

[55] R. Togneri, G. Camponogara, J.-P. Soininen, and C. Kamienski, "Foundations of data quality assurance for IoT-based smart applications," in *Proc. IEEE Latin-Amer. Conf. Commun. (LATINCOM)*, Salvador, Brazil, Nov. 2019, pp. 1–6.

[56] N. Haron, J. Jaafar, I. A. Aziz, M. H. Hassan, and M. I. Shapiai, "Data trustworthiness in Internet of Things: A taxonomy and future directions," in *Proc. IEEE Conf. Big Data Anal. (ICBDA)*, Kuching, Malaysia, Nov. 2017, pp. 25–30.

[57] R. Rogers, E. Apeh, and C. J. Richardson, "Resilience of the Internet of Things (IoT) from an information assurance (IA) perspective," in *Proc. 10th Int. Conf. Softw., Knowl., Inf. Manage. Appl. (SKIMA)*, Chengdu, China, 2016, pp. 110–115.

[58] Z. Bakhshi and G. Rodriguez-Navas, "A preliminary roadmap for dependability research in fog computing," *SIGBED Rev.*, vol. 16, no. 4, pp. 14–19, 2020.

[59] T. Fruhwirth, L. Krammer, and W. Kastner, "Dependability demands and state of the art in the Internet of Things," in *Proc. IEEE 20th Conf. Emerg. Technol. Factory Automat. (ETFA)*, Luxembourg, Europe, Sep. 2015, pp. 1–4.

[60] S. Bagchi, M.-B. Siddiqui, P. Wood, and H. Zhang, "Dependability in edge computing," *Commun. ACM*, vol. 63, no. 1, pp. 58–66, Jan. 2020.

[61] E. Ojie and E. Pereira, "Exploring dependability issues in IoT applications," in *Proc. 2nd Int. Conf. Internet Things, Data Cloud Comput. (ICC)*. New York, NY, USA: Association for Computing Machinery, Mar. 2017, pp. 1–5.

[62] D. Korzun, A. Varfolomeyev, A. Shabaev, and V. Kuznetsov, "On dependability of smart applications within edge-centric and fog computing paradigms," in *Proc. IEEE 9th Int. Conf. Dependable Syst., Services Technol. (DESSERT)*, Kiev, Ukraine, May 2018, pp. 502–507.

[63] K. Iwanicki, "A distributed systems perspective on industrial IoT," in *Proc. IEEE 38th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Vienna, Austria, Jul. 2018, pp. 1164–1170.

[64] F. Zafar, A. Khan, S. Suhail, I. Ahmed, K. Hameed, H. M. Khan, F. Jabeen, and A. Anjum, "Trustworthy data: A survey, taxonomy and future trends of secure provenance schemes," *J. Netw. Comput. Appl.*, vol. 94, pp. 50–68, Sep. 2017.

[65] R. Hu, Z. Yan, W. Ding, and L. T. Yang, "A survey on data provenance in IoT," *World Wide Web*, vol. 23, pp. 1441–1463, Mar. 2020.

[66] S. Hammoudi, Z. Aliouat, and S. Harous, "Challenges and research directions for Internet of Things," *Telecommun. Syst.*, vol. 67, no. 2, pp. 367–385, Feb. 2018.

[67] M. Zhao, C. Hu, X. Song, and C. Zhao, "Towards dependable and trustworthy outsourced computing: A comprehensive survey and tutorial," *J. Netw. Comput. Appl.*, vol. 131, pp. 55–65, Apr. 2019.

[68] A. Sharma, E. S. Pilli, A. P. Mazumdar, and P. Gera, "Towards trustworthy Internet of Things: A survey on trust management applications and schemes," *Comput. Commun.*, vol. 160, pp. 475–493, Jul. 2020.

[69] D. Ratasich, F. Khalid, F. Geissler, R. Grosu, M. Shafique, and E. Bartocci, "A roadmap toward the resilient Internet of Things for cyber-physical systems," *IEEE Access*, vol. 7, pp. 13260–13283, 2019.

[70] Z. Bakhshi, G. Rodriguez-Navas, and H. Hansson, "Dependable fog computing: A systematic literature review," in *Proc. 45th Euromicro Conf. Softw. Eng. Adv. Appl. (SEAA)*, Kallithea-Chalkidiki, Greece, Aug. 2019, pp. 395–403.

[71] L. Caviglione and M. Gaggero, "Multiobjective placement for secure and dependable smart industrial environments," *IEEE Trans. Ind. Informat.*, vol. 17, no. 2, pp. 1298–1306, Feb. 2021.

[72] A. Sharma, E. S. Pilli, A. P. Mazumdar, and M. C. Govil, "A framework to manage trust in Internet of Things," in *Proc. Int. Conf. Emerg. Trends Commun. Technol. (ETCT)*, Dehradun, India, Nov. 2016, pp. 1–5.

[73] Z. Lv, Y. Han, A. K. Singh, G. Manogaran, and H. Lv, "Trustworthiness in industrial IoT systems based on artificial intelligence," *IEEE Trans. Ind. Informat.*, vol. 17, no. 2, pp. 1496–1504, Feb. 2021.

[74] F. H. Rahman, T.-W. Au, S. H. S. Newaz, W. S. Suhaili, and G. M. Lee, "Find my trustworthy fogs: A fuzzy-based trust evaluation framework," *Future Gener. Comput. Syst.*, vol. 109, pp. 562–572, Aug. 2020.

[75] K. Wan and V. Alagar, "Integrating context-awareness and trustworthiness in IoT descriptions," in *Proc. IEEE Int. Conf. Green Comput. Commun. IEEE Internet Things IEEE Cyber, Phys. Social Comput.*, Beijing, China, Aug. 2013, pp. 1168–1174.

[76] O. Soultatos, M. Papoutsakis, K. Fysarakis, G. Hatzivasilis, M. Michalodimitrakis, G. Spanoudakis, and S. Ioannidis, "Pattern-driven security, privacy, dependability and interoperability management of IoT environments," in *Proc. IEEE 24th Int. Workshop Comput. Aided Modeling Design Commun. Links Netw. (CAMAD)*, Limassol, Cyprus, Sep. 2019, pp. 1–6.

[77] V. S. Dasari, M. Pouryazdan, and B. Kantarci, "Selective versus non-selective acquisition of crowd-solicited IoT data and its dependability," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, Kansas City, MO, USA, May 2018, pp. 1–6.

[78] K. S. Dar, A. Taherkordi, and F. Eliassen, "Enhancing dependability of cloud-based IoT services through virtualization," in *Proc. IEEE 1st Int. Conf. Internet-of-Things Design Implement. (IoTDI)*, Berlin, Germany, Apr. 2016, pp. 106–116.

[79] D. Macedo, L. A. Guedes, and I. Silva, "A dependability evaluation for Internet of Things incorporating redundancy aspects," in *Proc. 11th IEEE Int. Conf. Netw., Sens. Control*, Miami, FL, USA, Apr. 2014, pp. 417–422.

[80] V.-K. Stefan, P. Otto, and P. M. Alexandrina, "Considerations regarding the dependability of Internet of Things," in *Proc. 14th Int. Conf. Eng. Mod. Electr. Syst. (EMES)*, Oradea, Romania, Jun. 2017, pp. 145–148.

[81] E. Araujo, J. Dantas, R. Matos, P. Pereira, and P. Maciel, "Dependability evaluation of an IoT system: A hierarchical modelling approach," in *Proc. IEEE Int. Conf. Syst., Man Cybern. (SMC)*, Bari, Italy, Oct. 2019, pp. 2121–2126.

[82] E. Andrade and B. Nogueira, "Dependability evaluation of a disaster recovery solution for IoT infrastructures," *J. Supercomput.*, vol. 76, no. 3, pp. 1828–1849, Mar. 2020.

[83] P. Vedavalli and C. Deepak, "Enhancing reliability and fault tolerance in IoT," in *Proc. Int. Conf. Artif. Intell. Signal Process. (AISP)*, Amaravati, India, Jan. 2020, pp. 1–6.

[84] P. Dass, S. Misra, and C. Roy, "T-safe: Trustworthy service provisioning for IoT-based intelligent transport systems," *IEEE Trans. Veh. Technol.*, vol. 69, no. 9, pp. 9509–9517, Sep. 2020.

[85] B. Shayesteh, V. Hakami, and A. Akbari, "A trust management scheme for IoT-enabled environmental health/accessibility monitoring services," *Int. J. Inf. Secur.*, vol. 19, no. 1, pp. 93–110, Feb. 2020.

[86] R. Zheng, C. Wang, T. Zhang, M. Zhang, Q. Wu, and W. Wei, "A dependability optimization approach for high layers in IoT," *Int. J. Simul., Syst., Sci. Technol.*, vol. 17, no. 45, pp. 1–4, 2016.

[87] A. Strielkina, V. Kharchenko, and D. Uzun, "Availability models for healthcare IoT systems: Classification and research considering attacks on vulnerabilities," in *Proc. IEEE 9th Int. Conf. Dependable Syst., Services Technol. (DESSERT)*, Kiev, Ukraine, May 2018, pp. 58–62.

[88] S. Suhail, C. S. Hong, M. A. Lodhi, F. Zafar, A. Khan, and F. Bashir, "Data trustworthiness in IoT," in *Proc. Int. Conf. Inf. Netw. (ICOIN)*, Chiang Mai, Thailand, Jan. 2018, pp. 414–419.

[89] N. Karthik and V. S. Ananthanarayana, "Sensor data modeling for data trustworthiness," in *Proc. IEEE Trustcom/BigDataSE/ICESS*, Sydney, NSW, Australia, Aug. 2017, pp. 909–916.

[90] J. Henkel, S. Pagani, H. Amrouch, L. Bauer, and F. Samie, "Ultra-low power and dependability for IoT devices (invited paper for IoT technologies)," in *Proc. Design, Automat. Test Eur. Conf. Exhib. (DATE)*, Lausanne, Switzerland, Mar. 2017, pp. 954–959.

[91] M. Schuß, C. A. Boano, M. Weber, and K. Römer, "A competition to push the dependability of low-power wireless protocols to the edge," in *Proc. Int. Conf. Embedded Wireless Syst. Netw. (EWSN)*. New York, NY, USA: Junction Publishing, 2017, pp. 54–65.

[92] G. Signoretti, M. Silva, J. Araujo, I. Silva, D. Silva, P. Ferrari, and E. Sisinni, "A dependability evaluation for OBD-II edge devices: An Internet of intelligent vehicles perspective," in *Proc. 9th Latin-Amer. Symp. Dependable Comput. (LADC)*, Natal, Brazil, Nov. 2019, pp. 1–9.

[93] N. Banerjee, T. Giannetsos, E. Panaousis, and C. C. Took, "Unsupervised learning for trustworthy IoT," in *Proc. IEEE Int. Conf. Fuzzy Syst. (FUZZ-IEEE)*, Rio de Janeiro, Brazil, Jul. 2018, pp. 1–8.

[94] M. M. Hassan, A. Gumaei, S. Huda, and A. Almogren, "Increasing the trustworthiness in the industrial IoT networks through a reliable cyberattack detection model," *IEEE Trans. Ind. Informat.*, vol. 16, no. 9, pp. 6154–6162, Sep. 2020.

[95] H. Qiu, Q. Zheng, T. Zhang, M. Qiu, G. Memmi, and J. Lu, "Toward secure and efficient deep learning inference in dependable IoT systems," *IEEE Internet Things J.*, vol. 8, no. 5, pp. 3180–3188, Mar. 2021.

[96] L. Russell, F. Kwamena, and R. Goubran, "Towards reliable IoT: Fog-based AI sensor validation," in *Proc. IEEE Cloud Summit*, Washington, DC, USA, Aug. 2019, pp. 37–44.

[97] I. Belkacem, S. Nait-Bahloul, and D. Sauveron, "Enhancing dependability through profiling in the collaborative Internet of Things," *Multimedia Tools Appl.*, vol. 78, no. 3, pp. 2983–3007, Feb. 2019.

[98] B. Großwindhager, A. Rupp, M. Tappler, M. Tranninger, S. Weiser, B. K. Aichernig, C. A. Boano, M. Horn, G. Kubin, S. Mangard, M. Steinberger, and K. Römer, "Dependable Internet of Things for networked cars," *Int. J. Comput.*, vol. 16, no. 4, pp. 226–237, Dec. 2017.

[99] K. A. Awan, I. U. Din, A. Almogren, M. Guizani, A. Altameem, and S. U. Jadoon, "RobustTrust—A pro-privacy robust distributed trust management mechanism for Internet of Things," *IEEE Access*, vol. 7, pp. 62095–62106, 2019.

[100] F. H. Rahman, T. W. Au, S. H. S. Newaz, and W. S. Suhaili, "Trustworthiness in fog: A fuzzy approach," in *Proc. VI Int. Conf. Netw., Commun. Comput. (ICNCC)*. New York, NY, USA: Association for Computing Machinery, 2017, pp. 207–211.

[101] E. Kim and C. Keum, "Trustworthy gateway system providing IoT trust domain of smart home," in *Proc. 9th Int. Conf. Ubiquitous Future Netw. (ICUFN)*, Milan, Italy, Jul. 2017, pp. 551–553.

[102] M. Elkhodr, B. Alsinglawi, and M. Alshehri, "Data provenance in the Internet of Things," in *Proc. 32nd Int. Conf. Adv. Inf. Netw. Appl. Workshops (WAINA)*, Kraków, Poland, May 2018, pp. 727–731.

[103] K. Fysarakis, M. Papoutsakis, N. Petroulakis, and G. Spanoudakis, "Towards IoT orchestrations with security, privacy, dependability and interoperability guarantees," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Waikoloa, HI, USA, Dec. 2019, pp. 1–6.

[104] T. Ulz, T. Pieber, C. Steger, R. Matischek, and H. Bock, "Towards trustworthy data in networked control systems: A hardware-based approach," in *Proc. 22nd IEEE Int. Conf. Emerg. Technol. Factory Automat. (ETFA)*, Limassol, Cyprus, Sep. 2017, pp. 1–8.

[105] D. Silva, M. Nogueira, M. Rodrigues, I. Silva, P. Ferrari, and E. Sisinni, "Implementation of a dependable smart device in IoT era," in *Proc. 9th Latin-Amer. Symp. Dependable Comput. (LADC)*, Natal, Brazil, Nov. 2019, pp. 1–6.

[106] F. F. Borelli, G. O. Biondi, and C. A. Kamienski, "BIoTA: A buildout IoT application language," *IEEE Access*, vol. 8, pp. 126443–126459, 2020.

[107] S. Vashi, J. Ram, J. Modi, S. Verma, and C. Prakash, "Internet of Things (IoT): A vision, architectural elements, and security issues," in *Proc. Int. Conf. I-SMAC, IoT Social, Mobile, Anal. Cloud, (I-SMAC)*, Palladam, India, Feb. 2017, pp. 492–496.

[108] J. C. Laprie, "From dependability to resilience," in *Proc. Int. Conf. DSN*, Anchorage, AK, USA, 2008, pp. G8–G9.

[109] C. Tsigkanos, S. Nastic, and S. Dustdar, "Towards resilient Internet of Things: Vision, challenges, and research roadmap," in *Proc. IEEE 39th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Dallas, TX, USA, Jul. 2019, pp. 1754–1764.

[110] J. P. G. Sterbenz, D. Hutchison, E. K. Çetinkaya, A. Jabbar, J. P. Rohrer, M. Schöller, and P. Smith, "Resilience and survivability in communication networks: Strategies, principles, and survey of disciplines," *Comput. Netw.*, vol. 54, no. 8, pp. 1245–1265, Jun. 2010.

[111] V. Prokhorenko and M. A. Babar, "Architectural resilience in cloud, fog and edge systems: A survey," *IEEE Access*, vol. 8, pp. 28078–28095, 2020.

[112] K. A. Delic, "On resilience of IoT systems: The Internet of Things (ubiquity symposium)," in *Proc. Ubiquity*, Feb. 2016, pp. 1–7.

[113] S. Hosseini, K. Barker, and J. E. Ramirez-Marquez, "A review of definitions and measures of system resilience," *Rel. Eng. Syst. Saf.*, vol. 145, pp. 47–61, Jan. 2016.

[114] A. Jonathan, M. Uluyol, A. Chandra, and J. Weissman, "Ensuring reliability in geo-distributed edge cloud," in *Proc. Resilience Week (RWS)*, Wilmington, DE, USA, Sep. 2017, pp. 127–132.

[115] Y. Harchol, A. Mushtaq, J. McCauley, A. Panda, and S. Shenker, "CESSNA: Resilient edge-computing," in *Proc. Workshop Mobile Edge Commun. (MECOMM)*. New York, NY, USA: Association for Computing Machinery, Aug. 2018, pp. 1–6.

[116] T. Jeong, J. Chung, J. W.-K. Hong, and S. Ha, "Towards a distributed computing framework for fog," in *Proc. IEEE Fog World Congr. (FWC)*, Santa Clara, CA, USA, Nov. 2017, pp. 1–6.

[117] D. Zeng, L. Gu, S. Guo, Z. Cheng, and S. Yu, "Joint optimization of task scheduling and image placement in fog computing supported software-defined embedded system," *IEEE Trans. Comput.*, vol. 65, no. 12, pp. 3702–3712, Dec. 2016.

[118] B. P. Rimal, D. P. Van, and M. Maier, "Mobile-edge computing versus centralized cloud computing over a converged FiWi access network," *IEEE Trans. Netw. Service Manage.*, vol. 14, no. 3, pp. 498–513, Sep. 2017.

[119] A. Machen, S. Wang, K. K. Leung, B. J. Ko, and T. Salonidis, "Live service migration in mobile edge clouds," *IEEE Wireless Commun.*, vol. 25, no. 1, pp. 140–147, Feb. 2018.

[120] S. Helmer, C. Pahl, J. Sanin, L. Miori, S. Brocanelli, F. Cardano, D. Gadler, D. Morandini, A. Piccoli, S. Salam, A. M. Sharear, A. Ventura, P. Abrahamsson, and T. D. Oyetoyan, "Bringing the cloud to rural and remote areas via cloudlets," in *Proc. 7th Annu. Symp. Comput. Develop. (ACM DEV)*. New York, NY, USA: Association for Computing Machinery, 2016, pp. 1–10.

[121] A. Jonathan, A. Chandra, and J. Weissman, "Locality-aware load sharing in mobile cloud computing," in *Proc. 10th Int. Conf. Utility Cloud Comput. (UCC)*. New York, NY, USA: Association for Computing Machinery, Dec. 2017, pp. 141–150.

[122] R. K. Naha, S. Garg, D. Georgakopoulos, P. P. Jayaraman, L. Gao, Y. Xiang, and R. Ranjan, "Fog computing: Survey of trends, architectures, requirements, and research directions," *IEEE Access*, vol. 6, pp. 47980–48009, 2018.

[123] K. Fall, "A delay-tolerant network architecture for challenged Internets," in *Proc. Conf. Appl., Technol., Archit., Protocols Comput. Commun. (SIGCOMM)*. New York, NY, USA: Association for Computing Machinery, 2003, pp. 27–34.

[124] O. Kovalchuk, Y. Gordienko, and S. Stirenko, "The impact of MQTT-based sensor network architecture on delivery delay time," in *Proc. IEEE 39th Int. Conf. Electron. Nanotechnol. (ELNANO)*, Kyiv, Ukraine, Apr. 2019, pp. 838–842.

[125] J. E. Luzuriaga, M. Zennaro, J. C. Cano, C. Calafate, and P. Manzoni, "A disruption tolerant architecture based on MQTT for IoT applications," in *Proc. 14th IEEE Annu. Consum. Commun. Netw. Conf. (CCNC)*, Las Vegas, NV, USA, Jan. 2017, pp. 71–76.

[126] C. Kulatunga, L. Shalloo, W. Donnelly, E. Robson, and S. Ivanov, "Opportunistic wireless networking for smart dairy farming," *IT Prof.*, vol. 19, no. 2, pp. 16–23, Mar./Apr. 2017.

[127] M. Al-khafajiy, T. Baker, A. Waraich, D. Al-Jumeily, and A. Hussain, "IoT-fog optimal workload via fog offloading," in *Proc. IEEE/ACM Int. Conf. Utility Cloud Comput. Companion (UCC Companion)*, Zürich, Switzerland, Dec. 2018, pp. 359–364.

[128] G. Castellano, F. Risso, and R. Loti, "Fog computing over challenged networks: A real case evaluation," in *Proc. IEEE 7th Int. Conf. Cloud Netw. (CloudNet)*, Tokyo, Japan, Oct. 2018, pp. 1–7.

[129] H. Baqa, N. B. Truong, N. Crespi, G. M. Lee, and F. L. Gall, "Quality of information as an indicator of trust in the Internet of Things," in *Proc. 17th IEEE Int. Conf. Trust, Secur. Privacy Comput. Commun./12th IEEE Int. Conf. Big Data Sci. Eng. (TrustCom/BigDataSE)*, New York, NY, USA, Aug. 2018, pp. 204–211.

[130] O. B. Sezer, E. Dogdu, and A. M. Ozbayoglu, "Context-aware computing, learning, and big data in Internet of Things: A survey," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 1–27, Feb. 2018.

[131] D. A. Patterson and J. L. Hennessy, *Computer Organization and Design: The Hardware/Software Interface*, 5th ed. San Mateo, CA, USA: Morgan Kaufmann, Sep. 2013.

[132] C. Maiorano, E. Pascale, L. Bouillaut, P. Sannino, Y. Solorzano, S. Borriello, P. Marmo, and F. Schenkelberg, "MTBF (metric that betrays folk)," in *Proc. 29th Eur. Saf. Rel. Conf.*, 2019, p. 6.

[133] O. Elijah, T. A. Rahman, I. Orikumhi, C. Y. Leow, and M. N. Hindia, "An overview of Internet of Things (IoT) and data analytics in agriculture: Benefits and challenges," *IEEE Internet Things J.*, vol. 5, no. 5, pp. 3758–3773, Oct. 2018.

[134] I. Zyrianoff, F. Borelli, G. Biondi, A. Heideker, and C. Kamienski, "Scalability of real-time IoT-based applications for smart cities," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Natal, Brazil, Jun. 2018, pp. 00688–00693.

[135] B. Queté, A. Heideker, I. Zyrianoff, D. Ottolini, J. H. Kleinschmidt, J.-P. Soininen, and C. Kamienski, "Understanding the tradeoffs of LoRaWAN for IoT-based smart irrigation," in *Proc. IEEE Int. Workshop Metrol. Agricult. Forestry (MetroAgriFor)*, Nov. 2020, pp. 73–77.

[136] K. Mekki, E. Bajic, F. Chaxel, and F. Meyer, "A comparative study of LPWAN technologies for large-scale IoT deployment," *ICT Exp.*, vol. 5, no. 1, pp. 1–7, Mar. 2019.

[137] J.-S. Lee, Y.-W. Su, and C.-C. Shen, "A comparative study of wireless protocols: Bluetooth, UWB, ZigBee, and Wi-Fi," in *Proc. IECON, 33rd Annu. Conf. IEEE Ind. Electron. Soc.*, Taipei, Taiwan, Nov. 2007, pp. 46–51.

[138] M. B. Yassein, W. Mardini, and A. Khalil, "Smart homes automation using Z-wave protocol," in *Proc. Int. Conf. Eng. MIS (ICEMIS)*, Agadir, Morocco, Sep. 2016, pp. 1–6, doi: 10.1109/ICEMIS.2016.7745306.

[139] S. Sadowski and P. Spachos, "Solar-powered smart agricultural monitoring system using Internet of Things devices," in *Proc. IEEE 9th Annu. Inf. Technol., Electron. Mobile Commun. Conf. (IEMCON)*, Vancouver, BC, Canada, Nov. 2018, pp. 18–23.

[140] J. Arshad, M. A. Azad, M. M. Abdeltaif, and K. Salah, "An intrusion detection framework for energy constrained IoT devices," *Mech. Syst. Signal Process.*, vol. 136, Feb. 2020, Art. no. 106436.

[141] O. Said, Z. Al-Makhadmeh, and A. M. R. Tolba, "EMS: An energy management scheme for green IoT environments," *IEEE Access*, vol. 8, pp. 44983–44998, 2020.

[142] M. Chen, S. Mao, and Y. Liu, "Big data: A survey," *Mobile Netw. Appl.*, vol. 19, no. 2, pp. 171–209, Apr. 2014.

[143] V. C. Storey and I.-Y. Song, "Big data technologies and management: What conceptual modeling can do," *Data Knowl. Eng.*, vol. 108, pp. 50–67, Mar. 2017.
[144] J. Azar, A. Makhoul, M. Barhamgi, and R. Couturier, "An energy efficient IoT data compression approach for edge machine learning," *Future Gener. Comput. Syst.*, vol. 96, pp. 168–175, Jul. 2019.
[145] M. Lavassani, S. Forsström, U. Jennehag, and T. Zhang, "Combining fog computing with sensor mote machine learning for industrial IoT," *Sensors*, vol. 18, no. 5, p. 1532, May 2018.
[146] Y. Shi, G. Ding, H. Wang, H. E. Roman, and S. Lu, "The fog computing service for healthcare," in *Proc. 2nd Int. Symp. Future Inf. Commun. Technol. Ubiquitous HealthCare (Ubi-HealthTech)*, Beijing, China, May 2015, pp. 1–5.
[147] N. Narendra, K. Ponnalagu, A. Ghose, and S. Tamilselvam, "Goal-driven context-aware data filtering in IoT-based systems," in *Proc. IEEE 18th Int. Conf. Intell. Transp. Syst.*, Las Palmas, Spain, Sep. 2015, pp. 2172–2179.
[148] F. M. Ribeiro, R. Prati, R. Bianchi, and C. Kamienski, "A nearest neighbors based data filter for fog computing in IoT smart agriculture," in *Proc. IEEE Int. Workshop Metrol. Agricult. Forestry (MetroAgriFor)*, Trento, Italy, Nov. 2020, pp. 63–67.
[149] C. A. Kamienski, F. F. Borelli, G. O. Biondi, I. Pinheiro, I. D. Zyrianoff, and M. Jentsch, "Context design and tracking for IoT-based energy management in smart cities," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 687–695, Apr. 2018.
[150] M. Kiran, P. Murphy, I. Monga, J. Dugan, and S. S. Baveja, "Lambda architecture for cost-effective batch and speed big data processing," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Santa Clara, CA, USA, Oct. 2015, pp. 2785–2792.
[151] *ChirpStack, Open-Source LoRaWAN Network Server Stack*. Accessed: Nov. 2020. [Online]. Available: https://www.chirpstack.io/

FRANKLIN MAGALHÃES RIBEIRO JUNIOR received the B.S. and M.S. degrees in computer science from the Federal University of Sergipe (UFS), Aracaju, Brazil, in 2013 and 2015, respectively. He is currently pursuing the Ph.D. degree in computer science with the Federal University of ABC (UFABC).

He is also a Lecturer of information technology with the Federal Institute of Maranhão (IFMA). He is also a Research Fellow of the NUVEM Strategic Research Unit. His research interests include dependable systems, smart cities, and fog computing.

CARLOS ALBERTO KAMIENSKI (Senior Member, IEEE) received the B.S. degree in computer science from the Federal University of Santa Catarina, Florianópolis, Brazil, in 1989, the M.S. degree from State University of Campinas, Campinas, Brazil, in 1994, and the Ph.D. degree in computer science from the Federal University of Pernambuco, Recife, Brazil, in 2003. He is currently a Full Professor of computer science with the Federal University of ABC (UFABC), Santo André, Brazil, where he also holds the position of the Head of the NUVEM Strategic Research Group. His current research interests include the Internet of Things, smart cities, smart agriculture, network softwarization, and fog computing.