

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.Doi Number

# An Empirical Study on System Level Aspects of Internet of Things (IoT)

**NARASIMHA SWAMY S<sup>1,2,3</sup>, Student Member, IEEE  
KOTA SOLOMON RAJU<sup>1,2</sup>, Senior Member, IEEE**

<sup>1</sup> Academy of Scientific and Innovative Research (AcSIR), Ghaziabad, India

<sup>2</sup> CSIR- CEERI, Pilani, Rajasthan, India

<sup>3</sup> Rashtreeya Vidyalaya College of Engineering (RVCE), Bengaluru, Karnataka, India

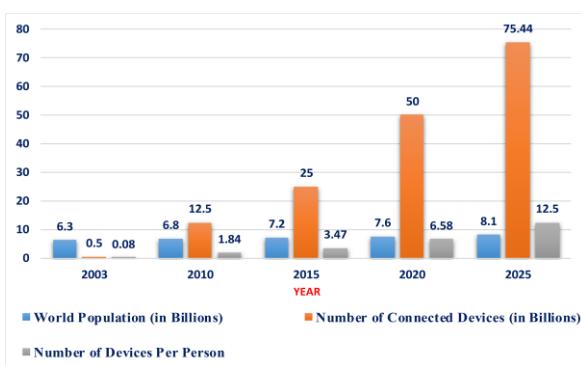
Corresponding author: Narasimha Swamy S (narasimha.rmgm@gmail.com).

**ABSTRACT** Internet of Things (IoT) is an integration of the Sensor, Embedded, Computing, and Communication technologies. The purpose of the IoT is to provide seamless services to anything, anytime at any place. IoT technologies play a crucial role everywhere, which brings the fourth revolution of disruptive technologies after the internet and Information and Communication Technology (ICT). The Research & Development community has predicted that the impact of IoT will be more than the internet and ICT on society, which improves the well-being of society and industries. Addressing the predominant system-level design aspects like energy efficiency, robustness, scalability, interoperability, and security issues result in the use of a potential IoT system. This paper presents the current state of art of the functional pillars of IoT and its emerging applications to motivate academicians and researches to develop real-time, energy-efficient, scalable, reliable, and secure IoT applications. This paper summarizes the architecture of IoT, with the contemporary status of IoT architectures. Highlights of the IoT system-level issues to develop more advanced real-time IoT applications have been discussed. Millions of devices exchange information using different communication standards, and interoperability between them is a significant issue. This paper provides the current status of the communication standards and application layer protocols used in IoT with the detailed analysis. The computing paradigms like Cloud, Cloudlet, Fog, and Edge computing facilitate IoT with various services like data offloading, resource and device management, etc. In this paper, an exhaustive analysis of Edge Computing in IoT with different edge computing architectures and existing status are deliberated. The widespread adoption of IoT in society has resulted in privacy and security issues. This paper emphasizes on analyzing the security challenges, privacy and security threats, conventional mitigation techniques, and further scope for IoT security. The features like fewer memory footprints, scheduling, real-time task execution, fewer interrupt, and thread switching latency of Real-Time Operating Systems (RTOS) enables the development of time critical IoT applications. Also, this review offers the analysis of the RTOS's suitable for IoT with the current status and networking stack. Finally, open research issues in IoT system development are discussed.

**INDEX TERMS** IoT, Pillars of IoT, Emerging IoT Applications, IoT Application Requirements, IoT Architecture, IoT Application Layer Protocols, Computing Paradigms (Edge, Fog, Cloudlets & Cloud), Privacy & Security, and Platforms for IoT

## I. INTRODUCTION

The Internet plays a significant role in information transmission [1]. However, the technology is moving towards data collection, analysis of the data, and controlling devices remotely via the Internet rather than just sharing the information, and this results in a new technology called IoT. IoT is an interconnection of various physical devices to collect, control, analyze, and share data in real-time[2][3][4]. IoT aims to enhance the quality of life [5]. The motivating factor for the extensive growth of IoT technology is that most of the manufacturing industries, service providers, and software industries are investing



**FIGURE 1.** Number of Devices Predicted to be Connected to the Internet by 2025

more and adopting IoT technologies more swiftly. Forbes estimated that the transformation of the hypothetical concepts of IoT to reality started during the year 2015 [6]. As per the survey, the number of internet-connected objects are likely to be 75.44 billion [7] and the economic growth of the IoT technology will range from \$2.7 to \$6.2 trillion by 2025 [8]; this shows the impact of IoT technology on society. Figure 1 illustrates the number of devices predicted to be connected to the internet by 2025, and these devices generate approximately 80 Zettabytes of data [9]. Figure 2 represents the research progress in the area of IoT [10]. IEEE Digital Library (IEEE Xplore) is considered for the search and selection process. IoT or Internet of Things is used as a search keyword. The number of conferences and journal papers are retrieved by limiting the year field of the IEEE Xplore.

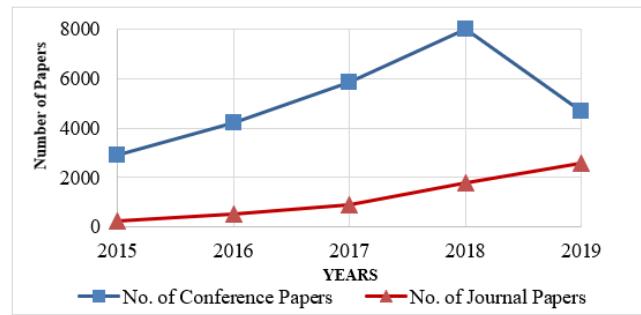


FIGURE 2. IoT Research Trend from 2015 (Source: IEEE Digital Library)

IoT has an impact on most of the applications such as healthcare, intelligent homes, smart farming, factory automation and industry 4.0, intelligent transport systems, smart cities, infrastructure monitoring, retail industry, environmental monitoring, smart water, and power grids, etc. [11]. Figure 3 shows the diversity of IoT applications.

IoT technology has traversed a long way (almost a decade) and offers numerous opportunities. Many IoT components and devices with multiple communication standards, messaging protocols, computation technologies, and security algorithms are under development. For example, different vendors such as ARM, Atmel, Silicon Labs, Texas Instruments, Intel, NVIDIA, Samsung, etc. are the primary producers of IoT components/ chips and development tools with their standards. BLE, ZigBee, WiFi, Z-wave, Sigfox, NB-IoT, 5G, etc. are the communications standards used by IoT devices for short and long-range communication. IoT applications use application/ messaging protocols like MQTT, CoAP, DDS, XMPP, RESTful APIs/ HTTP to send and receive messages. Also, IoT applications utilize computing technologies like Edge, Fog, Cloudlets, and Cloud to store, process, and analyze the data. In IoT, data plays a significant role in making decisions, which is highly private and sensitive.

Figure 4 illustrates the hybrid architecture of the IoT ecosystem. From Figure 4, it is being noticed that computation and communication technologies are the primary requirements of an IoT system. It also shows how

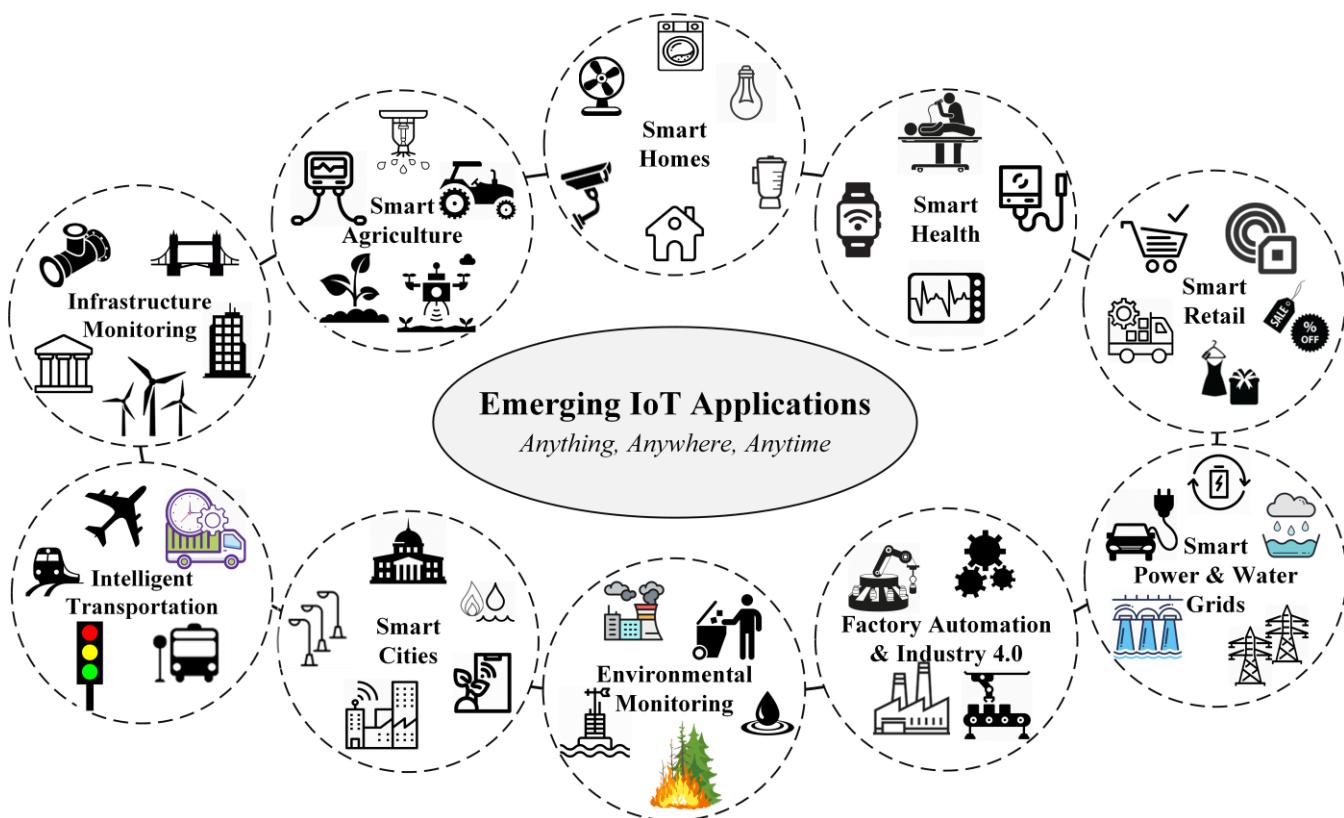
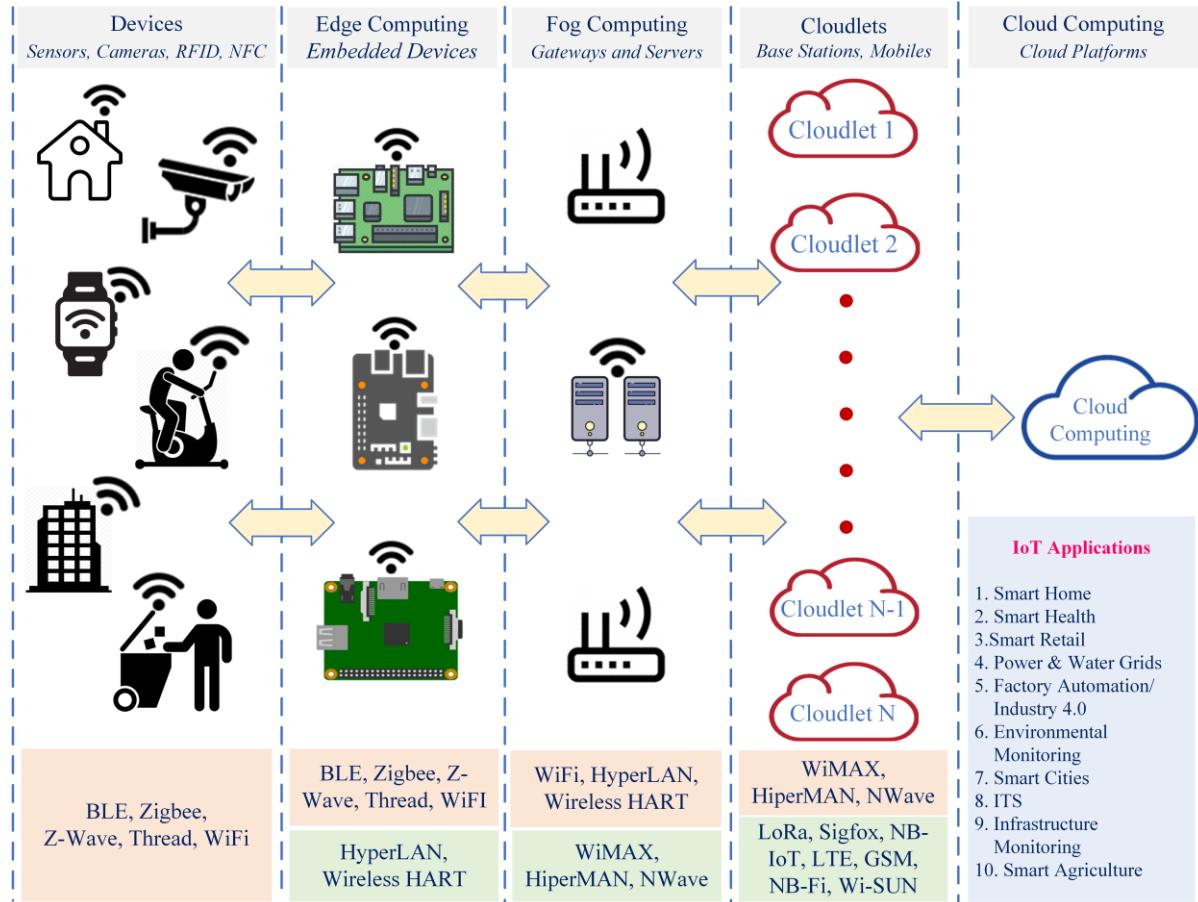


FIGURE 3 Applications of IoT



**FIGURE 4.** Hybrid Architecture of IoT

the above technologies have enabled the development and deployment of an IoT system. Figure 5 illustrates the detailed deployment scenario of IoT. IoT technology comprises of End-node, Edge, Fog, Cloudlets, and Cloud computing technologies, which brings the intelligence to IoT. In general, sensors and signal-conditioning circuits are connected to the end-nodes; these nodes are used when the application needs to deploy a massive number of high-density sensors to capture the physical parameters from the environment. Generally, a group of end-nodes forms a cluster. The edge-nodes serve as a head of the cluster, which collects the raw data from the end-nodes and preprocess the data. Also, edge-nodes are accountable for reliable data transmission using short and long-range communication standards. Introducing the edge-nodes at the edge of the network provides the different deployment scenarios based on the services required by the application. Scenario 1: In this, the edge-node forwards the data to the gateway (acts as fog node) for further processing, Scenario 2: In this, mobile edge nodes access to the services offered by the cloudlets, and Scenario 3: The edge nodes directly access the cloud services. Further, Gateway or BS acts as a fog node, which does the fog computing. Data processing and information extraction are the more common operations performed at the fog level. This information is also used to provide essential services to the underlying devices (Edge and end nodes), including the estimation of optimized

communication costs. Also, it establishes the communication either to cloudlets and cloud infrastructure to store and deliver the more service-oriented value(s) at a higher level. Generally, cloudlets services are used when IoT devices are highly mobile. For example: Streaming the High Definition (HD) video/ movie in an autonomous car. To share data effectively, IoT devices (end nodes, edge nodes, fog node, and cloudlets) use either short-range (RFID, BLE, Zigbee, Z-Wave, Thread) or long-range (LoRaWAN, NB-IoT, Sigfox, 5G, Telensa, Ingenu) communication standards. In IoT, both computation and communication resources need to be utilized efficiently and carefully since IoT devices are resource constraint devices and pose many challenges. The system-level aspects like energy-efficiency, robustness, heterogeneity & interoperability, and other aspects like data & device management and QoS parameters need to be considered while realizing a potential and mature IoT system. Additionally, privacy & security issues should be handled by the IoT system design phase. Though, some of these issues are solved in traditional internet technologies. However, it is impractical to apply those solutions/ mechanisms directly to IoT systems; this is due to the use of multiple sensors, communication standards, protocols, and computing technologies along with the demand-driven services in a constrained environment. IoT devices require novel light-weight algorithms with fewer memory

requirements and less computation complexity. This paper deliberates the system-level design issues in the development of an interoperable, secure, scalable, reliable and energy-efficient IoT system.

### A. Comparison of Related Survey Papers

Several survey papers have been published which cover the different characteristics of IoT technology. M. M. Dhanvijay et al. [12] discuss IoT healthcare networks, architectures, WBAN technology, security issues, challenges, and open issues in the IoT healthcare system. In [13], the authors have presented the existing short and long-range communication standards. The authors also provided their views on the emerging communication technologies like Compressive Sensing, Non-Orthogonal Multiple Access, Massive Multiple-Input Multiple-Output (mMIMO), and ML-based random-access protocols. In [14], the authors address the IoT enabling technologies, recent advancements in IoT communication standards (BLE), and application protocols (MQTT, CoAP, HTTP). The authors also presented various IoT applications and security issues. A. Nauman et al. [15] provided a comprehensive study on Multimedia IoT (M-IoT) with a focus on architectures (Agent, SDN, Fog, and AI-based architecture), protocols (Routing and PHY-MAC protocols), and different M-IoT applications. Also, the authors have discussed the importance of the Quality-of Experience (QoE) and QoS features in M-IoT. In [16], authors have investigated the trends in IoT access control and performed a detailed analysis of existing authorization frameworks in IoT. The authors have observed that there is no generic access control mechanism/ system for IoT applications. In [17], the authors have focused on fundamentals of trust modelling, the importance of trust information fusion, possible trust threats affecting the IoT applications, and IoT architecture for modelling the trust. The authors in [18] provided the advantages of fog computing over cloud computing (Characteristics), fog architecture, hardware, and software platforms for fog computing, and its use in various IoT applications, and future research directions. In [19], the authors have deliberated the IoT architecture, security issues, and requirements. The authors also provided the taxonomy of the IoT authentication protocols, detailed analysis of them, and problems that need to be considered while developing novel authentication schemes. K. A. P. da Costa et al. [20] provided a detailed analysis of machine learning algorithms applied to spot the intruders in IoT networks. The authors also provided information related to the different datasets available to conduct research in the field of security. In [21], the authors carried out an exhaustive survey on the state of the art of IoT security. The authors used applications like smart home, healthcare, and smart grids to review privacy, forensics, and other security challenges. The authors in [22] performed a detailed analysis of trust management techniques (E-Lithe, GTRS, TWGA, TrustCEP, TTBBS, etc.) for IoT. Based on the review, the authors have classified the various techniques with the pros

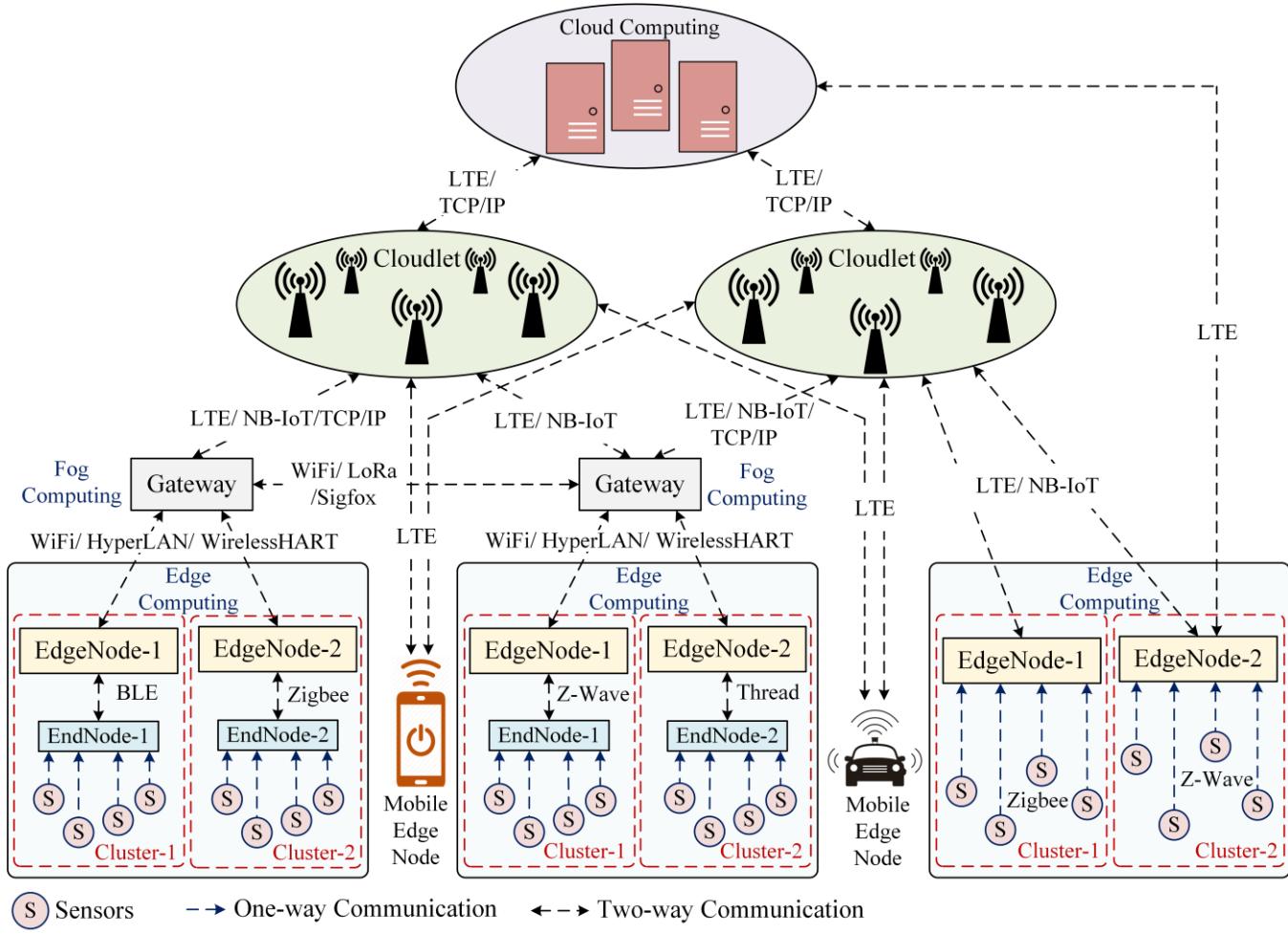
and cons. In [23], the authors provided the taxonomy of the machine learning techniques. Also, the authors have provided a detailed analysis of statistical, data mining, and machine learning techniques used in the different IoT applications, and a comparison of various ML algorithms. The authors also reviewed the different computing architectures used by IoT applications. J. Marietta and B. Chandra Mohan [24] performed a detailed analysis of the WSN and Adhoc routing protocols by considering the characteristics like bandwidth, topology, scalability, and mobility of the nodes. The investigation of these routing protocols is based on the least transmission time and shortest path. Finally, the authors have provided the characteristics of the IoT routing protocols. In [25], the authors presented the taxonomy of the security requirements and analyzed IoT security attacks. Also, the authors have analyzed few security solutions such as Intelligent Security Framework, Chaos-based Privacy Preservation, etc. In [26], the authors offered the review of currently available testbeds like MoteLab, ORBIT, Trio, Indriya, Flocklab, Sensorscope, TWIST, etc. for WSN and IoT applications. They have performed a detailed analysis of testbeds based on the characteristics. Y Lu et al. [27] discussed the cybersecurity issues in detail, along with the cybersecurity attacks taxonomy and middleware layered cybersecurity architecture. The authors also highlighted the different security schemes like Host identity, Datagram Transport layer, and Capability-based access control scheme. In [28], the authors have discussed the architecture, protocols (Physical, Transport, and Application Layer protocols, WSN and IoT routing protocols), scalability, and privacy and security concepts in IoT. Table I depicts the summary of the survey papers in the IoT area.

### B. Motivation

IoT applications are gradually evolving day by day. However, the development of a scalable, reliable, energy-efficient architectures and secure IoT system remains challenging in practice. In literature, the authors have discussed the various aspects of IoT. However, no works have presented a detailed analysis of the IoT system design issues. Therefore, increased recognition and more attention need to be imparted to this area. Hence, this survey provides a comprehensive study of system-level aspects and enabling technologies of IoT with a focus on interoperability, application layer protocols and security.

### C. Contributions to this Survey

The rapid growth of smart devices and the use of various technologies in IoT poses countless research opportunities. In reality, there are no large-scale IoT applications till date. Industries, research organizations, and academicians need to address the current research issues and development of standards in IoT. This survey paper aims to provide an extensive understanding of the current research status of IoT system-level aspects. It also provides the research directions in emerging IoT application domains, architectures, communication standards and application



**FIGURE 5. IoT Deployment Scenario**

protocols, computing paradigms, and RTOS. In summary, this work aims

- To provide deeper insights into the IoT system to the readers
- To describe the functional pillars of IoT
- To give an overview of emerging IoT applications, its classification, characteristics and requirements
- To review the IoT application requirements and challenges
- To realize the IoT (Three and Five layered architecture) and examine the relationship between the IoT device, OS's and IoT protocols in the layered architecture
- To analyze various communication standards and application layer protocols in IoT
- To review the computing paradigms like edge computing and its classification, fog, cloudlet and cloud computing architectures, characteristics and current status in IoT
- To discuss various privacy & security issues, threats, mitigation techniques, and current state of the art in IoT security
- To analyze the characteristics of different OS's for IoT and realizing the IoT protocol stack of different OS's.

- To provide open research issues and future directions in IoT.

The remaining part of the paper is organized as follows. Section II provides functional pillars of IoT. Section III deliberates emerging and popular IoT applications. Section IV reviews the IoT system-level challenges and current research state. Section V analyses the different IoT architectures, the current state of the art, and issues that need to be considered during architectural design. Section VI and VII provide a detailed analysis of communication standards and application layer protocols used in IoT, respectively. Section VIII provides the in-depth analysis of computing paradigms (Edge, Fog, Cloudlets, and Cloud computing architectures), different edge architectures, and areas where further research is required. Section IX provides a detailed analysis of privacy and security issues, mitigation techniques, and what needs to be addressed in the future. Section X highlights the platform support for IoT, and protocol stack supported by them. Section XI lists open research issues and future directions that need to be addressed. Finally, Section XII provides the concluding remarks.

TABLE I  
 SUMMARY OF THE SURVEY PAPERS

Ref. & Year	Applications	IoT Challenges	System Design Aspects			Computing Paradigms	Protocols	Security & Privacy
			Architecture	Communication	OS			
[14] 2020	Smart Home, Smart Retail, Smart Agriculture, Industry, Environment, Smart Health, Smart City	✗	✗	BLE, Zigbee, WiFi, LTE, 5G, Sigfox, NB-IoT	✗	✗	MAC Layer (CSMA, CSMA/CA, TDMA, CDMA)	✗
[15] 2019	Healthcare, Industrial IoT, Home Automation, Smart Cities, Emergency Care, Smart Agriculture	✗	✗	✗	✗	✗	Application Layer (MQTT, CoAP)	✓
[16] 2020	Road Management, Surveillance, Industrial application, Healthcare	✗	Three, Middleware, SDN & Cloud based Layered Architecture	Zigbee, WLAN, NB-IoT, BLE, LoRaWAN	✗	Cloud, Fog & SDN Computing	Routing and PHY-MAC Protocols	✗
[17] 2019	Smart Homes, Smart Health, Smart Building, Connected Vehicles, Manufacturing	✗	Middleware Architecture	Zigbee, WiFi, BLE, LoRaWAN, Z-Wave	✗	✗	HTTP, CoAP, MQTT, Restful, XMPP	Access Control
[13] 2019	Healthcare	Scalability, Privacy & Security	Middleware Architecture	✗	✗	✗	✗	✓
[19] 2019	Intelligent Transportation System, Public Safety, Smart Grids, Industry 4.0, Smart Homes	Security	Middleware Architecture	✗	✗	Fog Computing	✗	✗
[20] 2019	Smart Grids, Vehicular Networks, Smart Homes, Mobile Applications,	✗	Three, Five and Middleware Architecture	✗	✗	✗	✗	Authentication
[23] 2020	Automotive, Environmental Monitoring, Agriculture, Healthcare, Industrial, Retail, Banking, Supply Chain, Smart Homes, Smart City	✗	✗	✗	✗	Cloud, Fog & Edge Computing	✗	✗
[26] 2019	Smart Home, Healthcare, Transportation	✗	Middleware Architecture	Zigbee	✗	✗	6LowPAN, RPL, CoAP	✓
[27] 2020	Smart Homes, Smart Health, Smart Transportation, Retail, Industry 4.0, Smart Agriculture	Security	Middleware Architecture	RFID, BLE, Zigbee	✗	✗	Application Protocols (MQTT, CoAP, HTTP)	✗
This Survey	Smart Homes, Smart Health, Smart Farming, Intelligent Transportation, Factory Automation/ Industry 4.0, Smart Retail, Smart Cities, Environmental Monitoring, Infrastructure Monitoring, Smart Power and Water Grids	<b>System Level Challenges</b>  Architectures, Reliability, Interoperability, Self-Diagnosis & Adaptability <b>Other Challenges</b> Availability, Mobility, Scalability, Device and Data Management, Privacy & Security	Three, Five and Middleware Architecture	Communication Models and Standards RFID, Zigbee, Z-wave, WiFi, NB-IoT, BLE, LoRaWAN, Sigfox, 5G	Real-Time Operating Systems and IoT Protocol Stack	Cloud, Cloudlets, Fog and Edge Computing & its Architecture	Application Layer (MQTT, CoAP, XMPP, DDS, Restful APIs)	General Attacks, Mitigation Techniques

## NOMENCLATURE

AES	Advanced Encryption Standard
AMQP	Advanced Message Queuing Protocol
APIs	Application Interfaces
BAN	Body Area Networks
BLE	Bluetooth Low Energy
BS	Base Station
CART	Classification and Regression Tree
CHAID	Chi-squared Automatic Interaction Detector
CRP	Challenge-Response Pair
CoAP	Constrained Application Layer Protocol
DDS	Data Distribution Services
DTLS	Datagram Transport Layer Security
GIS	Geographical Information System
HTTP	Hypertext Transfer Protocol
IDPs	Intrusion Detection and Prevention System
IEC	International Electrotechnical Commission
IoT	Internet of Things
ISO	International Standard Organization
ITU	International Telecommunication Union
ITS	Intelligent Transportation System
KNN	K-Nearest Neighbor
LTE	Long-Term Evolution
MAC	Medium Access Control
ML	Machine Learning
MQTT	Message Queuing Telemetry Transport
NS3	Network Simulator-3
OS	Operating System
PDR	Packet Delivery Ratio
QoS	Quality of Services
QUEST	Quick, Unbiased, Efficient, Statistical Tree
RSSI	Received Signal Strength Indicator
SVM	Support Vector Machine
TLS	Transport Layer Security
TIS	Traveler Information System
WiFi	Wireless Fidelity
WWW	World Wide Web
XMPP	Extensible Messaging and Presence Protocol
6LoWPAN-	IPv6 Over Low-Power Personal Area
NHC	Networks - Next Header Compression

## II. PILLARS OF IOT

IoT increases the quality of life by providing numerous application services to the users. Sensing & Actuation, device identification, communication, computation, application services, management, and security are the major functional pillars of an IoT system. Figure 6 illustrates the functional pillars of IoT.

### 1) Sensing and Actuation

In IoT, sensing is all about collecting various data from the environment using different sensors. The IoT sensors can be acoustic, chemical, biological, pressure, thermal,

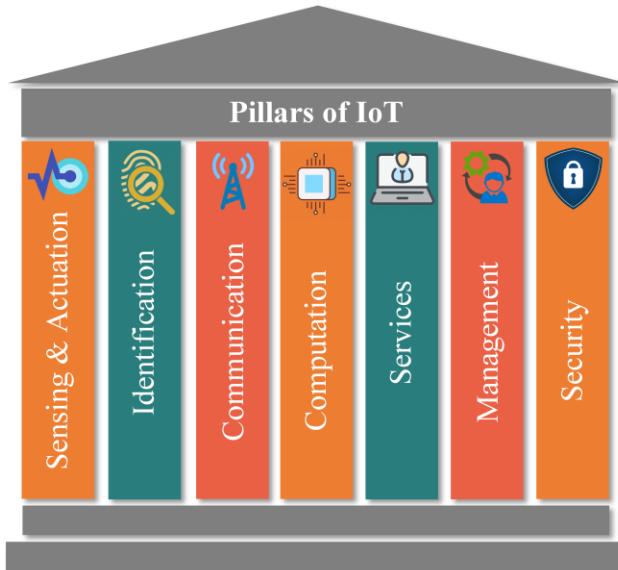


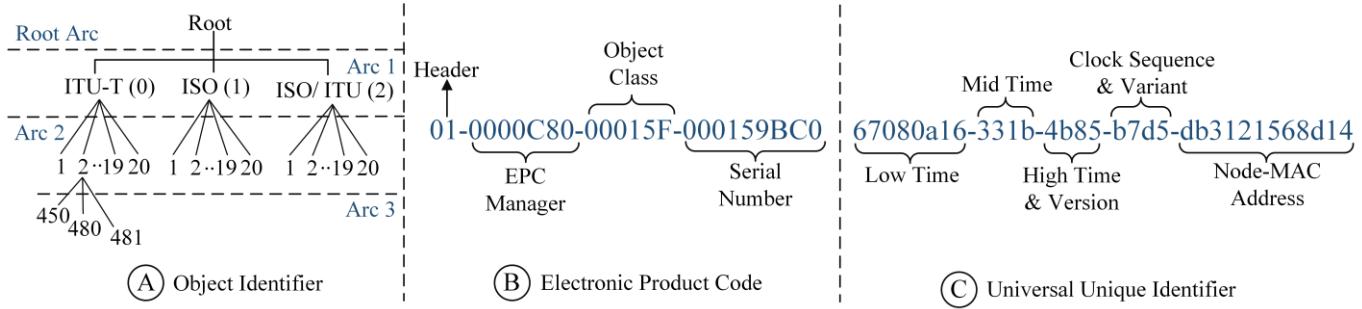
FIGURE 6. Pillars of IoT

wearable, implantable sensors, actuators, RFID Tags, etc. [29][30]. The data collected from these sensors are processed and analyzed to actuate and improve decision making in IoT. IoT is a collection of various types of sensors, which have their own requirements. To maintain the standardization among the IoT sensors, IEEE Electronic Engineering Association introduced the IEEE1451 smart transducer protocol stack for the development of smart sensors [31].

### 2) Device Identification

Devices are the central pillar of IoT; these are also called as objects/ things, which are the essential sources of data [32]. Object identification is used to identify the entity of interest in an IoT application. Objects are used to perceive, operate, monitor, and control the IoT application services. In IoT, devices need to be identified uniquely to provide application services with an increase in security. Object Identifier (OID), Electronic Product Code (EPC), and Universal Unique Identifiers (UUID) are commonly used to identify the devices within the network [11]. Figure 7 depicts the various identification standards used in IoT.

- a. OID: It is an object identification mechanism collaboratively developed by ITU-T and ISO/ IEC. This mechanism uses the hierarchical structure to assign unique identification numbers to the objects. Levels of the OID tree are referred to as an arc. The highest arc in the tree is the root, which does not contain any name, and comprises of three management organization such as ITU-T, ISO, and ISO-ITU-T (Jointly managed by ISO and ITU-T). These management organizations are uniquely identified using the values 0, 1, and 2, respectively. The second arc represents the category of the data, which belongs to (For example, administration, recommendation, questions, etc.), and the third arc represents the country code. The oneM2M system prefixes the OID with the device identity [33]. An OID



**FIGURE 7. Identification Standards in IoT**

mechanism was introduced to eliminate the interoperability at the device identification level [34].

- EPC: It is a unique identifier used to identify the devices around the globe. EPC is used to track the object information within the EPCglobal network. EPC is 64-bit (I, II, and III) and 96-bits long. Typically, a 96-bit EPC comprises of a header (8-bit)- used to identify the EPC version number, EPC manager (28-bit)- used to identify the manufacturer of the item, object class (24-bit)- identifies the type of product made by the manufacturer and serial number (36-bit)-identifier for the individual item. The first two fields are assigned by the EPCglobal, and the remaining fields are assigned by the EPC manager [11][34].
- UUID or Globally Unique Identifier (GUID): It is a 128-bit number used to identify an object uniquely. The probability of reproducing the same UUID is almost zero. The general format of the UUID comprise of five groups, such as 8-4-4-4-12 bits. The sample UUID is as follows 67080a16-331b-4b85-b7d5-db3121568d14. UUIDs are widely used in databases like MySQL, MariaDB and etc. as database keys [35]. Also, UUID are used to identify the services provided by the BLE devices in IoT network.

For example, in V2V communications, connected vehicles are uniquely identified using ISO 3779 standard. ISO 11784 is used to identify the animals uniquely in smart farming and animal husbandry. Similarly, ISO/ IEC 15459 is used to identify the products in the supply chain [36].

### 3) Communication

The aim of this element is to share the captured data using different communication standards across the globe, and a detailed description of this is presented in Section VI.

### 4) Computation

Processing units such as microprocessors, microcontrollers, SoCs, and FPGAs and software packages act as a brain of IoT. The operating systems like TinyOS, RIOT, Ubuntu Core, Raspbian, Contiki OS, etc. provides the networking and IoT application development environment [37]. Cloud, cloudlets, fog, and edge platforms facilitate the data storage, analysis, and ubiquity in IoT [38].

### 5) Services

Device Monitoring, Device Controlling, Data publishing,

and Device Discovery are the services offered by an IoT system.

## 6) Management

This aims to manage the IoT users, roles of the users, and access the IoT services. The managing of users includes adding and removing the user from the group. It also controls and records user activities.

- Device Management: Various operations like - authenticating, configuring, monitoring, and maintaining the objects in the network are described in this block [39][40].
- Service Management: IoT provides the following services [41]
  - Information Aggregation Service: The primary concern of this service is to sense, collect, and store the data from multiple environments and process the same to extract information through IoT infrastructure.
  - Collaborative Aware Services: Based on the aggregated data received, this service forms the better decision and makes reactions. The effectiveness of this service is based on the reliability of the infrastructure and computation cost.
  - Ubiquitous Services: This provides the Collaborative Aware Services at any-time and anywhere to anyone.

## 7) Security

This element intends to secure the Devices, Communication, and Services of an IoT system from modification and unauthorized access. A detailed analysis of this element is presented in Section VIII.

- Security in Devices: IoT device security is the process of protecting the objects connected over a network from the attacker [42]. The typical lifecycle of IoT objects consists of Booting (Loading of Firmware), Initialization (Connection Establishment and Data Collection), Operation (Desired Functionality of the Object), and Update (Installation of New Firmware and Rebooting). The entire lifecycle of IoT objects should be protected by employing security algorithms.
- Security in Communication: This provides end to end protection of the communication channels between the

objects. Most of the objects in the IoT system uses wireless communication technologies, and these channels are easily prone to various types of attacks [43].

- c. Security in Services: Securing the application services and its data from unauthorized access and modification by employing lightweight security algorithms.

### III. EMERGING IOT APPLICATIONS

The applications like Smart Homes, Smart Health, Smart Farming, Smart Shopping, Intelligent Transportation, Factory automation & Industry 4.0, Environmental & Infrastructure Monitoring, Smart Cities, Smart Water & Power Grids etc. use IoT technology to increase the quality of life of human beings [44]. Figure 8 depicts the broad domains and sub-domains of emerging IoT applications.

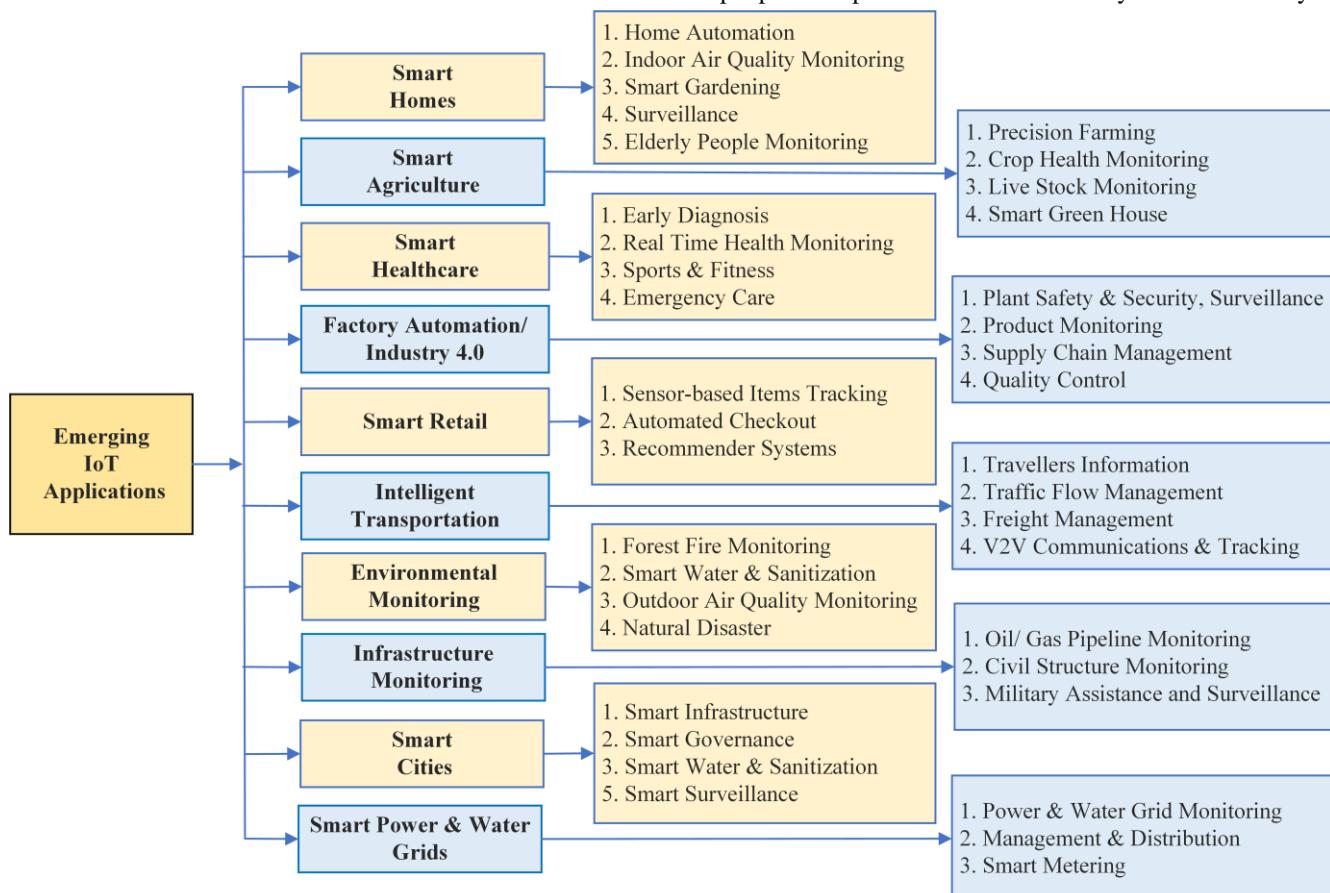
#### A. Smart Homes

Smart homes aim is to augment the resident's quality of life [45]. The smart home is an integration of various domains like - home automation, air quality monitoring, health care, surveillance, and smart gardening.

- **Home Automation:** Home automation enables the residents to control home appliances like Air Conditioners, Fans, Washing machines, Refrigerators, Toasters, Coffee makers, Personal computers,

Smartphones, etc. remotely using the internet [46].

- **Indoor Air Quality Monitoring:** It is essential to supervise the quality of the air inside the home or office, as people spent most of their time at home or in the office. Indoor air is more toxic than outdoor air. According to the survey, the use of stoves, tobacco, and detergents increases the levels of CO<sub>2</sub>, NO, and NO<sub>2</sub>. The researchers have noticed that children are susceptible to mental illness because of these pollutants [47]. IoT facilitates monitoring of the quality of the air and alerts the residents.
- **Smart Gardening:** In this busy life, most of the people do not have time to take care of the garden. Gardening improves the mental health of the residents and air quality in the surrounding habitation. The use of IoT in smart gardening allows people to monitor the plants and provides nutrition and water from time to time remotely [48].
- **Surveillance:** The surveillance using IoT aims to provide safety and security for the people in smart homes [49]. Integration of sensors, human-computer vision techniques, and IoT technology identify the anomalies and intruders, which provide real-time alerts to the residents [50].
- **Elderly People Monitoring:** The population of elderly people is expected to be increased by 16.7% in the year



**FIGURE 8. Broad Application Domains of IoT**

2050 [51]. Elderly people suffer from chronic diseases and conditions like blood pressure, diabetes, heart diseases, and cancer. The advancements in biomedical sensors and IoT allows the monitoring of elderly people health remotely; this increases the average life expectancy of elderly people [52].

Real-time video analysis of elderly people needs to provide identification and classifications of multiple activities such as gait activities, fall detection, facial expression, sleep cycles, etc. Automated health assessment techniques also need to be improved. The smart home concept comprises of various sensors, actuators, and devices like Alexa, Fitbit, wireless speakers, Google assistance, Laptops, and so on; most of these devices are portable and wearable devices, which are used for long-term monitoring purpose. Hence, communication and computation algorithms should be energy efficient. The devices used in smart homes are enormous in number and highly heterogeneous; maintaining the interoperability between them is also a major concern and tracking these devices and handling data generated by them are still challenging. Surveillance cameras should provide real-time alerts to the residents by recognizing and analyzing the multiple crime activities within a short period; this requires high-performance IoT systems to meet the real-time application requirements.

#### **B. Smart Health**

Smart health aims to provide immediate medical service by reducing the cost of the medication to increase the quality of health in humans and pets [53]. Healthcare includes various sub-domains such as fitness programs, remote monitoring of health, chronic disease diagnosis, and monitoring of elderly people. Various sensors, medical, and imaging devices are used to acquire data from the human body.

- **Early Diagnostics:** Early diagnosis help the doctor to find the diseases and disorders in patients at earlier stages. Integration of sensor technologies (Wearable Sensors and Implantable Sensors) and BAN (Collects the real-time data like- sugar levels, BP, ECG, etc. from the patients) [54], and IoT facilitates the doctor to monitor and analyze patient's health conditions remotely. Early diagnosis plays a vital role in observing Alzheimer's disease, Parkinsonism, malignancies and other health conditions [55]. The early diagnostics feature of IoT leads to better treatment and extended existence.

- **Real-Time Health Monitoring (RTHM):** The clinic-centric medical services have been changed to patient-centric medical services; this is due to the advancements in health monitoring devices and IoT. Real-time health monitoring plays a crucial role in monitoring the health of the sick and geriatric people on a timely basis remotely. The RTHM can be adopted by patients who are suffering from various diseases like cardiac and respiratory diseases. Also, patients with conditions like bradycardia, tachycardia, arrhythmia, etc [56].

- **Sports and Fitness:** Most of the people in this digital era are working professionals and engaged in different activities, due to which they have a habit of ignoring health and physical fitness [57]. Staying fit and maintaining the diet is very significant for patients and sportspersons. Smart devices like smartwatches, smartphones, smart gloves and shoes track the features like motion, calories burnt, analysis of sleep cycles, heart rate, etc. IoT enables fitness trainers and coaches to send recommendations to the sportspersons and patients based on the data received which ultimately improves the fitness of patients and athletes.

- **Emergency Care:** Emergency medical care is a network of patients, hospitals, health workers and emergency vehicles. This provides services to the citizens during natural disasters like earthquake, building collapse, accidents, fire accidents, floods, war fields, and others. Emergency care also provides services like locating nearby hospitals and emergency vehicles using the Global Positioning System (GPS) technology [58].

The critical requirements of smart health applications are data acquisition, real-time data processing, maintenance of the patient records, immediate intelligence, effective communication, availability of the data, and privacy and security. Also, the use of Telehealth is increasing due to the inevitable situations like the COVID-19 pandemic, etc. which poses other requirements like seamless real-time video streaming. Interoperability among medical devices is also a significant concern; the researchers should emphasize on developing faster, energy-efficient, and interoperable algorithms. Medical devices are resource-constrained and running the conventional security algorithms; it is very costly in terms of both energy and memory point of view, and maintaining the privacy of patient's health information is also challenging. The absence of real-time data processing may lead to unavoidable situations and even sometimes to death. Efficient big data analytics solutions are required to maintain a vast patient's health records.

#### **C. Smart Farming**

The agriculture industry is playing a vital role in nourishing every individual on earth [59][60]. Most of the farmers still follow the traditional methods in farming. The use of traditional farming techniques causes soil erosion, a decrease in the yield of the crop, waste of water, waste of fertilizers, etc. IoT plays a crucial role in reducing a few problems in agriculture.

- **Precision Farming:** The factors like soil, climate, flora, and water directly affect the growth of the crop, and these factors change from one place to another. Precision farming manages these factors efficiently to more crop using minimal resources. Precision farming is a combination of sensors and associated software. Sensors are used to gather real-time data from the fields, and the software is used to analyze data and manage the available resources [61].

• **Crop Health Monitoring:** Monitoring the health of the crops at regular time intervals increases the quality and quantity of the crops. Generally, most of the crops are majorly affected by fungal and bacterial diseases. IoT devices like drones and tractors are the conventional vehicles used in the agricultural fields to supervise the healthiness of the crop. These vehicles are equipped with various sensors and infrared cameras to collect crop health data and give the initial indications of the diseases. Data analysis and Image analysis techniques are used to predict the health of the crop [62].

• **Smart Greenhouse:** Nowadays, the greenhouse is one of the most popular and fastest-growing farming techniques. In this technique, plants are covered with the green-colored covers or glasses. The IoT based greenhouse enables the farmers to produce multiple crops at a time by maintaining the optimal conditions inside the cover [63]. This empowers the farmer to produce a high yield with better quality.

• **Live Stock Monitoring:** IoT based livestock monitoring enables real-time collection and analysis of animal data, which improves the health and productivity of the animals [64]. Integration of sensor and IoT technology allows the farmers to get the information on trespassing, digestion, grazing behavior, and other vitals of livestock.

Various sensors are used in agriculture to measure soil and environmental factors, integrating these multiple sensor values help in crop management. Several deep learning models have been developed to identify and analyze the crop diseases and weeds at the earliest stages. However, deep learning models should consider environmental factors to analyze the severity level of diseases and weeds. Effective prediction algorithms are required to predict the changes in the environment and market trends. Drones and robots are more common in smart agriculture. The design and development of low-cost intelligent drones and robots are essential to support regular agricultural activities (Monitoring the crops and live stocks). ML algorithms play a vital role in precision feeding by analyzing the grazing behavior of the livestock, which provides the required nutrients at the right time. Efficient algorithms to analyze the behavior patterns of the livestock are also necessary.

#### D. Intelligent Transport System (ITS)

ITS is one of the major applications of IoT, which comprises of different forms of transportation like Road, Air, Sea, and Rail. ITS offers services like Traveler Information, Traffic Management, Electronic Payment, Information Exchange, Freight Management, and V2V communication [65]. These services improve the human being's quality of life in terms of transportation using IoT technology.

• **Traffic Flow Monitoring and Controlling:** It is one of the significant elements of the ITS [66]. An increase in vehicle density in the cities is making traffic monitoring and controlling essential. Monitoring and controlling the congestion using traffic signaling system is the target of

this application. Cameras mounted on the roads are used to gather the congestion data, and present situations are analyzed using ML and AI algorithms [66]. This analyzed information is shared among the travelers.

• **Traveler Information System (TIS):** TIS offers information about the location, traffic regulations, routes, emergency services, safety, and warning measures to the traveler's [67]. TIS is meant to offer more precise information for travelers, thereby reducing the various hurdle in real-time [68]. TIS is a combination of technologies like WWW and GIS [69]. WWW allows travelers to retrieve required data anywhere at any time, and GIS is a system used to store, analyze, and represent the geographical data graphically.

• **Freight Management System (FMS):** IoT enables the gathering of real-time information [70] about the freight along with its transportation chains like the sea, and air. The use of IoT technology in FMS increases operational efficiency, maintenance, reduces electric consumption, and adapt to the environmental changes [71]. The FMS is divided into categories like Commercial Vehicle Operation (CVO) and Advanced Fleet Management System (AFMS). The CVO systems are designed to exchange safety information, electronic certificate supervision, and automated wayside inspection. AFMS is used to automate and reduce the complexities in the freight carriage operations [72].

• **V2V Communication:** The trend is now moving from the use of traditional vehicles to independent vehicles (Autonomous vehicles). Vehicle to Vehicle (V2V) communication plays a key role in the sharing of reliable information between the vehicles and the infrastructure [73]. V2V communication provides 360-degree information about the vehicles and associated infrastructure [74]. IoT alerts the drivers in the case of road hazards, accidents, traffic congestion, and emergencies in V2V communication [75].

Real-time data collection is challenging, inaccurate, and unreliable certain times in ITS; this is due to the high mobility of the vehicles. Real-time accident detection, traffic flow monitoring, and prediction are difficult due to the randomness of occurrence. Effective automatic incident detection and classification using roadside surveillance cameras make ITS more valuable. Several algorithms have been developed, such as SVM, KNN, ARMA, ANN, CNN, and many more to detect and analyze the incidents. However, accuracy is the prime concern in prediction algorithms. Data privacy and security are essential for ITS since people use the various services provided by the government and private organizations which collects the individual's and vehicle's private information such as location details, etc. Therefore, the development of privacy laws and efficient security algorithms need to be implemented to escalate the security feature. This application mainly demands the availability of the communication networks and ruggedness in the IoT system.

### E. Factory Automation & Industry 4.0

Industry 4.0 is the new stage of the industrial revolution that uses real-time data, machine learning algorithms, automation techniques, augmented reality, and interconnectivity technology to improve the overall efficiency of manufacturing [76][77]. Industry 4.0 is also referred to as Industrial IoT (IIoT). Industry 4.0 provides benefits like 1. shorter development cycle, 2. efficient resources utilization, 3. Decentralization to take swift and better decisions, and 4. flexibility in product development [78]. Factory automation and industry 4.0 provide services like plant safety and security (Surveillance), product monitoring, supply management, and quality control.

- **Plant Safety and Security (Surveillance):**

Manufacturing industries face a massive range of dangerous circumstances, which cause severe injuries to the workers and manufacturing plants [77]; this leads to a substantial financial crisis and human loss. IoT based surveillance systems capture possible hazardous situations and analyze the cause of injury to the workers and manufacturing plants. The IoT based surveillance systems are capable enough to inspect the situation like fire accidents, caught in machine, overexertion, and slip & fall accidents.

- **Product Monitoring:** The product development life cycle is shorter in industry 4.0, and it needs to be monitored carefully. The product monitoring using IoT helps in reducing the bottlenecks in the production, optimizing the production process, and prevents the problems before manufacturing [79].

- **Supply Chain Management:** IoT play a fundamental role in supply management by implanting sensor intelligence to the products and machinery. These sensors are connected together and also to the Internet. This advanced connectivity empowers real-time information tracking and sharing of the product information, which elevates the supply management operation [80].

- **Quality Control and Management:** Quality control and management ensures that the product is free from defects. The quality of the products and services is crucial to attaining sustainable economic success [81]. In this, IoT is used to monitor and analyze the quality of the products at critical points and alerts if it finds substandard quality materials [82].

Still, most of the companies are making efforts to adapt to Industry 4.0 but, Industry 5.0 is the future, which enables the collaboration between the people and machines to achieve business milestones. Device management is an essential requirement with the support of middlewares, which ensures the smooth operation of the machines in Industry 4.0. Scalable device management techniques are in demand. Efficient algorithms are required for the robots to understand the surrounding environment and exchange the information efficiently with the other robots and human workers in Industry 5.0, which increases safety in highly complex tasks. Rarely, security threats may lead to

manufacturing and operational disruption, which might lead to a financial crisis. Industry 4.0 requires effective threat detection and system recovery algorithms, along with self-debugging and reconfigurability in the system.

### F. Environmental Monitoring

The growth of population is higher in the urban area in all over the globe which results in urbanization. Air and water pollution is increasing, and meteorological conditions are degraded due to the urbanization [83]. The standard parameters like atmospheric temperature, humidity, pressure, quality of the air and pollutants like carbon dioxide and carbon monoxide are being monitored [84].

- **Forest Fire Detection and Mitigation:** It has been predicted that 80% of the forest loss is due to the forest fire. IoT can be used to build the early fire detection models and alert the fire brigades to take necessary measures [85].

- **Smart Water and Sanitization:** Both developed and developing countries are facing water distribution and sanitation service problems. IoT solutions like digital meters and geographic information systems increase the effective utilization of water and sanitation services [86][87].

- **Outdoor Air Quality Monitoring:** Industrialization and urbanization also lead to air pollution in recent years. Poor quality in the air increases lung, heart and skin diseases. IoT technology allows the evaluating of toxic and flammable gas proportions along with the concentration of air pollutants regularly [88].

- **Natural Disaster Management:** Natural disasters like earthquakes, hurricanes, floods, etc. are more common, which results in loss of habitats, lives, and properties. IoT technology plays an essential role in providing services like early warning systems, immediate responses to the victims, emergency medical care, etc. [89][90].

Sampling, analysis, spatial and temporal errors are more common at the time of measuring the environmental parameters. Decision making is efficient when these errors are eliminated from the dataset. These sensors produce a heterogeneous and enormous volume of data. Therefore, intelligent algorithms are required to clean and preprocess these types of data; this increases the analytical capabilities. Further attempts are essential to implement light-weight algorithms for big data analytics in environmental monitoring. The lifespan of the sensors used in environmental monitoring is reducing due to the meteorological properties. Efforts should be made to increase the durability of such sensors. Along with the above challenges, security issues also need to be considered.

### G. Smart Cities

A smart city aims to advance the urban economy, infrastructure and governance, quality of life (Physical, Mental, and Financial well-being), social and environmental smartness [91][92].

- **Smart Infrastructure:** Infrastructure is the basis for the successful implementation of smart cities. The infrastructure like intelligent electric grids, water grids, and transport infrastructure constitutes towards better and modern societies [93]. Smart infrastructure improves the understanding and control of the operation and optimizes resource utilization in a city.
- **Smart e-Governance:** It enables faster decision making in government organizations and schemes, transparency in the governing agencies, and accessibility of the public services [94].
- **Surveillance:** Surveillance plays a vibrant role in a smart city, which reduces the rate of crimes. In a smart city, surveillance cameras record the individual's activity in the crowded areas to detect and prevent crimes, where public safety officers cannot position; this increases public safety [95].
- **Smart Community:** It provides the essential services like energy and fuel management (green buildings and renewable sources of energy), water management (water quality monitoring, leakage identification, and smart meters), waste management, etc. to the residents [92].

Real-time data processing and analysis, reliable communication, availability, device management are the key enablers for the success of smart cities. Improved Self-learning algorithms are much needed to recognize and analyze the multiple events in real-time videos, detecting the damages in the buildings, real-time traffic flow monitoring, and control. Also, efficient mechanisms are essential to enhance on-demand services such as transportation, vehicle parking, and emergency services. Generally, the IoT system needs to handle video, audio, and signal processing efficiently while maintaining resource constraints.

#### **H. Infrastructure Monitoring**

The purpose of infrastructure monitoring is to collect real-time information related to civil structures such as buildings, bridges, monuments, tunnels, railway tracks, manufacturing and construction process to avoid the risks [96].

- **Oil/ Gas Pipeline Monitoring:** Pipelines are used widely for effective transportation of crude oil and gas; these pipelines need to withstand severe environmental conditions. Monitoring failure in oil and gas pipelines may lead to severe ecological disasters and financial losses. IoT smart sensors are used to monitor the oil/ gas pipelines for cracks and leakages [97].
- **Monitoring of Civil Structures:** Monitoring of civil structures involves various stages such as detecting, locating, identifying the types of damages, and quantifying the severity of the damage. Civil structure monitoring looks for parameters like stress, displacement, cracks in civil structures like bridges, railway tracks, buildings, monuments, etc. [98].

- **Military Assistance and Surveillance:** IoT also plays a major role in military operations to identify the subordinates during the search and rescue operations [99]. Traditional military bases can be converted to smart military bases by deploying the IoT devices; this increases the safety by automating the security screening tests. Intelligent surveillance systems allow armed forces to recognize threats swiftly with higher accuracy.
- **Aircraft Health Monitoring System:** A large number of public and military aircrafts have been surpassed their design lifetime [100]. It is essential to monitor the health of the aircraft structure to increase operational efficiency. Modern aircraft engines comprise of 5000 sensors and generate roughly 10GB of data per second [101]. IoT plays a key role in aircraft structural monitoring by collecting data from the sensors and analyzing them. Analysis results in efficient decision making and communicating the decisions to the aircraft authorities.

Data acquisition and fusion, real-time data processing and analysis, device management, reliable connectivity, and autonomous robots are significant in infrastructure monitoring. Efficient deep learning algorithms need to be developed to detect the damages in real-time. Intelligent robots need to be designed to depute in rescue operations. Efficient sensor data acquisition and fusion techniques are the major requirements, which fasten and simplifies decision making. This application requires massive deployment of IoT devices in the fields. The massive IoT device deployment requires clustering, network availability, and medium to high computational requirements. Efficient device management techniques and light-weight data mining algorithms make device and data management easier in infrastructure monitoring systems. This application also prone to privacy and security attacks; the IoT system should be tampering free.

#### **I. Smart Retail**

Smart retail improves the customer's experience by using communication, information, and augmented reality technologies at the time of shopping. Generally, smart retail shops comprise of RFID tags, interactive displays, recommended systems, and self-cash desks [102].

- **Sensor-Based Stock Maintenance and Tracking:** Smart sensors mounted on the smart shelves allow the customers to find items with no efforts. Generally, RFID tags are attached to the items, which are linked to the store computer system; this allows the store owners to keep track of the items [103].
- **Automated Checkouts:** This allows the customers to pay for their purchases with little or no time. Automated checkouts are relying on linear or barcodes for the identification of the items and their price. Automated checkouts use self-checkout terminals or tunnel systems. In the self-checkout terminal, customers need to scan the items. In the tunnel system, cameras scan the barcodes of the items. Some retail shops allow their customers to scan

TABLE II  
 CLASSIFICATION AND CHARACTERISTICS OF IoT APPLICATIONS

Characteristics	Emerging IoT Application									
	High Data rate Applications					Moderate Data rate Applications			Low Data rate Applications	
	Smart Health	Smart City	Factory Automation/ Industry 4.0	Infrastructure Monitoring	Smart Power and Water Grids	Intelligent Transportation	Environmental Monitoring	Smart Homes	Smart Farming	Smart Retail
Internet Connectivity	WiFi, 3g, 4g, BLE, Zigbee, Z-Wave	WiFi, GSM, Satellite Communication	BLE, Zigbee, WiFi, GSM Satellite Communication	WiFi, GSM Satellite Communication	WiFi, GSM, Satellite Communication	WiFi, GSM, Satellite Communication	WiFi, GSM, Satellite Communication	WiFi, GSM, BLE, Zigbee, Z-Wave	WiFi, GSM, Satellite Communication	RFID, BLE, WiFi
Network Size	Huge (> 10000 Nodes)	Huge (> 10000 Nodes)	Huge (> 10000 Nodes)	Huge (> 10000 Nodes)	Huge (> 10000 Nodes)	Very Large (< 10000 Nodes)	Very Large (< 10000 Nodes)	Very Small (10 Nodes)	Large (Up to 1000 Nodes)	Small (Up to 1000 Nodes)
Analysis	Local & Distributed	Local & Distributed	Local & Distributed	Local & Distributed	Local & Distributed	Local & Distribute	Local & Distributed	Local	Local & Distributions	Local and Distribute
Acceptable Delay	Seconds	Seconds to minutes	Seconds to Days	Seconds	Seconds	Seconds to minutes	Seconds to Minutes	Seconds to Hours	Seconds to 1Day	Hours
Type of Data	Historical, Stream	Stream	Stream	Stream	Stream	Stream	Historical, Stream	Historical	Historical, Stream	Historical
Computing Paradigm	Edge, Cloud	Edge, Fog, Cloud	Edge, Cloud	Edge, Fog, Cloud	Edge, Fog, Cloud	Edge, Fog, Cloud,	Edge, Cloud	Cloud	Edge, Cloud	Edge, Cloud
Technologies Used	SP, IP, AI (DL)	SP, IP, VP, AI (ML), Digital Twin	AI (DL) Robotics	SP, IP, VR, AR, Robotics, Digital twin	SP, Control algorithms, AI	SP, IP, VP, VR, AI	SP, IP, ML	SP, IP, ML	SP, IP, VP, Robotics, Drones	IP, VP, Drones, VR and AR, AI, Robotics
Location awareness & Sharing	✓	✓	✓	✓	✓	✓	✓	-	-	✓
Privacy and Security	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Sensors used	Bio, Pressure, Temperature, SPO2, Chemical Sensors	Environmental, Vehicular and Infrastructure Sensors	Cameras, Temperature, Gas, Fire Sensors	Accelerometer, Displacement, Strain, Inclinometer, Temperature, Laser Sensors	Accelerometer, Displacement, Strain, Inclinometer, Temperature, Laser Sensors, Current and Voltage Sensors, Leak noise localizers and leak noise correlators, Water Level Sensors	Cameras, Ultrasonic, LiDAR, RADAR, GPS, Infrared, Fuel Sensors	Moisture, Pressure, Tilt, Rain, Voltaic Organic Compound sensors, and Geno Sensors	Accelerometer, Temperature, Activity Sensors	Chemical, Temperature, Pressure, Tensiometer, Airflow Sensors	Cameras, Ultrasonic, NFC

SP – Signal Processing, VP – Video Processing, Global System for Mobile Communication (GSM) – 3G/ 4G/ 5G, IP – Image Processing, AI – Artificial Intelligence, VR – Virtual Reality, AR-Augmented Reality, ML-Machine Learning, DL-Deep Learning

the items while customers are moving by using mobile applications [103].

- **Recommender Systems:** Recommender systems suggest the set of items based on the potential interest of the customers. In this, potential customer interests are predicted based on the previous purchases data [104].

In the future, technologies like Virtual Reality (VR), Augmented Reality (AR), and intelligent robots play a crucial role in enhancing the overall customer's shopping experience. These technologies bring new features like virtual trial rooms, provide additional information about the products, search for the best deals of the day, check for the availability of items, and so on. The retailers have access to the customer's data for various purposes; this raises privacy and security issues. The development of advanced methods to analyze the real environment to create a virtual environment are the primary requirement. Efficient algorithms to analyze the customer's mood swings based on the availability of the data are essential.

#### J. Smart Power and Water Grids

The power grid is an interconnection of power producers, distributors, and consumers. Producers gather the electrical energy from various sources like solar, wind, thermal nuclear, and hydroelectric power plants and distribute them among the consumers using power transmission lines. Multiple sensors and control devices are used to monitor the activities of producers, consumers, and distributors; this saves energy, management, and distribution cost [105].

A smart water grid is the networking of water reservoirs (grids and micro-grids), distribution centers, and purification plants, which are equipped with the sensors and control devices to manage and monitor the water distribution system.

• **Power and Water Grid Monitoring:** Power grid monitoring performs activities like power loss monitoring during distribution, load balancing, and metering function to enable safe and efficient power delivery to the customers. Similarly, water grid monitoring comprises monitoring the parameters like pressure, quality of the water, flow, and leakage [106].

• **Management and Distribution:** Power and Water management systems continuously monitor the usage of electrical energy and water resources, respectively. This monitored data is analyzed and used to provide on-demand services to consumers; this ensures adequate levels of power, and water resources are available to the consumers [107].

• **Smart Metering:** It records the electrical energy (voltage levels and power factor) and water usage details automatically and communicates details to the consumer. Data analytics is performed on this data to enhance their operational efficiency potentially [108].

Most of the power grids use the Remote Terminal Unit (RTU) to monitor and control the devices in the smart grid. However, integration of the Phasor Measurement Unit

(PMU), communication, and advanced computation technologies assist the better monitoring of the power grids. Autonomous control of the power grid substation is a primary requirement; to achieve this requirement, efficient predictive algorithms need to be developed to understand the customer's electricity demands. Efficient sensor fusion techniques are required to maintain the water quality and quantity to maintain the water flow in smart water grids. Low power water grid monitoring and management techniques are of high priority. Additionally, efficient prediction models need to be developed to predict the water necessity for the urban areas. In the future, both power and water grids may be integrated; this is due to reducing the energy consumption in households and industries.

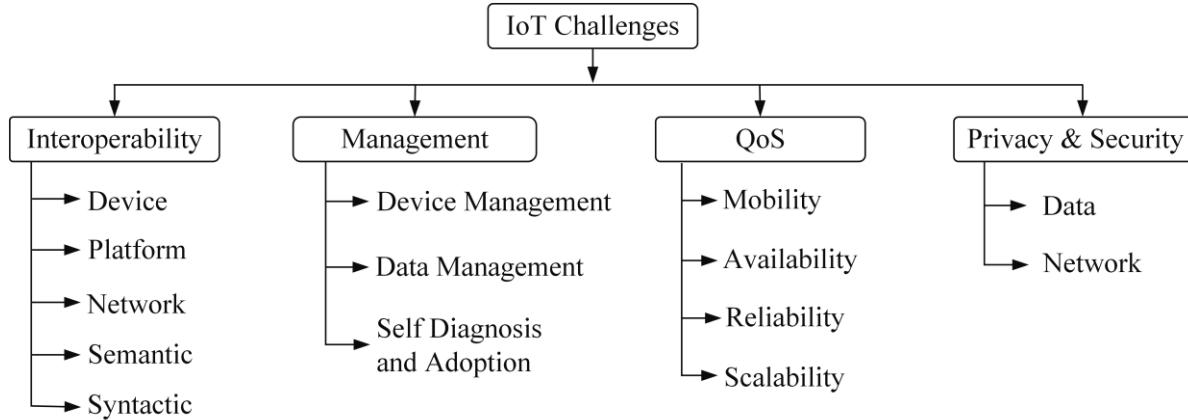
Table II describes the typical characteristics and classification of IoT applications [23][109]. IoT applications are classified based on data rate such as high, moderate, and low data rate applications. In most cases, high data rate applications include signal, image, and video processing technologies. For example, A typical High Definition (HD) video of resolution 1920\*1080 pixels, color depth of 24 bits, and 30 frames per second require a  $1920*1080*24*30 = 186.624$  Mbps data rate. Similarly, to transfer an HD image over a network with 1920\*1080 pixels resolution and color depth of 24 bits requires a data rate of  $1920*1080*24 = 6.221$  Mbps. In moderate data rate IoT applications, the major data sources are images, audio, accelerometer sensor, etc. and related data, which requires preprocessing. For example, an audio signal with a sampling rate of 44100 Hz, and each sample consists of 16 bits per sample; if we consider stereo, two channels will be used. The total data rate  $16*44100*2 = 1.41$  Mbps. Normally, the low data rate application comprises of discrete data values with floating-point representation, and the sampling rate of these values is less than ten samples per second, i.e., the required data rate is  $4*8*10 = 320$  bps. The IoT devices should employ the necessary communication bandwidth, and data preprocessing and data computation techniques, along with the system design requirements listed in Table III.

TABLE III  
REQUIREMENTS OF IOT APPLICATIONS

Type of IoT Application	Requirements			
Low Data rate	Heterogeneity, Security	Self-Diagnosis,	Privacy &	
Moderate Data rate	Heterogeneity, Security,	Self-Diagnosis, Mobility,	Privacy & Device Management,	Scalability
High Data rate	Heterogeneity, Security,	Self-Diagnosis, Mobility, Device & Data Management,	Privacy & Availability,	Scalability

#### V. IOT APPLICATION REQUIREMENTS AND CHALLENGES

The widespread of IoT is because of its benefits like sensing, processing, communicating, and actuating capabilities. The issues like Mobility, Interoperability, Privacy & Security, Storage Management, and Device



**FIGURE 9.** IoT Challenges

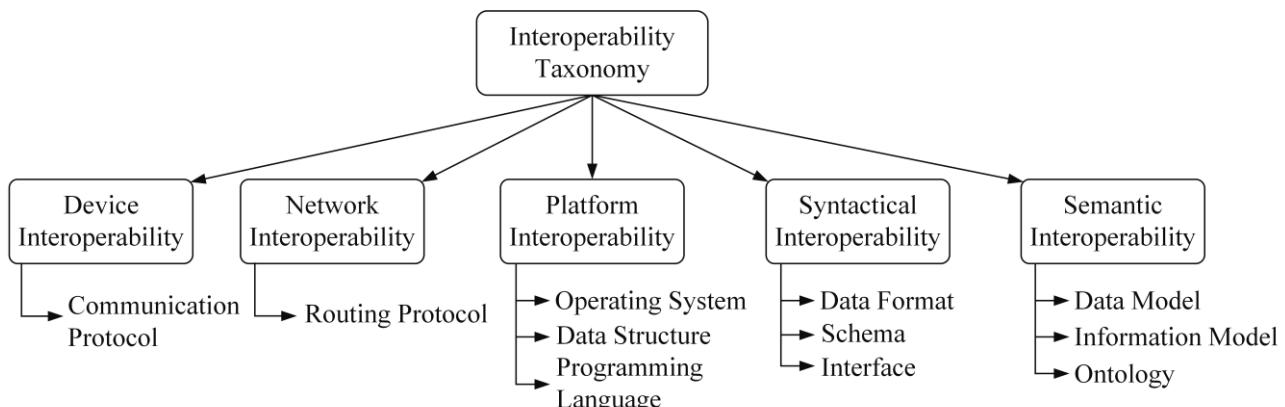
management are rising due to the use of sensors from different vendors, processing platforms, and various communication standards and protocols [110]. To achieve the IoT features such as sensing and actuation, unique identity, ubiquity, communication, intelligence, self-configuration and adaption and effective service delivery following challenges need to be considered. Figure 9 shows the taxonomy of the IoT challenges.

#### A. Interoperability

Interoperability remains a big challenge due to the use of heterogeneous devices in terms of underlying communication standards and protocols, data formats, and technologies [111]. Interoperability is the ability to exchange and make use of information irrespective of the hardware and the software platforms [4]. This issue should be considered in all the layers of the IoT system, i.e., Device interoperability needs to be addressed at the perception layer, network interoperability should be handled at the transport layer, and syntactic and semantic interoperability should be taken care at the application layer of IoT three-layered architecture. Interoperability gains the widespread due to 1. Availability of the heterogeneous devices in the market, 2. The rapid development of IoT applications, and 3. Lack of standards [47]. Approximately 47% of the issues can be solved if an IoT system attains 100% interoperability, which helps IoT services to meet

customer expectations [112]. Figure 10 describes the taxonomy of IoT interoperability. Open Connectivity Foundation (OFC), ETSI (ETSI TR 103, ETSI SR 003, ETSI TS 103), oneM2M (TS-0013), W3C organizations are making their continuous efforts to nullify the interoperability issue [113].

Arne Broring et al. [114] proposed a cloud-based BIG IoT architectural model to address the platform interoperability. This architecture offers communication between the different platforms by creating standard APIs. Semantic web technologies were used by the architecture to enable the interoperability between the IoT applications, platforms, and services. The architecture consists of Bosch, Consorzio per il Sistema Informativo (CSI), Siemens, Verkehr Mobilitat Zukunft, and World sensing platforms. The architecture also contains features like resource registration and discovery, authentication, and authorization. Hasan Derhamy et al. [115] analyzed the existing solutions to achieve protocol interoperability. The authors have proposed a secure, low latency on-demand transparent multi-protocol translator service for Industrial IoT. The proposed translator architecture was based on the service-oriented architecture, and the translators were located at the cloud layer of IoT. The translator uses information like interface design, service description, communication profile, and semantic profile to provide services to the producers and consumers. Shuo Yang et al.



**FIGURE 10.** IoT Interoperability Taxonomy

[116] presented a Tabdoc approach, which enables the semantic interoperability between the IoT users and devices for data accessing and controlling. In this approach, the messages exchanged between the IoT devices and the users were considered as the semantic documents. The proposed method uses the Semantic Extraction Algorithm (SEA) to get complex semantic document components and Semantic Inference Algorithm (SIA) to implement automatic semantic documentation interpretation. SEA provides the mapping between the Tabdoc and non-Tabdoc documents, and SIA performs the automatic cross-context semantic interpretation using the semantic inference.

### B. Management

The application like factory automation/ Industry 4.0 comprises of numerous devices; the status of these devices needs to be monitored for any failures. Time to time software updates needs to be maintained in the devices, etc. These devices also generate an enormous amount of data with different data formats; this needs to be taken care of IoT device and data management. The organizations like ISO and IEC are working jointly to address Big Data issues and formation of standards. These organizations formulated the standard like ISO/ IEC JCT 1/WG 9 for effective data management in IoT [117]. Open Mobile Alliance (OMA) proposed a device management standard to perform device management in the IoT network [118][119].

#### 1) Device Management

Device management allows the network administrator to perform activities like Device Monitoring & Diagnostics, Software Updates & Maintenance, Device, and Configuration & Control. Device management has become challenging due to 1. Rise in the number of devices, 2. Heterogeneous devices, 3. Device mobility, 4. Dynamic topology and 5. Numerous firmware updates.

Gunjae Yoon et al. [120] proposed an architecture to manage IoT sensors and their associated data using a Fog computing. Devices, Fog node, and Operational & Management Server (OMS) are the elements of this architecture. The devices capture the data from the environment and transfer it to the fog node for processing. The fog node acts as a monitoring node, which performs the activities like data management, connection management, and component monitoring. OMS is responsible for collecting the anomalous values with the timestamp from the fog node and send the reports to the administrator. Michael Haus et al. [121] developed an edge-based platform iConfig to administrate the IoT devices in smart cities. This platform addresses issues like registration,

configuration and maintenance of the IoT devices. iConfig platform focuses on the management of the BLE devices and provides services like device status, device functionality and localization of the devices. The edge module uses the centralized queue to synchronize the user actions and collects the maintenance data to identify the beacon status and broken beacons.

#### 2) Data Management

Data management [122] plays a leading role in handling the huge volume, variety, and velocity of data generated by IoT applications [123]. Collecting, storing, processing, analyzing, visualizing the data, actuation, and communication are the different steps in the IoT data management life cycle [124]. Figure 11 depicts the data management life cycle in IoT.

a. Data Collection and Storage: Internet, sensors, satellites, wearable devices, and cameras are the major sources of data [125][126], and these data can be accumulated in the local databases or cloud database. The data may be structured, semi-structured, and unstructured. SQL databases are generally used to store structured data. The most common type of data produced by IoT devices is Semi-structured and Un-structured data. Usually, NoSQL databases are an excellent choice to store IoT data. NoSQL databases have a focus on reading and writing rather than organizing the data. MongoDB, HBase, Hive, Redis, Neo4j, and Cassandra are examples of the NoSQL database [127].

b. Data Processing: Data processing can be done locally (where data is generated), using edge nodes, fog nodes, cloudlets, and cloud platforms [128]. Data filtering and aggregation are crucial operations in data processing [129]. Apache Storm, Spark, Rapid Miner, and High-Performance Computing Cluster (HPCC) are the tools used to process the data in real-time scenarios.

c. Data Analysis: Data Analysis is the critical aspect to bring out the hidden patterns from the processed data [129]. This data is beneficial to control and actuate the decisions. IoT systems should offer both offline and online data analysis methods [124]. The different analysis techniques like streaming, spatial, time series and prescriptive analysis are more common in IoT.

**Streaming:** This technique takes actions by performing the analysis of real-time data using continuous queries.

**Spatial Analysis:** It is a process of analyzing the attributes and their relationship of an entity based on the location information.

**Time Series Analysis:** This technique analyses the data captured at different time intervals.

**Prescriptive Analysis:** It is used to automate the recommendations and make better decisions based on the findings of the analytical models.

#### 3) Self Diagnosis and Adaption

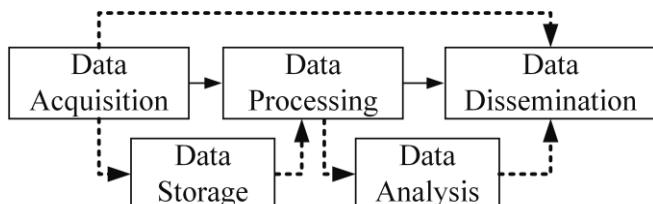


FIGURE 11. Data Management Life Cycle

Self-diagnosis and adaption are the important features of IoT, which identify and minimizes faults in the IoT system. Self-diagnosis is the ability of an IoT device to detect and analyze the glitches in it or in the IoT system. Self-adaption is the ability of the IoT devices/ system to customize itself to the changes in the hardware, software, and resources as and when required [130][131]. For example, self-diagnosis and adaption play vital role in infrastructure monitoring. A dedicated IoT node can be used to monitor the status of the sensor nodes using machine learning algorithms. This runs troubleshooting algorithms automatically to solve the issues.

Junghee Lee et al. [131] proposed a self-diagnosis technique to identify the defective device in the IoT network. The proposed technique deals with the identification of memory, processor, and peripheral faults. Monitoring agents and monitoring programs are the major blocks of the technique. The monitoring agents collect the information of the IoT devices. It keeps on updating the device information to the monitoring program, which is running in the processing unit. The processing units (sophisticated servers or cloud platforms) are rich in resources. The monitoring program analyses the call stack to identify whether the device is working correctly or not. Also, the monitoring program analyses exceptions that occurred during the execution of the code and state of the devices (Hanging state, etc.). Huakun Huang et al. [132] proposed Gaussian Bernoulli restricted Boltzmann Machine (GRBM) based Deep Neural Network with auto Encoder (DAE). The authors have extracted the features from the IoT networks automatically by using the GRBM blocks. The learned features are merged into the encoder neural networks to identify the faulty nodes in the IoT networks. To examine the proposed algorithm, the authors have used the dataset of server machines faults collected by Google data center operators. The size of the dataset is of 41 GB with 77, 776 rows and 13 columns/ attributes. The main features used in the self-diagnosis are timestamp, missing info, machine Id, event type, priority, scheduling class, and task index. The authors have claimed that the fault detection accuracy of the algorithms is 89.2 percent. M. Usman Iftikhar et al. [130] proposed a DeltaIoT self-adaption model for IoT networks. The statistics engine in the model is responsible for gathering and storing the statistics related to the IoT network. The statistics engine provides information about the performance and energy consumption of all IoT devices in the network. The network setting engine communicates with the IoT devices via a gateway to collect and adapt the IoT devices network settings. The authors have tabulated the generic adaption scenarios, which consist of the type of uncertainty, type of adaption, and adaption goals; these generic adaption scenarios play a major role in self-adaption. For example, If the IoT network is facing the uncertainty in wireless interference, transmission power and modifying the path to gateway adaptions are required, and the adaption goals are

reducing the packet loss and energy consumption in the IoT network.

### C. Quality of Service (QoS)

In IoT, QoS manages the network capabilities and resources to provide the required services to IoT users. The use of QoS in IoT results in effective utilization of the services, resources, reduces the network delay and traffic. QoS helps service providers to describe the services based on customer needs [133].

#### 1) Mobility

IoT devices like smartphones, laptops, autonomous cars, robots, and drones are highly mobile in nature [134]. Mobile device management is difficult as compared to static device management in IoT. Mobility is divided into micro and macro mobility. In Micro mobility, devices are roaming from one gateway to another gateway within the network, but in the case of macro mobility, devices are roaming from one network to other networks [135]. Protocols like Mobile IPv4, IPv6, Hierarchical IPv6, Fast mobile IPv6, and proxy mobile IPv6 were introduced to handle the mobility issues effectively [136]. Unfortunately, it is difficult to incorporate these existing mobility standards into IoT due to the large processing time and energy consumption. For example, mobility management plays a vital role in applications like V2V communication, Automatic number plate identification, and traffic diversion, etc. in ITS. Since the vehicles change their location over time and share the information with neighboring vehicles, any information loss during mobility of the vehicles may lead to a dangerous situation or accidents. The parameters like signaling cost, handover latency, packet loss, end-to-end waiting time, and power depletion need to be considered while designing the mobility management mechanisms [137].

Diu Wu et al. [138] presented a software-defined UbiFlow mobility management system for IoT. The system provided features like mobility management, selection of access points, and optimizing the handover mechanism using coordinated distributed controllers. The controllers are scalable, fault-tolerant, and work in resource-constrained environments. Taun Nguyen Gia et al. [135] proposed an energy-efficient and low latency handover mechanism for IoT to support the mobility of the devices in remote locations. The mechanism uses the RSSI, multilevel thresholds, frame injection, and link connection (Bandwidth utilization and the number of nodes connected to each gateway) parameters to attain better handover conclusions. The gateways have identical coverage distance and specifications. The authors claimed that the proposed handover mechanism reduces the latency of switching between the gateways by 10% to 50%. Huai-Lei Fu et al. [139] proposed a group-based mobility management technique in which devices with the same mobility pattern were grouped together, and this was done by using the location database. Location database checks for the mobility pattern of the device when it receives the registration request and determines whether the device can

be merged with the existing mobility group or not. For each group, a group leader is selected to perform the mobility management of the devices. The device is moving from one local network to another; the leader node of the group will perform the registration activity.

## 2) Availability

Time-critical applications like smart health, structural health monitoring, disaster management applications use IoT technology. The term availability in IoT is realized in both software and hardware levels. Software availability provides services to users anywhere and anytime. The hardware availability provides the data required by the application services. For example, the Oil leakage monitoring application provides real-time oil leakage detection services; these services make use of the hardware devices and sensors like GPS, infrared sensors, and cameras to detect the leakages. This application completely relies on the data from these sensors. Here, both hardware and software availability play a significant role.

To ensure data availability during the node failure, Waleed Bin Qaim et al. [140] proposed a distributed hop by hop data replication technique (DRAW). This technique selects the best replica node to maintain a copy of the data items. The technique applies a series of conditions on the degree of replication, availability of the memory in the device, a number of hops, common neighbors of the devices, and previous replicas of the data items. DRAW technique was simulated using the NS-3 tool. Qiang Wei et al. [141] proposed an environmental-based method for modeling the services in IoT. The proposed method uses the WSMO-Lite that characterizes WSDL based web services and RESTful APIs. WSMO-lite is used to model the dynamic context and availability of the services. At first, the method is used to capture the features of the context, and based on this context, the availability of the services is represented in three levels: 1. Two-dimensional properties (Location and Time), 2. Resource and Mobility, and 3. Dynamic behavior of the services.

## 3) Reliability

Reliability is the ability of the components (Software, Hardware, and Network) in the system to perform its essential functions at different conditions and specific intervals of time [142]. Reliability requirements need to be discussed in each layer of the IoT architecture. For example, in factory automation/ Industry 4.0, the devices are working in cooperation to achieve the manufacturing milestone. Here, communication plays a significant role, and loss of connectivity between the devices creates a severe problem in the manufacturing process.

Halah Mommed Al-Kadhim et al. [143] proposed four scenarios to minimize the energy depletion and enhance the reliability at the network layer of the cloud-based IoT network using the Mixed Integer Linear programming (MILP) model. In the route selection process, the failure nodes are identified and replaced with the standby nodes to increase the data delivery rate. The desired reliability level scenario ensured that the selected route would be 99%

reliable. The reliability-based sub-channel scenario reduces the overhead on the reliable routes by employing multiple transmission channels. Reliability-based data compression scenario uses the Sequential Lossless Entropy Compression (S-LEC) technique to minimize the size of the data, which leads to a reduction in transmission power. All these scenarios use the routing path between the IoT device and cloud. Soraya Sinche et al. [144] designed a structural redundancy-based reliability model, and this model is designed to enhance data reliability and communication reliability. Generally, sensors are connected to the gateway using the communication links. Gateways are responsible for connecting the sensors to the central servers, and it is the target of the hackers. In the first scenario, each gateway is linked to the backup gateway (Gateway and the backup gateway uses the Master-Slave Configuration). In the second scenario, the backup link is created for each main link connecting each gateway. Finally, in the third scenario, reliability can be achieved using reliable communication links and gateway redundancy. James Adu Ansre et al. [145] proposed a two-way Dynamic Collaborative Spectrum Sensing Algorithm (DCSSA) to enhance the energy efficiency of data transmission in licensed channels. In this, both source and destination use Channel State Information (CSI) to improve the reliable message transmission at higher data rates. The use of significant CSI results in the sensing of channels at low SNR regimes and multiple secondary searches for the availability of the licensed channels collaboratively to reduce energy consumption. The secondary users (SU's) finds the availability of the spectrum channel by sensing the busy tone among the primary users and listening for an idle channel. To ensure data transmission reliability, the SU's share the information among them and maintains records of the packets sent by them and its time.

## 4) Scalability

Growth in the number of IoT devices and their associated data results in scalability issues. [146]. IoT scalability is characterized as a system's ability to manage the increasing number of devices and their information. The technologies like cloud, fog, and edge computing can be used to tackle scalability issues in IoT. For example, network devices and related data are increasing in the application like smart cities. IoT infrastructure should accept, process, and respond to the requests from the devices and users without much delay (Average waiting time should be reduced).

Hui Guo et al. [147] projected the transparent computing-based IoT architecture to develop scalable and manageable IoT applications. This architecture enables the centralized supervision of resources like Operating Systems (OS), services, application data, and on-demand services to run on the devices. This architecture consists of end-users, edge network, centralized network, service & storage, and management layers. An end-user layer is a collection of IoT devices. The edge network contains the high-performance routers and small-scale servers to perform several tasks. The centralized network layer bridges the edge, service, and

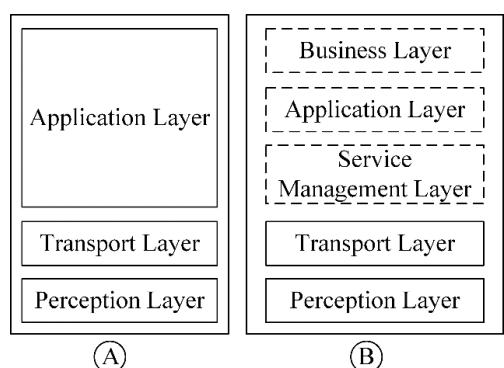
storage layer using different communication standards. The service & storage layer is accountable for data storage and providing the application services to the customers based on the necessities. The management layer is liable for managing all the application services and assigning the jobs to the control server. Janice Canedo et al. [148] added the scalability feature to the gateway by using parallel computing. In this architecture, the devices that exist in the perception layer are connected to the gateway in the transport layer. To create scalable architecture, the authors presented the concept of the scalable gateway to handle data from an enormous number of devices. The Jetson TX1 is used as the gateway, and twelve Arduino Uno devices are connected to it. The gateway uses parallel computing techniques to handle the data coming from diverse devices effectively. The authors claimed that the use of high-performance gateways with parallel computing techniques increases the performance (data collection, processing, and analysis) and allows us to connect a greater number of devices. Aleksandra Tiurlikova et al. [149] proposed a technique to boost the scalability of the LoRaWAN by selecting the efficient Spreading Factor (SF). SF is used to specify the chirp rate, and LoRa SF can choose from the SF7 to SF12. Lower the SF higher will be the data rate. In this approach, the selection of SF is based on the PER 0.01 criterion and PDR. PDR is the ratio of the number of packets received by the gateway and the number of packets sent by the devices. This approach enables the use of a greater number of devices by selecting the appropriate SF.

#### D. Security and Privacy

IoT applications are more common in society and most likely subjected to diverse types of security attacks [150][151]. Minor changes to IoT data may lead to serious situations. For example, the patient data in smart healthcare comprises of personal and highly sensitive health data; privacy and security of the IoT data need to be maintained. Section IX provides a detailed description of privacy and security issues.

### V. IOT ARCHITECTURE

Several architectures have been introduced to understand the concept of IoT more precisely; among them most prominently used architectures are three and five-layered



**FIGURE 12.** IoT Layered Architecture Three Layered & B. Five Layered Architecture)

architectures. Three and five-layered architectures are depicted in Figure 12 [152][153].

#### 1) Perception Layer

The perception layer is a set of objects. Objects serve as an intermediate between the environment and the digital world by using sensors [154]. The primary motive of this layer is to capture data from the environment using various sensors such as temperature & humidity, gas, lights, GPS, camera, etc. depending upon the application requirements. The researcher's focus is on unique object identification, object management, and security in this layer.

#### 2) Transport Layer

This layer aims at connecting the objects and sharing information among the connected objects in a secure way. Wired or wireless communication standards like Ethernet, WiFi, Wi-MAX, ZigBee, and BLE are adapted to share the information. The reduction of energy consumption in the network, enabling quality of service (QoS), adaptation to dynamic topologies are some of the issues need to be addressed at this layer [153].

#### 3) Service Management Layer

This layer is also termed as a middleware layer, which facilitates the use of heterogeneous devices in IoT applications. This layer also processes the raw data recorded by the objects in the perception layer. The typical characteristics of the data captured are of enormous volume and varied [153].

#### 4) Application Layer

This layer is responsible for providing application-specific services to end-users. For example, Smart Homes Application offers services such as home automation, intelligent gardening, surveillance, monitoring of older people, and others.

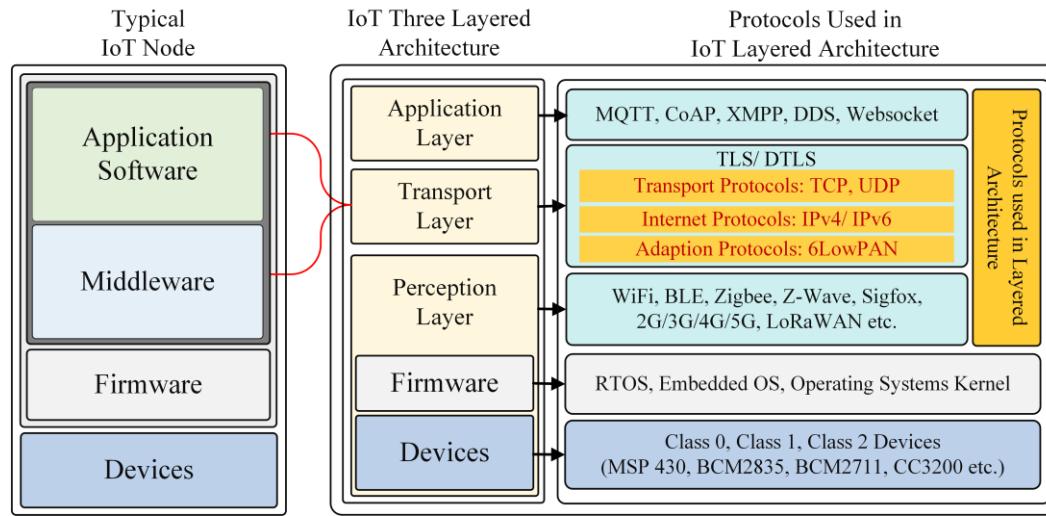
#### 5) Business Layer

This layer supervises the IoT system's operations and its various services; this creates business models, flow charts, and graphs based on the raw data acquired from other layers. This layer is responsible for analyzing, monitoring, and evaluating the IoT system and its related elements. Decision-making is one of the business layer's primary activity [11][7].

Figure 13 illustrates the mapping of IoT node (left side) to IoT three-layered architecture (center) and corresponding protocols used in communication. Any IoT node comprises of devices (SoCs, Processors, etc.), firmware, middleware, and application software.

**Devices:** Devices are SoCs, Processors with inbuilt interfaces, which can capture, process, send, and receive the data. Also, the devices execute the commands with the help of human intervention or automatic. The size of the deployed IoT devices may vary from a small cuff link sized button to the big industrial machines.

**Firmware:** It is a fixed set of instructions on the devices, which facilitates the users/ programmers to access the lower-



**FIGURE 13.** Relationship between Device, Operating System and IoT Protocol Stack

level hardware by locating the operating system's kernel. A significant number of real-time, embedded, and other operating systems are available. RIOT, TinyOS, LiteOS, ContikiOS, Nut/OS, and μC/OS are the embedded OS's. The OS's like OpenTag, FreeRTOS, VxWorks, μ-velocity, and ErikaEnterprise are the few real-time OS's. The OS's like Tizen, Ubuntu core, Android Things, Raspbian, etc. come under the category of other OS's, which are specifically designed for IoT applications. Section X provides the meticulous analysis of operating systems for IoT and their protocol stack.

**Middleware:** It is a software, usually comes with the operating system, which provides specific services in the form of a library. Sometimes these libraries can be modified according to the functional requirements and added as new external libraries. It is used to connect two or more applications. Generally, applications use communication stack to establish a communication, which is a part of middlewares in the operating system either as an inbuilt function or additional library function. In IoT, the perception layer uses different standards like IEEE 802.11, IEEE 802.15.4, RFID/ NFC, BLE4.0/ BLE5.0, Zigbee/ Zigbee Pro/ Zigbee 3.0/ Zigbee NAN, LTE-A, Z-Wave, EPCglobal, MiWi, WirelessHART, Thread, Ant+, Wi-SUN, LiFi, Ingenu, Insteon, Telensa, Wireless M-Bus, NB-Fi, GSM, NB-IoT, Sigfox, LoRa, DASH7, Weightless, etc. Internet protocols like IPv4 or IPv6 are used in the internet layer; these protocols are responsible for assigning the unique identity called IP address to the devices and transmission of packets over the internet. 6LowPAN is also called as IPv6 for low power wireless devices like BLE, Zigbee, and Thread. 6LowPAN performs encapsulation and header compression in low power wireless networks. The transport layer in IoT uses protocols like TCP and UDP. The security protocols like TLS and DTLS are used by the TCP and UDP, respectively, to protect the communication channels. The application layer protocols like MQTT, Secure-MQTT, CoAP, XMPP, DDS,

WebSocket, AMQP, Lightweight M2M, Simple Text Oriented Protocol, Simple Media Control Protocol, Simple Sensor Interface Protocol, RESTful, and HTTP are used in the application layer.

**Application Software:** It is a set of programs working in cooperation to meet the user needs. The objectives of the application software is to collect the data from the sensors, preprocess the data, and extract the value. This value may either be used to control the actuators or input to any other service.

Bahijar Vogel et al. [155] proposed an Open IoT architecture to address the sustainable necessities of IoT. The architecture supports flexibility, extensibility, and customizability. Flexibility allows users to perform various settings and operations with a shorter delay. Robustness, easiness, and cost-effectiveness are the properties of flexibility. Extensibility offers the developers to enhance the architecture with minimum cost upgrades. Modularity, Adaptability, and compatibility are the properties of the extensibility. Customizability allows the user to modify the features of the system. Customizability has properties like easiness and cost-effectiveness. Simone Cirani et al. [156] proposed a scalable and self-configurable architecture for IoT to implement automated services and resource discovery features. In this architecture, the Zeroconf mechanism (No prior configuration information is needed to discover new services and resources) is used to discover the services and resources within the local network. Service discover protocol is used to find out the services and resources based on the Universal Resource Identifier (URI) and the number of nodes between the client and the server. P. Hu et al. [157] proposed a Software-Defined Device-based IoT System architecture. In this architecture, a centralized controller is used to manage the devices in the IoT network. This architecture aims at providing features like device discovery services, sharing, and reusing of the device resources. To enable these features, the authors

introduced the Software Defined Device layer between the transport layer and application layer. Yi Xu et al. [158] proposed a scalable Cloud-Edge-Beneath based architecture for IoT. This architecture consists of beneath, edge, and cloud layer. The beneath layer comprises of the physical and sensor platform layer. The physical layer is a collection of devices and descriptions about themselves. The authors have used XML based device description language to define the specification of the devices. The sensor platform in this layer is responsible for device configuration, booting, and control. Also, this layer performs data acquisition and caching. The edge layer provides the service discovery and configuration mechanism using Open Services Gateway initiative (OSGi) framework. The cloud layer is liable for the deployment and running of the application-specific services. Chayan Srkar et al. [159] proposed a scalable distributed architecture for IoT. This architecture proposed an approach to handle issues like scalability, heterogeneity, interoperability, and security. The architecture consists of Service Layer (SL), Virtual Object Layer (VOL), and Composite Virtual Object Layer (CVOL). SL is responsible for the creation and management of IoT services. The virtualization of the physical objects is carried out in this layer, and this contains the characteristics and capabilities of the physical objects. Virtual nodes in the CVOL acts as a coordinator and also performs the smart scheduling of the tasks. A cross layer security management module is defined by the architecture to address privacy and security issues.

## VI. COMMUNICATION TECHNOLOGIES FOR IOT

Communication plays a significant role in the sharing of raw and analyzed data between the objects. The role of communication in IoT is to enable the seamless connectivity between the two endpoints anytime, at anyplace [160]. IoT comprises of Edge to Edge, Edge to Gateway, Edge to Cloud, and Backend Data Sharing models [161].

### A. Communication Models

#### 1) Edge to Edge Communication Model (E2EC)

In this model, devices communicate directly with other devices. This communication model takes advantage of wired or wireless communication standards. Figure 14 depicts the working of the Edge to Edge Communication model. This model plays a vital role in indoor applications like home automation, industrial automation, and others [161][162].

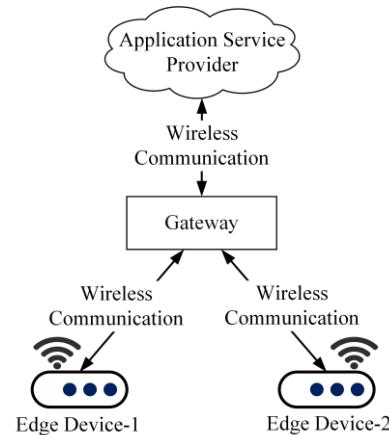


**FIGURE 14. E2EC Model**

#### 2) Edge to Gateway Communication Model (E2GCM)

In this model, the gateways are loaded with the application layer software and serves as an intermediate between the devices and application service providers. Figure 15

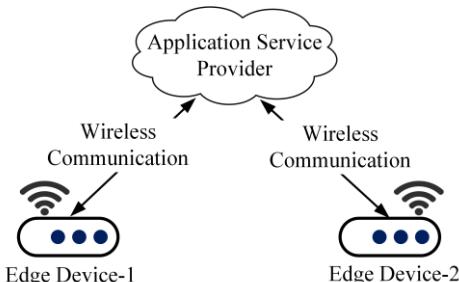
illustrates the working of the Device to Gateway Communication Model [161]. This mode plays a key role in applications like smart homes, V2V communications, livestock monitoring, sports and fitness, and others.



**FIGURE 15. E2GC Model**

#### 3) Edge to Cloud Communication Model (E2CC)

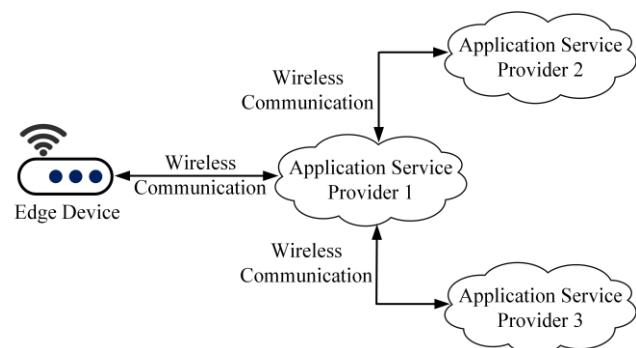
In this communication standard, devices are directly connected to the cloud. Devices read and write data from and to the cloud, respectively. This communication model is given in Figure 16 [161]. Applications like Smart Grids, FMS, Connected Labs, etc. use this communication model.



**FIGURE 16. E2CC Model**

#### 4) Back end Data Sharing Model (BDS)

This model allows the trusted third parties to obtain the sensory data from the cloud for aggregation and analysis. The efficient implementation of this model enables the trusted third parties to transfer their data between the multiple IoT services. Figure 17 represents the concept of the Backend Data Sharing model [163].



**FIGURE 17. BDS Model**

Generally, these communication models use communication standards like RFID, WiFi, Bluetooth, ZigBee, Z-wave, LoRa, Sigfox, NB-IoT, 5G, etc.

### **B. Communication Standards**

IoT devices can use wired or wireless communication standards. The use of wireless communication standards is increasing day by day in society. Therefore, this section provides the short and long-range wireless communication standards used in IoT. At present, the predominant organizations like IEEE (IEEE 802.11, IEEE 802.15.1, IEEE 802.15.4, IEEE 802.16, etc.) [164], and Internet Engineering Task Force (IPv6 over Networks of Resource-Constrained Nodes (6lo), Constrained RESTful Environments (CORE), IP Wireless Access in Vehicular Environments (IPWAVE), IPv6 over Low Power Wide Area Networks (IpWAN), Software Updates for Internet of Things (SUIT), etc.) [165] are providing the communication standards for IoT.

#### **1) SHORT-RANGE COMMUNICATION STANDARDS**

Generally, the perception layer is the collection of sensors that aims to collect and transmit the physical parameters to the gateway using short range communication standards. Most of the sensors at the perception layer are battery operated and use short-range, low-power wireless communication standards like RFID, BLE, WiFi, Zigbee, Z-wave, Thread, Light-Fidelity (LiFi), and Wireless HART to transmit the data. The use of these communication standards will increase the lifetime of the sensors.

##### **a. RFID**

RFID is a wireless communication technology designed for automatic identification, tracking, and collection of data from the objects [166]. RFID is a combination of RFID tags and readers. RFID-tag is a microchip used to store the identification information; this identification information is used to track the objects later. RFID-reader fetches the data when an RFID-tag comes into contact. RFID-reader can be fixed or mobile [167]. The communication range of RFID exists between 3 to 90 meters. This type of communication is beneficial in applications like production monitoring and control, supply chain management, etc. [168].

##### **b. WiFi**

In WiFi, transmission, and reception of data over a short distance are using 802.11 radio technology. Various classes of WiFi currently exist in the market, such as 802.11a, 802.11b, 802.11g, and 802.11n. The WiFi coverage range is about 50 meters, and the data rate reaches up to 11 Mbps [169]. WiFi is ubiquitous in smart homes, buildings, etc.

##### **c. Bluetooth Low Energy (BLE)**

It is also known as the Bluetooth Smart, built for short-range communication. Compared to classical Bluetooth technology, BLE consumes less power and provides high data rates for audio and video applications. It uses a 2.4 GHz ISM frequency band and can reach up to 100 meters. BLE provides the data rate up to 1Mbps. Applications like smart

homes, smart offices, smart shopping, etc. are benefited from BLE [170].

##### **d. ZigBee**

ZigBee is a communication standard developed by the ZigBee Alliance to carry small chunks of data. ZigBee is of low power and low cost, which consumes 1mW most of the time. The operating range of the ZigBee is up to 150 meters. The bandwidth of the channel is 1 MHz, and its data transfer rate is about 250kbps. Zigbee is very useful in applications like smart homes, industry 4.0, etc. [171].

##### **e. Z-Wave**

This communication standard establishes communication over a short distance, and Zensys developed it. Z-Wave forms the mesh networks to enable the device to device communication. Z-Wave reduces the latency in transmitting the smaller packets at the rate of 100kbps. The communication range of this protocol is about 30 meters [172].

##### **f. Light-Fidelity (LiFi)**

LiFi is a wireless communication standard, which uses the visible light spectrum (i.e., Light Emitting Diode Bulbs) to transmit the data. Sometimes, this standard is also alluded as Visible Light Communication (VLC). LiFi supports greater mobility and multiuser access. It provides a hundred times faster data rate than WiFi (i.e., 1 Gbps). However, the range of communication is very limited. Smart homes and Industry 4.0, Virtual and augmented reality applications get more benefit from this standard [173][174].

##### **g. Wireless HART**

Wireless HART is an extended version of wired HART protocol, and it is commonly used in process automation. This standard supports time synchronization, self-organization, and self-healing mesh topology in industrial networks. Wireless HART operates using a 2.4 GHz ISM band. The data rate of the protocol is about 250 Kbps and covers a radius of 230 meters. This protocol is best suited for the Factory automation/ Industry 4.0 applications [175].

#### **2) LONG-RANGE COMMUNICATION STANDARDS**

Long-range communication standards are commonly used by the gateways to share the data across the remote locations. LoRaWAN, NarrowBand-IoT, Sigfox are generally used for long-distance communication, and the use of 5G technology is increasing slowly. These standards provide features like high data rate, less power consumption, and more extensive coverage area (communication range).

##### **a. LoRaWAN**

LoRaWAN is a Low Power Wide Area Network (LPWAN) technology designed to connect battery-operated devices wirelessly [176]. LoRaWAN uses the LoRa spread spectrum modulation technique at the physical layer to achieves the characteristics of the LPWAN. Chirp Spread Spectrum (As time increases, frequency also increases linearly) modulation technique is used by the LoRa. The maximum admissible

data rate in LoRaWAN is about 50 kbps with an area coverage of 5 km (urban) to 20 km (rural) [177].

#### b. Narrow Band (NB)-IoT

NB-IoT is an LPWAN cellular radio technology developed by the 3GPP to cover a wide area, and it uses the LTE standard [178]. The NB-IoT aims to enable low-power, low-cost, and a wide range of communication in the IoT environment. Per each cell, NB-IoT accommodates up to 50,000 devices, and the data rate is ranging from 230 kbps (Downlink) – 250 Kbps (Uplink) with an average area coverage of 35 km [179]. NB-IoT based devices have a better life for up to ten years. This technology is very flexible with the 2G, 3G, and 4G technologies and eliminates the need for gateway (Directly connected to the BS).

#### c. NB-Fi

NB-Fi is an LPWAN connectivity solution developed by WAVIoT. This standard uses 433MHz, 868MHz, and 915MHz ISM bands. Single BS can manage up to 2000, 000 smart devices, which fall in its range; this provides the high scalability in NB-Fi networks. Artificial intelligence is used in NB-Fi networks efficiently in order to enable the self-management and optimization network characteristics. The data rate of this standard is about 25 kbps, and it can cover up to 16 km in urban and 50 km in the rural area. NB-Fi plays a vital role in the application like smart agriculture and smart retail [180][181].

#### d. Sigfox

Sigfox is an LPWAN communication technology used to deploy IoT networks, which operates at 433 MHz in Asia, 868 MHz in Europe, and 915MHz frequency in North America. Sigfox coverage area ranges from 10 km (rural) to 40 km (urban) with a data rate of 100 bps [177]. Sigfox uses the Ultra Narrow Band (UNB) modulation techniques [182], which reduces the consumption of energy and increases the receiver's sensitivity [183]. A Sigfox device can send only 140 messages per day with a payload length of 12 bytes. A Sigfox device survives up to a decade [184].

#### e. INGENU

INGENU is a proprietary LPWAN communication

standard, and it is also known as the On-Ramp Wireless standard. It uses the 2.4GHz free ISM bands for communication and Random Phase Multiple Access (RPMA) technology; this enables robust and reliable communication. RPMA obeys the specifications of IEEE 802.15.4k. The communication range of INGENU is about 15 km, and the data rate is about 20kbps [185].

#### f. Telensa

Telensa is an LPWAN standard, which uses an ultra-narrowband transmission mechanism. The PLANet is a central management system used by Telensa to manage the end-to-end operations. PLANet uses an automated fault detection system, which reduces the energy consumption and manages up to 5000 nodes. It operates using 868MHz and 915MHz unlicensed ISM bands. The communication range of this communication standard is of about 2 km in urban and 4 km in rural areas. The estimated lifetime of a Telensa node is up to 20 years. As a real-time example, Telensa was used to deploy 369,000 street-lights in the city of Georgia. This communication standard is suited for smart city applications [186].

#### g. 5G (Fifth Generation) and beyond Wireless Communication

5G is the foundation for realizing the full potential of IoT. It is a fifth-generation cellular network, which provides seamless connectivity between the more significant number of devices with high data rates and greater mobility [187]. The standardization of 5G technology started in the year 2016 by International Telecommunication Union (ITU) and 3GPP [13]. 5G offers high data rates and spectral efficiency by using millimeter wave (mmWave) communication technology. In 5G, Massive MIMO enhances the coverage area and user experience. The operating frequency of the 5G technology is between 350 MHz to 100 GHz, and the latency is about 1ms. 5G communication technology can reach the data rate up to 20 Gbps and supports different scenarios like indoor, rural, urban, and suburban scenarios, etc. [187][166].

Table IV summarizes the characteristics of different communication technologies [188].

TABLE IV  
CHARACTERISTICS OF COMMUNICATION TECHNOLOGIES

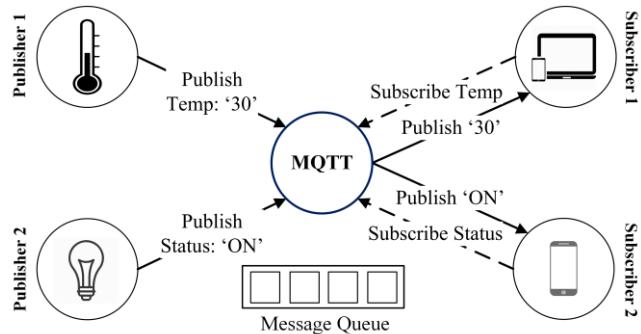
Characteristics	WPAN	WLAN	WMAN	WWAN
Communication Standards	Bluetooth, BLE, Zigbee, Thread Z-Wave, EnOcean, ANT <sup>+</sup>	WiFi, ETSI HiperLAN, Wireless HART	WiMAX, ETSI HiperMAN, Zigbee-NAN, NWave, LoRa, Sigfox, NB-IoT, Wi-SUN	3G/4G/5G, LPWAN, WRAN, NB-IoT, Satellite Communications, Cognitive Radio, INGENU, Telensa
Communication Range	10 – 150 Meters (Short Range)	150-1000 Meters (Short/ Medium Range)	2 – 50 km (Medium Range)	Up to 100 km (Long Range)
Frequency	2.4GHz	2.4 – 5.9 GHz	10 – 66 GHz	900 – 1800 MHz
Maximum Data Rate	1-2 Mbps	11-54 Mbps	Up to 268 Mbps	10 Kbps – 2.4 Mbps
IoT Applications	Home Automation, Smart Agriculture, Healthcare, Structural Health Monitoring	Home Automation, Smart Agriculture, Healthcare	Smart Cities Multimedia, Digital TV Broadcasting	Smart Cities, Intelligent Transportation

## VII. IOT APPLICATION LAYER PROTOCOLS

The application layer is the topmost layer in IoT layered architecture and delivers the application services to the users by initiating actual communication between the IoT devices [189]. This layer is liable for formatting and presenting the data to the end-users by using the application layer protocols. MQTT, CoAP, XMPP, DDS, and RESTful HTTP protocols are the most commonly used in IoT applications [190]. This section provides the detailed analysis of the above listed application layer protocols. OASIS defines the application standards like MQTT, AMQP, and COE[191]. The oneM2M defines the protocol bindings for CoAP (TS-0008) [192], HTTP (TS-0009) [193] and MQTT (TS-0010) [194][195].

### A. MQTT

MQTT is a lightweight application layer protocol used to transmit messages. Figure 18 depicts the MQTT messaging model. MQTT operates over the transport control (TCP/IP) protocol and uses the Publisher/Subscriber messaging pattern [196]. The publisher, broker, and subscriber are part of the MQTT messaging model. The publisher's role is to publish the data to the broker instead of sending messages directly to the subscriber. To receive messages from the broker, the subscriber must register with the broker. The broker administers all the message flow between the publishers and subscribers using the concept of topics. MQTT protocol ensures reliable message delivery in resource-restricted environments [197].



**FIGURE 18. MQTT Messaging Model**

MQTT offers various QoS [198][199] to perform better under the unstable networking conditions, as listed below.

1. Level 0 (QoS - 0): This makes sure that the messages are sent to the broker almost once, and it does not require any

acknowledgment [198][200].

2. Level 1 (QoS - 1): This makes sure that the messages are sent at least once to the broker, but it could be sent more than once also [198][200].
3. Level 2 (QoS - 2): This ensures that the messages are sent only once to the broker. This uses the four-way handshake and which affects the overall performance of the protocol [198][200].

The exchange of the data between the MQTT devices is based on the predefined format depicted in Figure 19 [201][202][203].

Control Header	Packet Length	Variable Length Header	Payload
1 Byte	1 – 4 Bytes	0 – N Bytes	0 – N Bytes
Fixed Header Present Always	→	Not Always Present	Size Depends on Control Header Type

**FIGURE 19. MQTT Packet Structure**

1. Control Header: The control header is of 8 bits field, and it is divided into two subfields, such as Packet Type and Flags, respectively. Each subfield is of 4 bits. Table V provides the type of packets used by MQTT. MQTT comprises of Duplicate (DUP), QoS, and Retain flags [201][91].
2. Packet Length: Signifies the number of bytes left in the existing message, including data in the variable length header and payload [90] [91].
3. Variable Length Header: This field is an optional MQTT message. It is used to carry additional control information [201][91].
4. Payload: Payload is the last field of MQTT, which contains the application data [201][91].

### Analysis

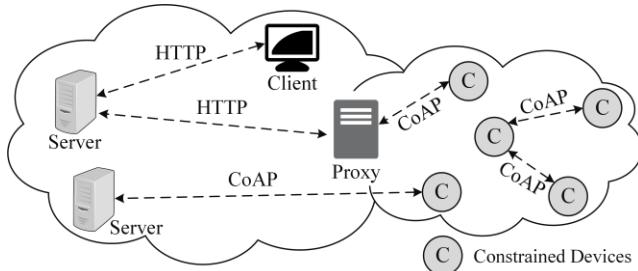
- It is a lightweight protocol suitable for the constrained network of devices.
- It provides the flexibility to choose the level of QoS.
- It provides excellent services, even if the internet connection is unreliable.
- The security of the data can be enhanced using data encryption algorithms.
- Power consumption is more due to the TCP based connection.
- Limited support for transmitting high-resolution images, audio, and videos.

**TABLE V**  
**PACKET TYPES IN MQTT**

Packet Type	Value	Flow Direction	Description
RESERVED	0	Forbidden	-
CONNECT	1	Publisher to Broker	Publisher request to connect to the broker
CONNACK	2	Broker to Publisher	Acknowledgment to Publisher from Broker
PUBLISH	3	Publisher to Broker	Publishing the message by Publisher
SUBSCRIBE	8	Subscriber to Broker	Subscriber request to connect to the broker
SUBACK	9	Broker to Subscriber	Acknowledgment to Subscriber from Broker
UNSUBSCRIBE	10	Subscriber to Broker	Unsubscribe request to the broker from the Subscriber
UNSUBACK	11	Broker to Subscriber	Unsubscribe acknowledgment from Broker to the Subscriber

### B. CoAP

CoAP is an application layer protocol for power and memory restricted IoT devices. CoAP implemented on the top of the User Datagram Protocol (UDP) [204]. The architecture of this protocol is illustrated in Figure 20. CoAP uses the Client-Server communication architecture model. The CoAP uses the RESTful APIs, which reduces communication overhead [204]. It uses HTTP methods like - GET, POST, PUT, and DELETE to empower resource-oriented services in Client-Server applications [205].



**FIGURE 20.** CoAP Messaging Model

Figure 21 shows the CoAP messaging format [206][207]. The Version field is of about 2 bits, which indicates the version number of the CoAP message. CoAP uses the message types such as Confirmable (0), Non-Confirmable (1), Acknowledgment (2) and Reset (3) [208].

Version 2 Bits	Type 2 Bits	Token Length 4 Bits	Code 8 Bits	Message Id 8 Bits
Token (8 Bits)				
Options (If Any)				
Pay Load (If Any)				

**FIGURE 21.** CoAP Messaging Format

1. Confirmable (CON): Sender will get the notification from the receiver/ server regarding the reception of the messages.
2. Non-Confirmable (NON): Sender will not get any notification from the receiver/server.
3. Acknowledgement (ACK): This is a replay message to the CON message.
4. Reset: Indicates the confirmable message was received with some missing context. This condition usually occurs when there is a rebooting and no idea on about state of the receiving node.

The code represents the message type of the request and response messages. CoAP uses the GET (1), POST (2), PUT (3), and DELETE (4) message types to retrieve, update, create and delete the messages respectively [206]. The message ID is used to indicate the type of message generated by the sender of the CoAP node. The token field is used to differentiate concurrent requests. The token generated by the CoAP node should be unique for the current source and destination pair.

### Analysis

- The communication cycle is faster due to its smaller packet size.

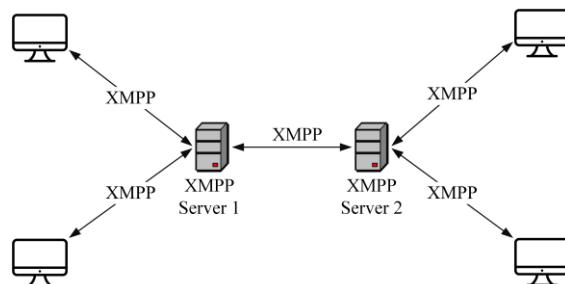
- It has lower latency.
- The use of data encryption results in better security of the data during transmission.
- Since it uses UDP, there is no guarantee in the message delivery.

### C. XMPP

The XMPP protocol uses Decentralized Client-Server (DCS) for instant messaging. DCS assists in separating the client and server developers. The client developer's job is to increase the user experience, and the server developer's job is to provide scalability & reliability [209]. XMPP supports both publisher/subscriber (asynchronous) and request/response (synchronous) messaging patterns. The usage of the messaging pattern is up to the IoT application developer. The client, server, and gateway are the components of this architecture [210].

**Clients:** Clients are the one who initiates the data stream and communicate with the servers.

**Servers:** It enables the implementation of security features like authentication and encryption algorithms to provide secure application services.



**FIGURE 22.** XMPP Messaging Model

Authorized client and server connections are also managed by the servers. XMPP uses the XML sections to create a communication link between the client and the server. An XML stream is split into three subsections, such as message, presence, and information/query (iq). Figure 22 depicts the XMPP messaging model [211] and Figure 23 highlights the XML sections [212][213] used by the XMPP protocol respectively.

```
<stream>
<presence>
  <show/>
</presence>
<message to = 'x'>
  <body/>
</message>
<iq to = 'y'>
  <query/>
</iq>
</stream>
```

**FIGURE 23.** XMPP Message Stanza

The presence section in XMPP is used to provide client, server, and users status. `</show>`, `</status>` and `</priority>` are the optional elements of the presence section. Non-human-readable characters are part of `</show>` element. This section contains information about the available resources. Only one `</show>` element allowed inside the presence section. `</status>` element is used along with the `</show>` element, which includes XML characters. The `</show>` element provides a detailed definition of the state. The `</priority>` element contains the non-human-readable XML characters, which indicates the level of resources priority. The priority value must be between -128 to +127. If no priority value is provided, then by default, the server considered the value as zero [95][96][97].

A message section comprised of the title of the message and its content. This section composed of any one of the elements such as `</subject>`, `</body>` and `</thread>`. The `</subject>` element contains the human-readable XML characters, it specifies the message topic. The `</body>` element contains the content of the actual message to be carried. `</thread>` element is used as an identifier, which contains the non-human-readable XML characters to track the conversation between the entities. Iq section provides the request and response mechanism to communicate [95] [96] [97].

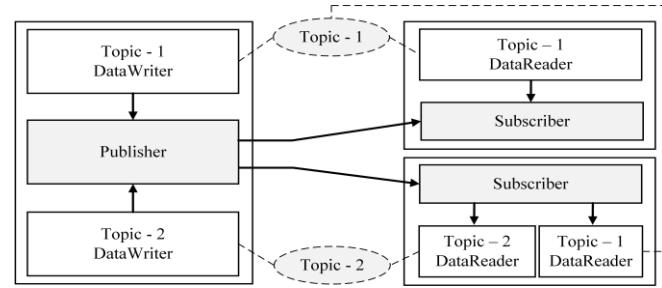
#### Analysis

- Best suited for carrying text messages (Chatting application).
- The protocol is Decentralized, Extensible, and adaptable nature.
- Excess traffic can be reduced by eliminating the redundant data
- Power consumption is more due to the use of XML messaging formats.
- Enabling QoS increases the performance of the protocol.

#### D. DDS

DDS was designed and implemented by the Object Management Group, and it utilizes the publisher/subscriber messaging model [214][172]. Publishers, Data Writers, Subscribers, Data Readers, and Topics are the components of DDS architecture, and it is depicted in Figure 24.

The job of the Publisher is to distribute the data. The function of the subscriber is to receive data from publishers and makes it available for the application. Publishers and subscribers are attached to the data writers and data readers,



**FIGURE 24. DDS Messaging Model**

respectively. The Data Writer and Reader are attached to the Topic. Since DDS supports QoS and reliability, this protocol is the right choice for M2M applications.

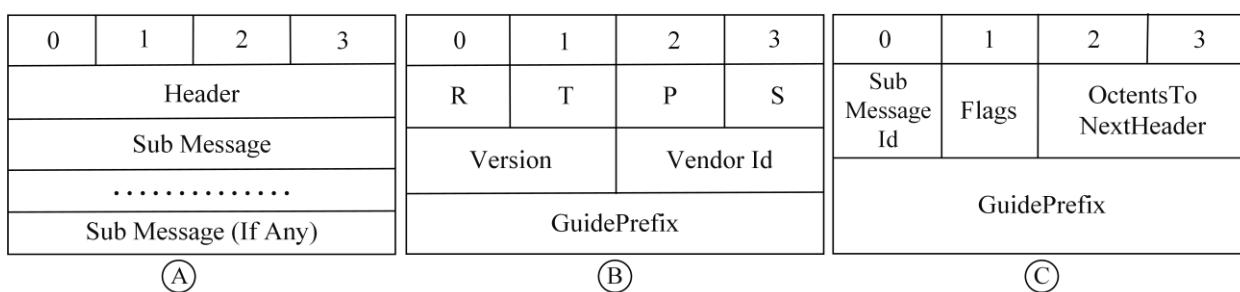
DDS is used by many applications like Rail Network management, Volkswagen Smart Cars for driver assistance, Air Traffic Management System, and many more [215]. The DDS relies on the implementation of the Real-Time Publisher Subscriber Protocol (RTPS) because of its unique features like QoS, plug, and play connectivity, extensibility, and fault tolerance. RTPS is designed to be compatible with both TCP and UDP transport layer protocol suite [213].

The RTPS message format is shown in Figure 25(A), and it consists of a leading header and a variable number of sub-messages. Figure 25(B) realizes the RTPS message header format, and its length is about 16 bytes with five core sections. Firstly, the protocol field is 4 bytes long; this value must be set to RTPS; The second field is 2 bytes version field, which is used to indicate the RTPS version; The third field is two bytes long VendorId field. Each RTPS implementation must obtain from OMG VendorId; The last field, GuidePrefix, has a length of eight bytes. The Publisher, Subscriber, DataReader, and DataWriter are GuidePrefix entities with their global identifiers, and these entities are used to guide the source and destination [216].

The sub-message header and payload are part of the sub-message, and it is depicted in Figure 25(C). The SubmessageId, Flags, and OctetsToNextHeader fields are part of the sub-message. The SubmessageId uniquely identifies the type of the sub-message. The sub-messages use flags. Payload length is indicated by the OctetsToNextHeader field [216].

The sub-message uses 12 message types [216]

1. Data Message: It comprises of an application data object value
2. DataFrag Message: It is used to address the oversized



**FIGURE 25. Message Format of DDS (A. RTPS Message Format, B. RTPS Header Format & C. RTPS Sub Message Format)**

- data problem. Here, the data is transmitted in the form of multiple fragments.
3. HeartBeat Message: Checks the readiness of the Reader by the Writer.
  4. HeartBeatFrag Message: Indicates the available fragments.
  5. Gap Message: Indicates the relevance of the information to the reader by the writer.
  6. AckNack Message: Received by the writer from sender to provide information on data delivery.
  7. NackFrag Message: Received by the writer from sender to indicate the missing of the fragment.
  8. Info-SRC Message: Used to change the source of the sub-message.
  9. Info-DST Message: Used to change the destination of the sub-message.
  10. InfoReply Message: It indicates where to send the reply to this sub-message and the sub-messages following this sub-message.
  11. InfoTimeStamp Message: Indicates the TimeStamp of the Source.
  12. PAD Message: Used for sub-message Padding.

#### Analysis

- Point to Point latency is very low compared to other application layers protocols.
- It provides a rich set of QoS options.
- DDS security provides advanced features like authentication and encryption.
- Multicasting is used to enable scalability of the data.
- It is not suitable for constrained devices.

#### E. RESTful APIs/ HTTP

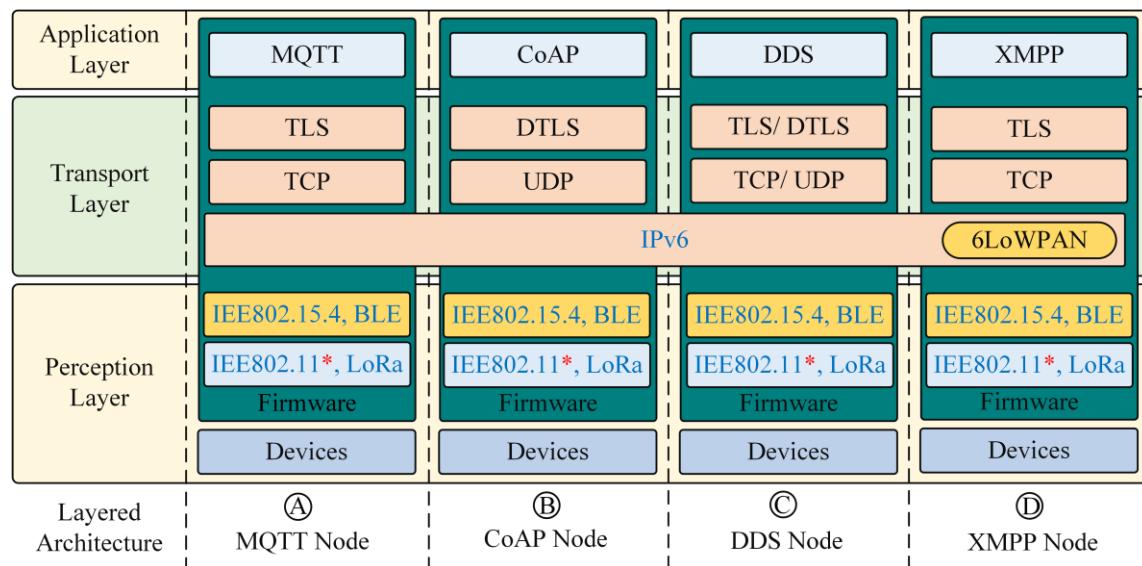
RESTful API is also termed as a RESTful web service or REST APIs, which are adhered to the constraints of the

REST architecture. RESTful APIs use HTTP methods like GET to retrieve, PUT to change or update, POST to create, and DELETE to remove the resources. These methods are defined in RFC2616 protocol. A typical resource is nothing but an image, audio, video, text, shopping order, etc., which are represented using Text, XML, and JSON formats. RESTful API is based on the client-server operation, which is stateless and use URI to identify the resources uniquely. Both clients and servers are independent of each other. RESTful APIs allow the developer to bring multiple services to IoT applications swiftly and make IoT applications distributed and independent. RESTful APIs receive the data from both client and server; this data is used to create new data. The created data is updated among clients and servers; finally, the data is deleted.

#### Analysis

- Use of fewer resources
- Supports multiple data formats (Text, XML, and JSON)
- Not suitable for handling the massive amount of data
- Need to be incorporate security features
- Offers greater flexibility and Scalability

Figure 26 represents the deployment of the different IoT nodes based on the application layer protocol used (i.e. A. MQTT Node, B. CoAP Node, B. DDS Node, and D. XMPP Node). The devices and firmware and communication standards may varies based on the IoT application developer's interest. However, Zigbee, BLE, and WiFi are the most commonly used for short range communication standards. Similarly, for long-range communication LoRa is used in IoT. IPv6 is the developer's choice for identification in IoT as the number of devices are increasing in the world. The communication standards like Zigbee and BLE use 6LoWPAN, which enables the transmission of data with minimum processing capabilities. WiFi standard has the option of adapting the 6LoWPAN. However, most



**NOTE:** \* 6LoWPAN can be adapted to work with IEEE 802.11 also (WiFi)

**FIGURE 26.** Working Models different of IoT Nodes

of the WiFi-based IoT networks do not use 6LowPAN. All the time, the IoT application layer protocols use either TCP or UDP as the transport layer protocols. TCP and UDP use the TLS and DTLS protocols for security purposes, respectively. Figure 27 depicts the layerwise protocol stack of IoT, and Table VI provides a feature analysis of IoT application layer protocols.

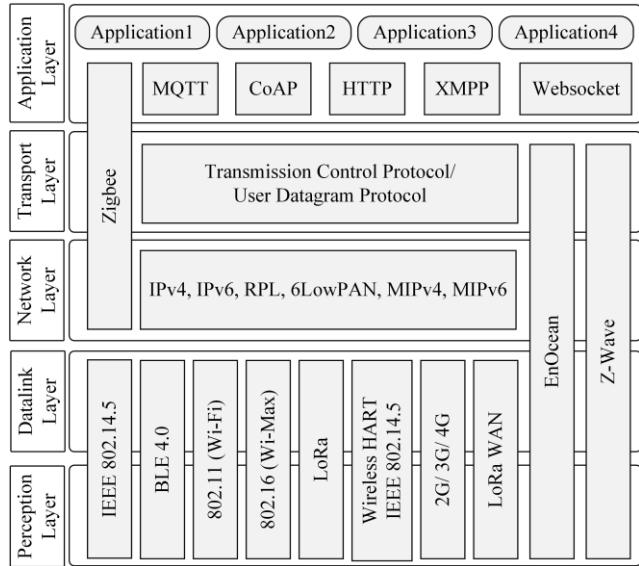


FIGURE 27. Protocol Stack of IoT

Countable IoT application layer protocols were introduced, but still, HTTP and MQTT protocols are the developer's choice to implement IoT applications. IoT application layer protocols like CoAP, AMQP, and XMPP protocols are infrequent in real-time IoT deployments [217]. Tanin Sultana et al. [218] provided a detailed analysis of MQTT, AMQP, DDS, HTTP, CoAP, and DDS protocols using IoVT framework. In this framework,

cameras were used to capture real-time data. The analysis was performed based on latency, throughput, power consumption, memory, and CPU utilization. MQTT and AMQP have shown better results in terms of latency, throughput, and power consumption. MQTT and HTTP showed less memory usage, and XMPP has shown higher memory utilization. AMQP exhibits higher CPU utilization and XMPP with the lowest CPU utilization. Raspberry Pi based health monitoring system was used by Yuang Chen et al. [219] to analyses the performance of the MQTT, CoAP, and DDS. MQTT and DDS protocols have shown the zero-packet loss. DDS has shown better performance compare to CoAP in constrained networks. However, the bandwidth consumption of DDS was much high compare to MQTT and CoAP. Jasenka Dizdarevic et al. [220] provided the analysis of IoT application layer protocols (HTTP, MQTT, CoAP, AMQP, DDS, and XMPP) in fog and cloud-based IoT systems. The analysis was based on the characteristics of the protocols and performance metrics (Latency, Power Consumption, and Throughput). On the basis of tests conducted by the authors, the MQTT protocol has shown improved performance compared to other protocols. Fog and cloud-based IoT systems still use the HTTP protocol. The wide adoption of HTTP is due to the lack of maturity and stability in IoT application layer protocols. Shahid Raza et al. [221] summarized the features of DTLS and presented a scheme to compress the DTLS headers. The aim of this compression scheme is to reduce energy consumption without compromising end to end security. DTLS header compression is performed using the 6LoWPAN-NHC compression mechanism, and the overall performance of the proposed scheme was analyzed using Contiki OS. Lukas Malina et al. [222] proposed the security framework for MQTT, which consists of three distinct levels (Level 1, Level 2, and Level 3). Level 1 is used to protect the messages against the tampering and

TABLE VI  
FEATURE ANALYSIS OF APPLICATION LAYER PROTOCOLS

Parameters	CoAP	DDS	MQTT	XMPP	HTTP/ HTTPS
Standard	IETF RFC 7252	OMG (Object Management Group)	OASIS Standard	IETF RFC 6120, 6121	RFC2616
Transport Layer Protocol	UDP	TCP/ UDP	TCP	TCP	TCP
Messaging Pattern	Req/Resp	Pub/Sub	Pub/Sub	Pub/Sub & Req/Resp	Req/Resp
Processing Type	Decentralized	Decentralized	Centralized	Decentralized	Decentralized
Security	DTLS	TLS, DTLS & DDS Security	TLS	TLS	TLS
QoS	Yes (2 Levels)	Yes (23 Levels)	Yes (3 Levels)	No	No
Topology	Tree	Bus	Tree	Star	-
RESTful Support	Yes	No	No	No	No
Data Encoding	Defined	Defined	Undefine	Defined	Defined
Resource Locator	URI	Topic Name	Topic Name	Jabber Identity	URL
Default Port Number	5683	Dynamic	1883	5222	8080/ 443
Header Size	4 Bytes	16 Bytes	2 Bytes	-	-
Multicast	Yes	Yes	No	Yes	Yes
Libraries (Python)	CoAPthon, txThings	pyDDS, rticonnextdds	pahoMQTT	SleekXMPP	Requests
Purpose	Applications Based on Constrained Devices	Industrial Automation	Lightweight M2M Communications	Instant Messaging (Chat Applications)	Web Browsing

modification attacks. This level is not responsible for message confidentiality. Publishers and subscriber's privacy were protected using Level 2; this layer is also resistant against tampering and modification attacks. Confidentiality and partial anonymity security features were provided by this level. At this, the authentication is done using the Schnorr digital signatures, and the data is one-time encrypted using the Rabin cryptosystem. Hence, this level is useful to share the messages, which contains personal information. Level 3 enables mutual authentication (using Schnorr digital signatures) between the participants i.e. Publishers, Broker and Subscribers. This protects the data against the tampering, modification and eavesdropping. This enables to create long-term secure communication sessions between the participants. In this, the confidentiality of the data is achieved by AES algorithm. Heng Wang et al. [223] proposed lightweight XMPP publisher/ subscriber scheme for IoT devices. This scheme, allows the event driven data exchange between the constrained IoT devices and it is compatible with both TCP and UDP. Based on the interest, subscribers send the service request to the servers. Publishers reacts to the service requests using the subscription rules. Sleep and wake-up modes are used by the proposed scheme to enhance the life of the battery. Publishers provide the requested services to subscribers, when they are in wake-up mode. For this cause, publishers share the sleep and wake-up messages with the server. These messages are used by the server to manage the subscription between the publisher and subscriber effectively.

## VIII. COMPUTING PARADIGMS

In traditional data models, data was generated by a limited number of companies, and others were consuming the data. However, the days have been changed; everyone is generating and using the data. IoT also contributes significantly to create a large volume of data. Due to the lack of resources and computing facilities, handling of the enormous volume of data in IoT is a tedious task. Computing paradigms like cloud computing, cloudlets, fog computing, and edge computing came into the picture to facilitate the IoT data cycle (Storage, Pre-processing, and Analysis) [224].

### A. Edge Computing

Edge computing is a networking architecture used to perform certain computations (Analysis), where data is

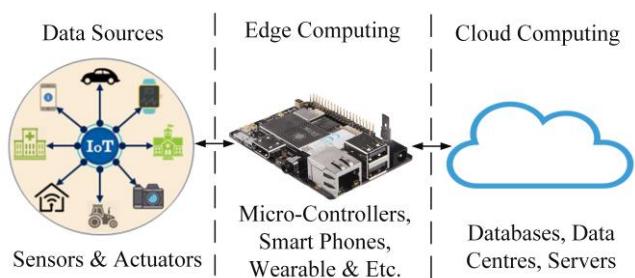


FIGURE 28. Edge Computing Architecture

captured [225]. Figure 28 depicts the concept of the General Edge Computing Architecture (GECA). Proximity, Intelligence, and Control are considered as the major components of edge computing [226].

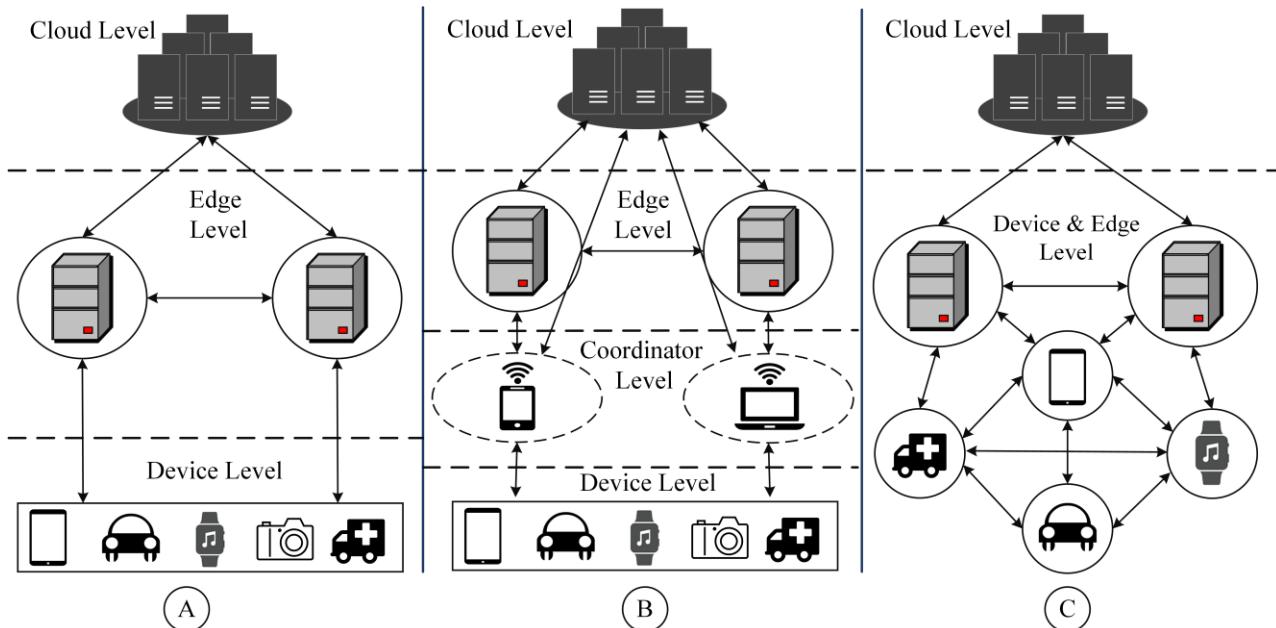
#### 1) Characteristics of Edge Computing

- Latency: Centralized cloud computing is not appropriate for all applications, as some applications require immediate solutions. Latency is more due to the sharing of resources and distance between objects; the cloud server must serve multiple objects at a time. The latency problem can be reduced by placing the edge nodes between the objects and the cloud server [227]. Researches have constructed a proof-of-concept platform for running face recognition applications by shifting the computation from cloud to edge and which reduces the response time from 900 to 169 milliseconds [228].
- Security: The objects have insufficient memory to implement conventional security algorithms in IoT, and cloud platforms are the target of hackers most of the time. Once the edge nodes have become the part of IoT networks, data generated by these objects can be secured by implementing the novel lightweight security algorithms in it [229].
- Real-Time Data Analysis: Connected objects are increasing in number, which results in the generation of excessive volume of data. Understanding and extracting the useful patterns from the excessive amount of heterogeneous data generated by the objects effectively is a tedious task. The edge computing paradigm enables the processing and analysis of data in real-time at the edge of the IoT network [230].
- Mobility: Mobile devices are increasing rapidly in day to day life. Mobility of the edge nodes enables the distributed computing in the network. Locator Identity Separation Protocol (LISP) is used by edge computing devices to communicate with mobile devices [231].

#### 2) Edge Computing Architectures

Currently, there are no architectural standards in edge computing, but there are industries and research institutes like Open Edge Computing Community, Open Fog Consortium, and European Telecommunication Standardization group working on edge computing architectures. The three Edge Computing Architectures (A). Edge Server Architectures (B). Edge and Coordinator Device Architecture, and (C). Device Cloud Architecture is shown in Figure 29 [232], and Table VII gives the differences between the different edge computing architecture.

- Edge Server Architecture (ESA): This architecture is generic, where all the objects are connected to the Edge Server (ES), and ES is connected to the rest of the network, including the cloud. In this architecture, ES is static with high computational power. This architecture gives a clear difference between the device level, edge



**FIGURE 29.** Edge Architectures (A. Edge Server Architecture, B. Edge Coordinator Device Architecture & C. Device Cloud Architecture)

level, and the other networks. Nano Data Centers are examples of edge servers [232].

- Edge and Coordinator Device Architecture (ECDA): In this, the communication between the objects and the edge servers is via coordinator objects. The coordinator objects may be smartphones, tablets, or laptops. These objects have more power, bandwidth, and computation facilities compared to edge servers. The coordinator devices can communicate with the other networks via edge servers, or they can communicate directly without the help of edge servers [232].
- Device Cloud Architecture (DCA): Both device level and edge level are merged in this architecture. If necessary, the devices communicate with the cloud via edge devices. The researcher considers this type of architecture as opportunistic computing or transient clouds (Cooperative computing platforms, which enables nearby devices to form a network and offers various services) [232][233].

TABLE VII  
COMPARISON OF EDGE ARCHITECTURES

Parameters	Edge Architectures		
	ESA	ECDA	DCA
Bandwidth	High	Moderate	Low
Computational Capacity	High	Moderate	Low
Mobility of the Nodes	Fixed Position	Mobile	Fixed and Mobile
Data Storage	Less Storage	High Storage	No Storage

Edge computing is seeking attention among the academicians and researchers. Most edge computing frameworks use dedicated physical devices to coordinate IoT devices. Edge devices in IoT provide temporary storage and data analysis features. Query processing, activity

recognition, malicious activity, and malware detection, augmented reality, surveillance, and resource management applications are taking the advantages of edge computing technologies. Md. Zia Uddin et al. [234] developed a wearable sensor-based system to predict human activities using Recurrent Neural Networks (RNN) over an edge device. In this work, MHEALTH dataset was used, which comprises of magnetometer, electrocardiography, accelerometer, and gyroscope sensors data. All the data from the sensor are recorded at a sampling rate of 50 Hz. The system predicts the activities like standing, relaxing, lying down, walking, climbing, jogging, cycling, running, jumping, and bending. The authors claimed that they achieved a 99.69% mean prediction rate. A laptop with a Graphical Processing Unit (GPU) was used as an edge device to perform the activity predictions. Xianlong Zhao et al. [235] proposed a data offloading in mobile edge node based on the deep learning technique. A multi-Long Short-Term Memory (LSTM) deep learning model was used to predict the real-time traffic, which helps to perform the data offloading process accurately. Based on the prediction result, the cross-entropy method was used to perform the data offloading, and it is based on the offloading indicator. In this, the JINHUA dataset is used to approximate real-time traffic. Jie Yuan et al. [144] proposed a lightweight and reliable trust mechanism to increase the QoS and mitigate bad-mouthing attacks triggered by malicious feedback providers. This mechanism is based on the calculation of the trust using multi-source feedback (Multi-Source Feedback - feedback gathered from both devices and edge devices) fusion. The feedback trust is divided into 1. Device to Device Trust – In this, device trust is formed using the communication history of the devices, 2. Device to Broker Trust – In this, the broker computes the real-time trust of each device after completing the computation task and shares with the other devices, and 3. Overall Trust – is

based on the fusion calculation using the trust of multiple devices. The authors have introduced the broker layer (Layer consists of edge devices) between the device and network layer, and it is responsible for monitoring the activities of the devices and feedback trust calculations using objective information entropy theory.

### B. Fog Computing

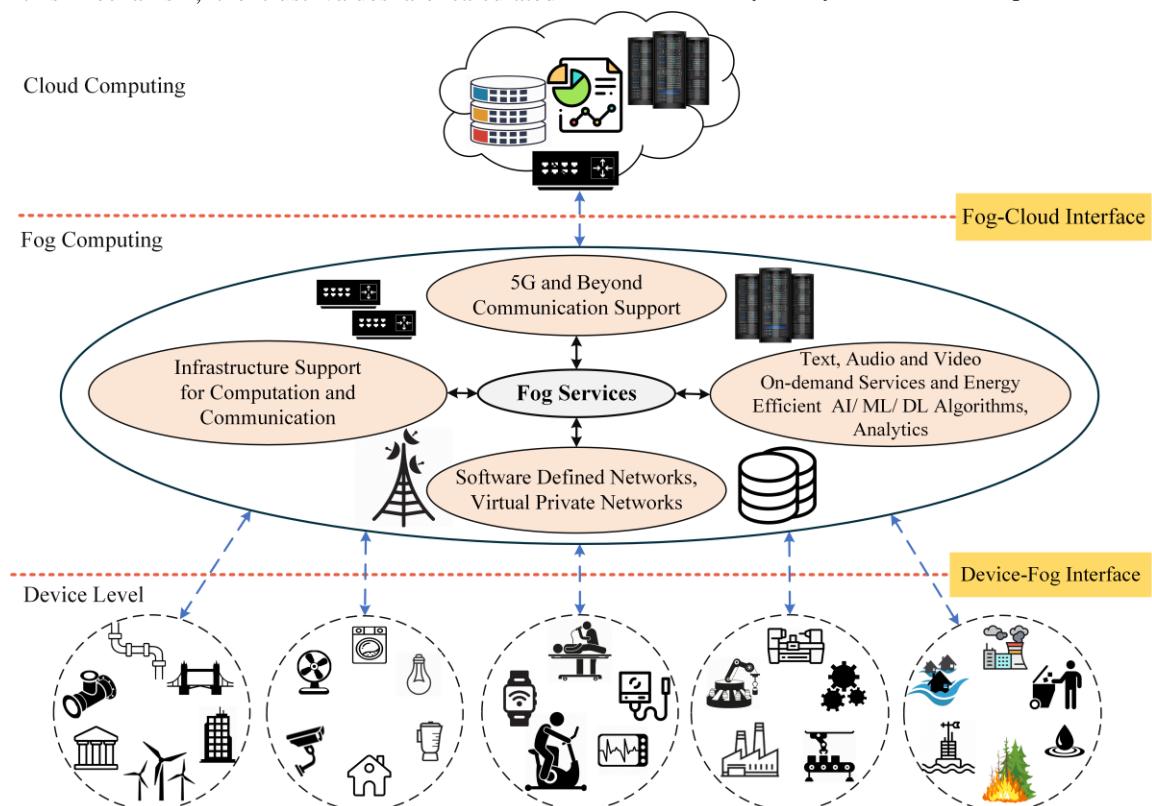
Fog Computing or Fogging is a computing architecture, which was coined by CISCO in 2014 [236]. This computing paradigm minimizes the load on the cloud by providing the computing, communication, storing, and networking facilities [237]. The fog Computing layer acts like a micro data center for IoT applications [238]. Typically, Fog Layer consists of Servers, Routers, Switches, and Access Points. The objects are connected to the fog nodes to acquire the services provided by them. This computing paradigm plays a key role in the applications where immediate resources and responses are needed. IoT Applications like Disaster Management Systems, Structural Health monitoring, smart health, etc. comprise of critical data, which require immediate processing. In IoT, the fog nodes perform the data aggregation, removal of redundant data, filtering and data analysis tasks swiftly [239]. Figure 30 illustrate the architecture of Fog Computing.

Geetanjali Rathee et al. [240] proposed a security mechanism to mitigate various attacks based on the trust values in fog based IoT networks. A trust manager is used to keep track of all fog and IoT nodes utilizing a look-up table. In this mechanism, the trust values are calculated

using Social Impact Theory Optimizer, and values vary between 0 and 1. These values are assigned to the neighbors using the previous interaction history. The trust manager maintains a copy of the fog node's identity, address, trust values, and ratings. Tidal trust algorithm is used to calculate the ratings of the nodes and establish the trust between the fog nodes and users. The fog node is said to be malicious when the trust value of the node is less than the assumed threshold value. Gunjae Yoon et al. [241] proposed a sensor data management method using a fog node, where the fog node acts as a monitoring node, and the authors also discussed the framework of the monitoring node. The monitoring node analyzes the sensors incoming data by means of data patterns (Varying of data values of a sensor over a time). The monitoring node reports to the Operational and Management Server (OMS) only when the analyzed data is beyond the normal range. The fog node is capable of controlling the sensor devices based on the instructions from the OMS. Shreshth Tuli et al. [242] proposed architecture to diagnose heart diseases using the features of IoT and Fog computing. The authors developed the deep learning-based HealthFog system to diagnose or predict heart diseases in the patients, and the system is deployed using the FogBus framework to analyze the performance of the system, such as accuracy, response time, and energy consumption. The proposed method used the Ensemble of deep neural network models for predicting the heart diseases on the Cleveland dataset.

### C. Cloudlets

Mahadev Satyanarayanan et al. have provided the notion of



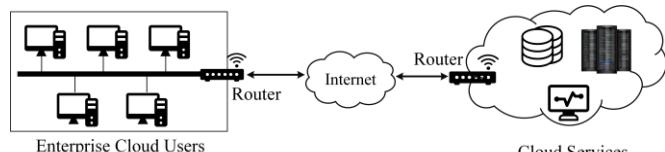
**FIGURE 30. Fog Computing Architecture**

cloudlet in the year 2009 [243]. A cloudlet is a trusted data center, which swiftly provides cloud computing services to devices like smartphones, wearable, autonomous vehicles, and so on. Typically, cloudlets are situated at the base stations [244]. Cloudlets aims at reducing the end to end latency between the cloud and mobile devices. Also, cloudlets maintain the privacy of the user's data before offloading them to the cloud [245].

The authors in [246] have summarized the problems associated with the use of Unmanned Aerial Vehicles (UAVs) to provide the latency-sensitive services using IoT. In this, UAVs are used as cloudlets, which offers the computation offloading ability to resource constraint IoT devices. In this work, servers are mounted on the UAVs. These UAVs fly near to the IoT devices and process the data generated by them. S. Lai *et al.* [247] proposed a Fairness-oriented computation offloading scheme FairEdge, which enables balanced task distribution in the IoT and cloudlet networks. The FairEdge uses the Balls-and-Bins load balancing theory to offload the IoT tasks into cloudlets. The fairness index of the scheme is calculated using the Jain's fairness index, and it is a part of the task offloading scheme. The authors mentioned that the proposed FairEdge scheme reduces the unbalanced task offloading by 50%. In [248], IoT enabled cloudlet assisted e-health framework is proposed. The aim of this framework is to make it easier to access the real-time data using cloudlets. This framework comprises of IoT, cloudlet, and cloud layers. IoT layer is a collection of healthcare devices to monitor the status of the health in real-time, and these devices are communicating with IoTHub (Gateway) using short-range wireless communication standards. Cloudlets are introduced between the IoTHub and cloud. Usually, data processing tasks were carried out in the cloudlet layer, and cloudlets belong to different geo-locations share the healthcare data among themselves. The cloud layer performs the various queries and analytics operations on the healthcare data.

#### D. Cloud Computing

It is a computing standard [249], which enables the users to access a set of services offered by the service providers over the internet. Cloud computing architecture is depicted in Figure 31. Infrastructure, Data storage, resource sharing, and software access are the services offered by service providers [250]. Cloud computing consists of following entities [251]: 1. Cloud Provider – delivers the set of services to the interested parties, 2. Cloud Consumer – utilize the services offered by the cloud provider, 3. Cloud Broker – acts as an intermediate between the cloud provider and consumers to manage the service delivery, 4. Cloud



**FIGURE 31.** Cloud Computing Architecture

Carrier – intermediate that provisions the connectivity and delivery of cloud services and 5. Cloud Auditor – evaluates the cloud infrastructure, services, system operations, performance, and security of the cloud implementations. Cloud Computing helps small enterprises to create and deliver Information Technology solutions as early as possible to the end-users [252]. E-Governance, Smart Healthcare, Intelligent Transportation Systems, Life Sciences, Smart Grid, Data Analytics, Disaster Management System are some of the applications, which make use of cloud services for different purposes. These applications produce a huge volume, velocity, and variety of data. Cloud Computing is responsible for storing and analyzing the data more swiftly [224].

Private, Public, and Hybrid Cloud are the three categories of the cloud [224][253]. Always, the private cloud access is provided to the selected people of the organization. Irrespective of affiliation and origin, all the users can pay and access the services provided by the public cloud. The Hybrid cloud possesses the features of both private and public cloud [224]. Data analysis, data management, device management, scalability, and security features should be considered while designing and developing cloud-based IoT solutions. Table VIII illustrates the features of the different Cloud platforms for IoT.

Tae-Dong Lee *et al.* [254] presented hierarchical cloud computing architecture to manage context-aware services in IoT. The architecture comprises of Cloud Control and User Control layers. The cloud control layer is responsible for resource allocation and scheduling. Similarly, the user control layer is responsible for end to end connection between the devices and the cloud. The proposed architecture consists of 1. Non-uniform Service Binding Model – this model delivers an application-oriented computing environment at the platform level, 2. Real-time Adaptable Service Binding Model – this model comprises of application and transmission bindings, and this uses the meta object-based reflective system and 3. Intelligent Service Management – this model provides intelligent service management using the supervised and reinforcement ML algorithms. Ing-Ray Chen *et al.* [255] proposed a service management protocol called IoT-HiTrust for bulk mobile cloud systems. The proposed protocol uses trusted nodes to increase the performance of the applications with respect to security, scalability, and accuracy. The proposed mechanism was based on public relations among the owners of the IoT devices. The trust node's calculations are based on the recommendations received from the IoT devices. The recommendation filtering algorithm applied on these recommendations, which possess the features like familiarity (list of other friends IoT devices), community contact (represents the closeness), and community attention (represents the knowledge on the subject matter). Smart city travel and air pollution detection applications were used to analyze the feasibility and performance analysis of the proposed protocol. The authors demonstrated the simulation of the

TABLE VIII  
FEATURES OF IOT CLOUD PLATFORMS

Characteristics	Application Layer Protocols	Sensor Data Visualization	Security	Language Support
<b>Amazon IoT Cloud Platform</b> ( <a href="https://aws.amazon.com/iot/">https://aws.amazon.com/iot/</a> )	MQTT, WebSocket, HTTP	Yes	TLS	C, Node.js, Python, Java
<b>Azure IoT Hub</b> ( <a href="https://azure.microsoft.com/en-in/services/iot-hub/">https://azure.microsoft.com/en-in/services/iot-hub/</a> )	MQTT, AMQP, HTTPS, AMQP	Yes	X.509	C, C++, Python, Java, Node.js, C#
<b>OpenIoT</b> ( <a href="http://www.openiot.eu/">http://www.openiot.eu/</a> )	MQTT, CoAP, AMQP, XMPP	Yes	TLS	Java
<b>Thing Speak</b> ( <a href="https://thingspeak.com/">https://thingspeak.com/</a> )	MQTT, REST API's	Yes	SSL/ TLS	C, Python, MATLAB
<b>IBM Watson IoT Platform</b> ( <a href="https://www.ibm.com/in-en/internet-of-things">https://www.ibm.com/in-en/internet-of-things</a> )	MQTT, HTTP	Yes	-	Python, Java, Node.js, C#, Embedded C
ThingWorx ( <a href="https://developer.thingworx.com/platform">https://developer.thingworx.com/platform</a> )	MQTT, WebSocket, HTTP	Yes	-	C, Java, Python
<b>Xively</b> ( <a href="https://xively.com/">https://xively.com/</a> )	MQTT, WebSocket, HTTP	Yes	TLS	C, C++, Python, Java
<b>Zetta</b> ( <a href="https://www.zettajs.org/">https://www.zettajs.org/</a> )	MQTT, CoAP, WebSocket, HTTP	Yes	Yes	Node.js
<b>Temboo</b> ( <a href="https://temboo.com/iot">https://temboo.com/iot</a> )	HTTP, MQTT, CoAP	-	-	C, Java, Python, Ruby, JavaScript

protocol using the NS-3 tool. The proposed mechanism withstands against the attacks like Self-promoting, Bad-mouthing, and Ballot-stuffing attacks. Yi Xu et al. [254] proposed a Cloud-Edge-Beneath architecture to enhance the scalability features in IoT networks. The architecture is made up of Beneath, Edge, and Cloud layer. The beneath layer is further comprised of physical sensors and sensor platform layer. This layer is accountable for gathering the raw data using sensors and establishing the communication

between sensors and edge nodes using the sensor platform layer (Low power computing and communication devices). The edge layer is a collection of edge devices used to manage the group of sensors. The formulation of the sensor group is based on their functionality, type, and location. The cloud layer is a collection of services accessed by the devices in the beneath and edge layer. This architecture implemented using Service Oriented Device Architecture (SODA). SODA

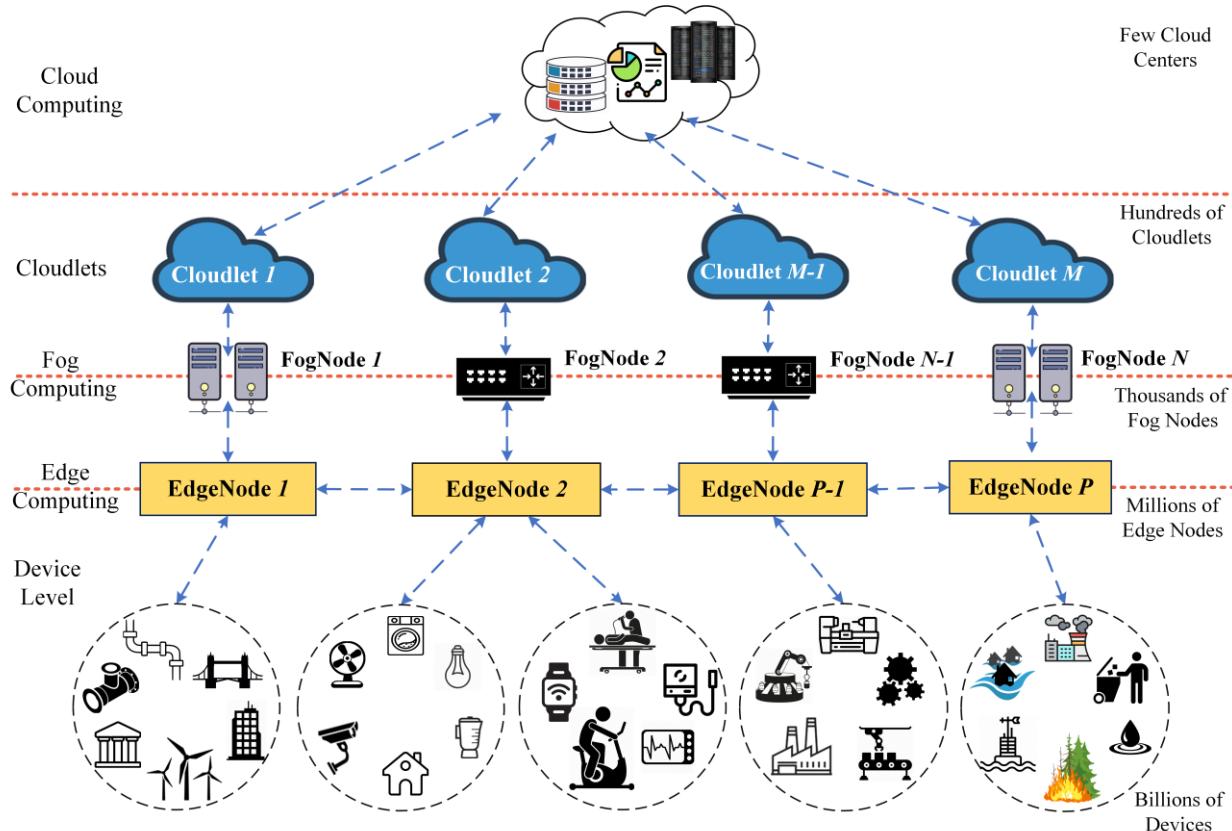


FIGURE 32. IoT Computing Hierarchy

TABLE IX  
COMPARISON BETWEEN THE COMPUTING PARADIGMS

Characteristics	Edge Computing	Fog Computing	Cloudlet	Cloud Computing
Architecture	Distributed	Distributed	Distributed	Centralized
Response Time	Milliseconds	Seconds	Seconds	Minutes
Number of Hops from Source	One Hop	One Hop/ Few Hops	One/ Few Hops	Multiple Hops
Latency	Less	Medium	Medium	High
Data Analysis	Real Time	Near Real Time	Near Real Time	No
Target of Attack	Very Low	Low	Moderate	High
Mobility	Supported	Limited Support	Support	Not Supported
Multitasking	Limited Support	Supports	Support	Supports
Security	Secured	Secured	Less Secure	Less Secure
Scalability	Less	Intermediate	High	Very High
Network Congestion	Very Low	Low	High	Very High
Data Storage	Very Less	Less	High	Very High
Operating Overhead	Less	Less	High	Very High
Energy Consumption	Less	Less	High	Very High
Type of Storage	Temporary	Hours to Days	Permanent	Permanent
Communication Standards	WiFi, Zigbee, Z-wave, BLE	WLAN, WiFi, 3G,4G, 5G ZigBee, Z-wave, BLE	WLAN, WiFi, 3G,4G, 5G and IP Networks	IP Networks
Computing Devices Used	Smart Sensors, Smart Phones, Smart Vehicles, Sometimes Serve	Servers, Switches, Routers, Access Points	Servers,	Servers, Data Stores
Applications	Home Automation, Smart Farming, Structural Health Monitoring, Smart Health, Smart Gardening, Connected Cars	Traffic Signaling & Monitoring, Smart Health, Connected Cars	Mobile Games, Smart City, Maps, Intelligent Transportation	Smart Homes, Smart Cities, Smart Grids, Banking, SHM

automates the sensor/ device integration using the Atlas Sensor Platform and middleware.

Figure 32 shows the levels of IoT computing architecture, which included device, edge, fog, cloudlet, and cloud computing levels. Edge nodes provide the required computing services to the devices and consume the fog services. The fog nodes also provide energy-efficient routing and communication services. Cloudlets bring cloud services to users and devices. Table IX shows the comparison between the computing paradigms used in IoT.

## IX. IOT SECURITY

Security is defined as the group of techniques used to prevent, restore, and ensure the safety of the data (Both

application and network data) from malicious attacks [256]. Sensitive data captured from the physical world is the target of hackers. The attacker identifies and exploits the vulnerabilities to steal and modify the data [257]. Security and privacy concerns in IoT are unsolved and challenging for the researchers due to the nature of the devices being used and enabling technologies [258][259]. Further, security threats are increasing due to the widespread adoption of IoT devices in daily life. The security measure should be addressed from the device design stage to the deployment stage [260]. In the Japanese language, the term Hajime means ‘beginning.’ It is a malware used to perform a DDoS attack. The devices connected over the internet are the target of this malware. Totally, 297,499 unique devices

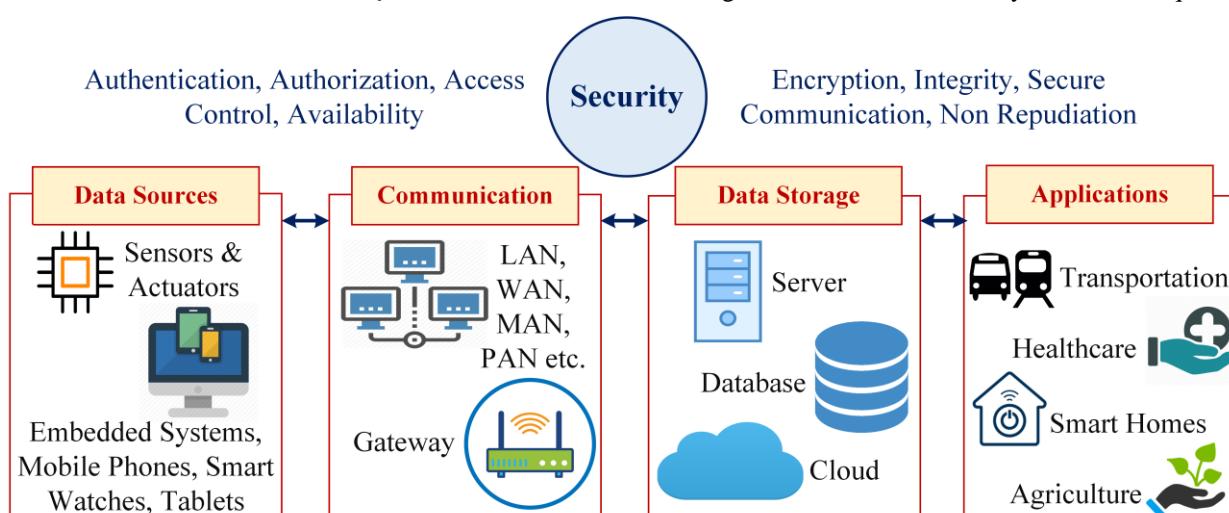


FIGURE 33. Security Concern in Different Stages of IoT

TABLE X  
MECHANISMS USED IN SECURITY SERVICES

Security Level	Security Services	Security Mechanism	Algorithms
Information Level	Confidentiality	Cryptography	Symmetric (AES, DES, 3DES) and Asymmetric Cryptographic (RSA, DSA, IBE, ABE) Algorithms
	Integrity	Hash Functions, Message Signature	SHA-256, MD5, HMAC
	Privacy	Pseudonymity, K-Anonymity, Zero Knowledge Proof (ZKP)	EPID, DAA, Pedersen Commitment, Data Tagging, Data Obfuscation
Access Level	Authentication	Message Authentication, Network Authentication	HMAC, Password Authentication, Challenge Handshake Authentication, Extensible Authentication, Kerberos
	Authorization	Access Control Mechanism	Role Based Access Control, Attribute Based Access Control, Discretionary Access Control
Functional Level	Availability	Intrusion Detection and Prevention Systems, Access Control, Firewalls	Machine Learning and Artificial Intelligence based Intrusion Detection Algorithms

were affected by this malware. This Malware originated from Vietnam, Taiwan, and Brazil [261]. IoT security concern in different stages of IoT is depicted in Figure 33. The Organization like the National Institute of Standards and Technology (NIST) working continuously to draft the cyber-security specification. Also, NIST drafted some security specifications like NISTIR 8259, 8259A, and 8228 [262]. In 2019, The European Union Agency for Cybersecurity (ENISA) published good practices for IoT security [263][264]. ETSI TS 118 103 and TS 003 are the standards defined by the oneM2M, which specifies the security solutions for IoT [265].

#### A. Security Services

Security services are used to protect computer networks, information system, and its associated data from disclosure, alteration, disruption, destruction, unauthorized access, and use. Confidentiality, Integrity, Privacy, Authentication, Authorization, and Availability are general security requirements [256]. The security requirements are grouped into Information, Access, and Functional level, which helps to identify and classify the target of attackers [260]. Figure

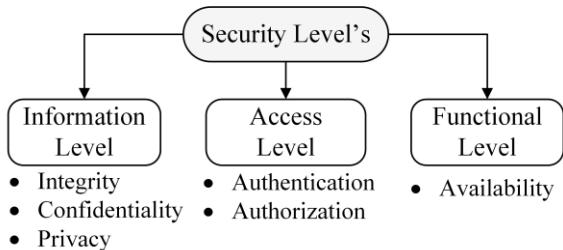


FIGURE 34. IoT Security Levels

34 depicts the classification of the security levels, and Table X summarizes the common mechanisms used to enable security at each level.

##### 1) Information Level

At this level, the utmost concern is to defend the attacks against confidentiality, integrity, and identity of the data.

a. Confidentiality: It ensures that the data is disclosed to authorized users only. Achieving confidentiality is more complicated in the IoT since the attacker can place the sensors and capture the data [266]. Eavesdropping is the

most common attack used to exploit confidentiality [267].

- b. Integrity: Integrity ensures that the data reached the endpoint without modification. Maintaining integrity is significant; otherwise, it leads to false alarms/ decisions. For Example, an attacker may change the sensor data sending to the server. The most common integrity attack is false data injection [268].
- c. Privacy: The private information of the device and users should not be disclosed during data dissemination to the unauthorized users [269]. Maintaining privacy in IoT networks is crucial as application data contains the user's personal identity and location information most of the time. Identity theft, traffic analysis, and tracking are the most common attacks on privacy.

##### 2) Access Level

The aim of the security mechanism at this level is to identify, monitor, and grant access to the services and resources to the users or devices.

- a. Authentication: Authentication ensures that only legitimate users or devices are consuming the application services and resources [270]. This is the first operation performed by the server or nodes when the user sends any request. Brute Force attack is most commonly used to exploit authentication.
- b. Authorization: Authorization ensures that users or devices are permitted to access the services and resources of the IoT ecosystem. Session fixation and session hijacking are the most commonly used techniques to access the un-authorized services and resources [271][272].

##### 3) Functional Level

This security mechanism guarantees the provision of services and resources to the users and devices, even in the event of an attack.

- a. Availability: This ensures resources like data, network infrastructure, and devices are accessible by authorized users when required. In IoT, the data generated by the devices are real-time in nature, and this data is used by the different services to take necessary actions. The

most common attacks to exploit availability is Denial-of-Service (DoS) and Distributed DoS (DDoS) attack [4][273].

### B. IoT Security Implementation Challenges

#### 1) Constrained Devices

The majority of IoT devices have limited resources (Computing and Storage). The use of traditional security algorithms in IoT results in high time, memory, and power consumption [274]. To secure IoT applications, an energy and memory aware lightweight security algorithms are needed. Table XI and Table XII highlight the classification of resource constraint devices [275][276] and an overview

TABLE XI  
CLASSIFICATION OF RESOURCE CONSTRAINT DEVICES

Name	RAM Size	Flash/ ROM	IoT Protocol Stack Support
Class 0 (C0)	<10KiB	<100KiB	Partial
Class 1 (C1)	~10KiB	~100KiB	Extended Support
Class 2 (C2)	~50KiB	~250KiB	Full Support

of a few constraint devices used in IoT, respectively.

#### 2) Battery Life

Most of the IoT devices are functioning in unattended

environments, and it is difficult to replace the battery of the devices immediately. Power consumption is directly proportional to the computation and communication cost. There are different ways to reduce this impact on regular actions; 1. Security mechanisms should consist of fewer calculations, 2. Increase the battery capacity. (This is difficult, as IoT devices are tiny and there is no additional space to accommodate larger batteries) and 3. Use of natural resources such as solar and wind energy. However, this requires upgrades to hardware and special skills.

#### 3) Complex Environment

IoT is not a standalone system; it's an integration of entities like heterogeneous devices, interfaces, and people. While designing the security models, researchers should analyze the environment of an IoT system and enable the security feature for every entity of it.

### C. Common Security Mechanisms for IoT

Figure 35 illustrates the common privacy and security threats in the three-layered architecture of IoT. Lightweight cryptography, Hardware security, and Intrusion Detection and Prevention Systems are used to mitigate security and privacy threats in IoT networks.

#### 1) Lightweight Cryptography

Lightweight Cryptography or Encryption is not a new

TABLE XII  
LIST OF RESOURCE CONSTRAINED DEVICES

Device Name	Controller	Architecture	RAM	ROM	Class
<b>Arduino Ethernet REV3</b> ( <a href="https://store.arduino.cc/usa/arduino-ethernet-rev3-without-poe">https://store.arduino.cc/usa/arduino-ethernet-rev3-without-poe</a> )	8-bit	AVR Enhanced RISC	2KiB	32KiB	C0
<b>PIC32MM0256GPM064</b> <a href="https://www.arrow.com/en/products/pic32mm0256gpm064-ipt/microchip-technology">https://www.arrow.com/en/products/pic32mm0256gpm064-ipt/microchip-technology</a>	32-bit	RISC	16KiB	64KiB	C0
<b>MSP430F5438A</b> ( <a href="https://www.ti.com/product/MSP430F5438A-EP">https://www.ti.com/product/MSP430F5438A-EP</a> )	16-bit	RISC	16KiB	256KiB	C1
<b>CC2640</b> ( <a href="https://www.ti.com/product/CC2640">https://www.ti.com/product/CC2640</a> )	32-bit	ARM Cortex-M3	20KiB	128KiB	C1
<b>Raspberry Pi Zero</b> ( <a href="https://www.raspberrypi.org/products/raspberry-pi-zero/">https://www.raspberrypi.org/products/raspberry-pi-zero/</a> )	32-bit	RISC	512MiB	Variable Size	C2

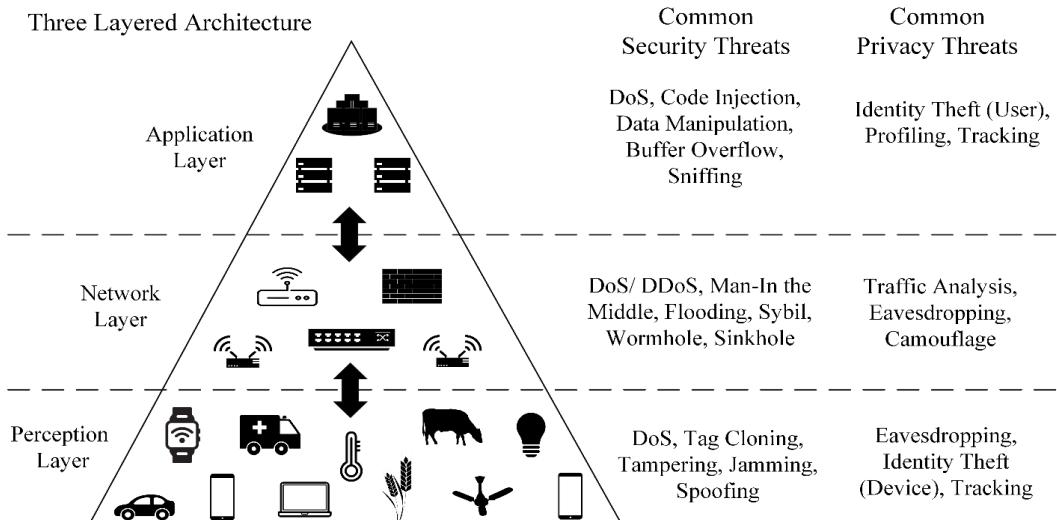


FIGURE 35. Privacy and Security Concern at Different Layers of IoT

technology; R&D on it was started in 2004 [277]. Lightweight cryptography is a new branch of cryptography [260] designed to run on the constrained devices like sensors, RFID tags, smart cards, healthcare devices and so on [278]. Lightweight cryptographic algorithms are aimed at providing adequate security to the devices, data, network, and application services with minimum resource requirements in IoT. The lightweight cryptographic algorithms should possess characteristics like smaller key size, smaller block size, and should take the minimum number of cycles to generate ciphertext [279]. Bassam J Mohd. et al. [280] analyzed different lightweight algorithms and defined features such as 1. The block size of the algorithm should be between 48-bit and 96-bit, 2. The number of rounds is limited to 16 rounds. These lightweight features offer better performance in terms of energy as compared to the traditional cryptographic algorithms.

To transmit smart grid power data securely, Yuancheng Li et al. [281] proposed a lightweight quantum encryption algorithm. The scheme consists of key distribution, authentication, and data transmission process. This scheme is based on the photon arrival time. Quantum Random Number Generator (QRNG) is used to generate the session key in the Quantum Key Distribution (QKD) terminal, and both the parties (Sender and Receiver) use the quantum channels to share the session keys during data transmission. Muhammad Usman et al. [282] proposed a lightweight symmetric Secure IoT (SIT) algorithm to encrypt the data. The block size used in the SIT algorithm is of 64-bit (size of the key was 64-bit) and used five rounds to produce ciphertext. The authors implemented the SIT algorithm on AT-mega 328 and found that the total amount of memory occupied by the algorithm is 22 bytes. It is observed that the time taken to encrypt and decrypt the data is 0.188 and 0.187 milliseconds, respectively. Yang Shi et al. [283] proposed a lightweight white-box encryption mechanism to protect the data against white-box attacks in wearable devices. The encryption and decryption are completely based on the random padding of the values. The algorithm comprised of 50-bit plain text and a 30-bit random value. It's been claimed that the amount of memory occupied by the algorithm is 95 KB.

## 2) Hardware Security

IoT devices are deployed in isolated locations and subjected to various attacks like node tampering, jamming, cloning, DoS, and node tracking attacks [284]. The emerging areas like Physical Unclonable Functions (PUFs) can be adopted to enhance the security level of security in IoT [285]. PUFs are functions that generate unique identifiers for the devices by using dedicated secure processors during the manufacturing phase, and often, these identifiers are used to generate secure secret keys and key management [286].

A lightweight Secure Communication protocol for IoT devices using PUFs was developed by the Priyanka Mall et al. [287]. The proposed protocol consists of the enrolment and authentication phase. During the enrolment phase, a set of challenges was sent by the server to the devices. On the other hand, devices use the PUF module to generate the

corresponding responses to the challenges sent by the server. Devices use the secure channels to send the responses, and the server is responsible for maintaining the repository of the challenge-response pairs of each node. The nodes use the corresponding identities, challenge and response pair, timestamp, and the random number to perform the authentications. The protocols designed for IoT should be efficient in terms of chip area usage, storage, processing, and energy. To address this problem, Muhammad Naveed Aman et al. [288] proposed a mutual authentication protocol using PUFs. A newly deployed IoT device exchanges the primary CRP with the server by making use of the time-based One-Time Password (OTP) algorithm, and the server maintains the identity and CRP of the devices in the repository. For the authentication process, the server generates the encrypted message ( $M = E \{Id, N, R\}$ ) using device identity (Id), nonce (N), and response (R), and sends the message, challenge, and Message Authentication Code (MAC) to the device. The IoT device generates the response using its PUF based on the challenges it received from the server. Further, generated and received responses are compared and verified against integrity, source, and freshness of the message

## 3) Intrusion Detection and Prevention System (IDPS)

IDS is a security mechanism used to identify malicious activities by analyzing the data packets in the network layer [289]. Generally, IDS is comprised of following stages 1. Monitoring Stage – Monitors the incoming traffic, 2. Analyzing Stage – which analyzes and generates the pattern based on the data received from the first stage and 3. Detection Stage – It performs anomaly detection or malicious node detection [290]. The IDS plays a vital role in mitigating the DoS, DDoS, Leakage, and Masquerade attacks in the host and networks [291]. To mitigate DoS, Probe, and Generic attacks in the IoT environment Vikash Kumar et al. [292] proposed a Unified Intrusion Detection System (IDS) for IoT Environments (UIDs). The authors used the UNSW-NB15 dataset, which contains 175341 records with 47 features. The authors used 13 features to train the different decision trees like C5, CHAID, CART, and QUEST models and generated approximately 359 rules. Based on the threshold confidence factor, UIDS selects the rules generated by each model to perform the analysis on the dataset. This IDS has shown the improved attack detection rate compare to decision tree models. Fang-Yie Leu et al. [293] used data mining and forensic technique to develop the Internal Intrusion Detection and Protection System (IIDPS). The system creates the personal profile of the users at the System Calls (SC) level to keep track of the usage habits of the users. The major components of the IIDPS are 1. SC Monitor and Filter – gathers the SCs submitted to the kernel and stores, 2. Mining Server – analyzes log data and users usage behavior patterns, 3. Detection Server – compares the user's behavior pattern and SC patterns collected to detect the malicious activity, and 4. Computational Grid – Mining and detection server are part of the computational grid to enhance the

speed of the computing. They claimed that the rate of accuracy is about 94.00%, and the response time approximately 0.50 second. Ayyaz-ul-Haq Qureshi et al. [294] proposed a Heuristic Intrusion Detection System for IoT to detect the attacks like DoS, Root-to-Local, probe and User-to-root attacks. The IDS is developed using the Random Neural Network (RNN). The authors used the NSL-KDD dataset with 125973 records and for analysis 29 features were used out of 41. The proposed RNN-IDS based IDS shown the better performance against the different ML algorithms with the accuracy rate up to 95.2%.

### C. Common Security and Privacy Threats

IoT devices and networks suffer from various security and privacy threats. This section describes common security and privacy threats. Table XIII depicts possible security

threats and mitigation techniques. Table XIV shows the probable privacy threats and mitigation techniques.

#### 1) Node Replication Attacks

Pinaki Sankar Chatterjee et al. [295] proposed a lightweight algorithm to detect the clone nodes in Cognitive Wireless Sensor Networks (CWSN). In this system, for every specified time interval, LEACH protocol elects the Cluster Head (CH), and that acts as a Monitoring Node (MN). The MN node keeps track of its neighbor's information and acts as an intermediate between nodes and base stations. Cognitive Sensor Nodes (CSNs) in the network take the local sensing decisions and sends the details to the MN. Further, MN forwards all this information to the base station with the unique identities of the nodes. To find cloned nodes in the network, the base station applies the Clone Node Detection (CND) algorithm, and it is based on

TABLE XIII  
DESCRIPTION OF COMMON SECURITY THREATS

Potential Security Threats	Description	Mitigation Techniques
Traffic Analysis	The aim of the attacker is to find IoT devices and monitor activities of them.	
Eavesdropping	The aim of the attacker is to steal the data while exchanging the data.	
Spoofing Attack	This attack purpose is to get the complete access to the application services by using valid device identity.	Protecting the Communication Channels using Cryptographic Algorithms, Strong Mutual Authentication Protocols
Man-in the Middle Attack	In this attack, communication between the two users is captured, modified and forwarded by the unauthorized user.	
Code Injection	Attacker aims at taking the control of the device by introducing the malicious code into an IoT device or application.	
Replay Attack	An attacker retransmits an old message to the authorized user and this message accepted as a new message by the IoT devices.	
Data Manipulation	In this attack, the aim of the attacker is to alter the data (compromise the data integrity) generated by the IoT devices.	Use of Strong Digital Signature Techniques and Cryptographic algorithms to secure Network
DoS/ DDoS	The aim of the attacker is to halt the services provided by the application by flooding the SYN and ACK packets.	
Jamming	Jamming is a kind of attack, which disrupts the existing communication at receiver side by transmitting the interfering wireless signals.	
Blackhole Attack	Malicious node pretends to be as the shortest path to the destination node and does not forwards the routing packets to its neighboring IoT devices (Drops the routing packets).	Analyzing the Packet Flow information in the Network, Cryptanalysis and Steganography Techniques to prevent Jamming Messages
Sybil Attack	An attacker introduces a malicious device into the IoT network and creates the illegitimate identity of it. This malicious device acts as a legitimate device by sharing its identity within the network. The data from the devices within the network should passes through the sybil device, even though multiple paths are available.	
Sinkhole Attack	A malicious device acts as a sink node and tries to capture whole traffic in the IoT network.	Analyzing the network flow information, Artificial Intelligence to identify malicious nodes, Identity based Authentication Schemes
Tag Cloning Attack	Attacker eavesdrop the tag-reader communication to creates the replica of a legitimate tag. The attacker reads the data from the compromised tag and write them into clone tag.	

TABLE XIV  
COMMON PRIVACY THREATS

Potential Privacy Threats	Description	Mitigation Techniques
Identification	An attacker tries to grab the personal information of the individual and devices such as name, address, etc.	Anonymization, Identity Management, and Local Processing
Localization & Tracking	An attacker target is to find the location of the person and device	Location Anonymization Mechanisms
Profiling	In this attack, an attacker generates the new information by collecting and relating the original data	Perturbing, Obfuscating and Anonymization Techniques
Lifecycle Transitions	An attacker tries to capture the private information when the control handover happening between the different phases of the service	Lifecycle Transitions Detection Algorithms, Private Information Locking Techniques
Inventory Attack	An attacker maintains the repository of persons and device existence and characteristics	Unauthorized Access Methods

the Cuckoo filtering technique. The cuckoo filter is a data structure, which contains the fingerprint of the original keys in the hash table (look-up table). The CND algorithm applied to the list of nodes present in the base station. If the node details already exist in the look-up-table such nodes are marked as cloned nodes otherwise the nodes information is added to the look-up table. Mojtaba Jamshidi et al. [296] used the location information and watchdog nodes to spot the replicated nodes. In this method, watchdogs use the buffer, and which comprises of unique node identity, time and current location details of the nodes. To identify the node replication, the watchdog nodes share the available information that exists in the buffer with the neighboring watchdogs in the encrypted form. Each watchdog verifies the shared and its own buffer content to find the replica node. If the different location information of a node at different time intervals is found in multiple buffers, such nodes are marked as the replica of a node. This procedure is based on the Euclidean distance between the nodes and the maximum predefined speed (threshold) of nodes. After identifying the replica node in the network watchdog report to the base station to propagate an alarm signal to the network. The implementation of the algorithm is done using the J-SIM simulator (Java Simulator), and results have shown that the false detection rate is less than 0.5%. Tassos Dimitriou et al. [297] proposed a lightweight and decentralized protocol to detect the replicated nodes. The detection scheme is based on random numbers (Nonce). As soon as two nodes in the network meet for the first time, both the nodes generate the random number and exchange them. Further, these nodes meet again, both the nodes request for the random number exchanged in their preceding meeting. If a node reply with an incorrect random number, then the node details are added to the quarantine list maintained by the nodes. Each node maintains the nonce list and claims list. The nonce list contains the values shared by the other nodes in the network, and this list helps in authenticating the nodes each other. The claim list comprises of nodes that are being used by the replicated nodes. The nodes exchange the quarantine list soon after the successful authentication of the nodes. The quarantine list is then added to the claims list, and once the number of claims exceeds the predefined threshold against the particular node, such nodes are considered as the replicated node.

## 2) Spoofing Attack

Monali Mavani et al. [298] proposed a resilient method against spoofing attacks in 6LoWPAN networks. The authors used the time to live and Attack Disruption Window (ADW) parameters to analyze the characteristics of the spoofing attack. The ADW length is based on the frequency with which the address of the node is changed. Therefore, the use of a temporary node address reduces the size of the ADW spoofing, which results in resilient against the spoofing attack. IPv6 addressing allows the node to change its address periodically. In this method, the authors used the temporary address, which allows the node to

register with the routing table whenever the address is changed. At this moment, the MAC address of the node is bound with the new IP address. However, the attacker MAC address is bound with the node's previous IP address. The address of the node spoofed by the attackers is no longer valid to participate in the construction of the routing table. Ning Wang et al. [299] proposed a two-step detection scheme to identify spoofing attacks for IoT in Mm-Wave and massive MIMO 5G communication. Firstly, Access Point (AP) extracts the MAC address and corresponding channel information of the devices and analyzes whether the anomaly is present or not by using the Virtual Channel Space (VCS). Further, the AP decides the type of anomaly (Spoofing or Others) using Logistic regression. VCS provides information about the number of virtual channels occupied by each MAC address. Therefore, the distinction between the spoofing traffic and normal traffic is based on the sparsity, overall energy, and path gains of the virtual channel. Monowar Hasan et al. [300] proposed a Contego-TEE framework to secure the IoT edge devices from the spoofing attacks where an adversary node sends the malicious signals to the controller and the framework developed using the embedded Linux kernel. The framework uses the trusted hardware and real-time characteristics of the system to safeguard the physical system from intrusions. In this system, the invariant checking mechanism and timing analysis are used to detect the spoofing attack, and this continuously monitors the actuation command executed by the tasks. Since the system is deterministic, the number of actuation requests are restricted during the design. Hence, if a task tries to contact the actuator more than predefined actuation requests within a given period, is considered as a threat. At this, the mechanism rejects the subsequent access requests from the tasks and avoids sending the actuation command to the hardware.

## 3) Man-in-the Middle Attack (MitMA)

Farouq Aliyua et al. [301] proposed a technique to detect and prevent MitMA in IoT. In this technique, the IoT system consists of Fog nodes and Intrusion Detection and Prevention System (IDPs) nodes. IDPs uses lightweight encryption algorithms to prevent MitMA and its variants, such as eavesdropping and packet modification. The primary job of the fog nodes is to provide the services to the IoT devices in lower layers. In this, all the packets except headers are encrypted using the AES algorithm, and the key exchange is using the Diffie-Hellman key exchange algorithm. IDP nodes are used to examine the Fog nodes periodically by sending the encrypted examine request (contains integer value). The IDPs define the rule, the receiver must decrypt the examined request, and the payload should be multiplied by the constant value two, and further, it should encrypt the message and resend to the IDPs. The IDPs declares the node is malicious when the response to examine requests failed to attain the criteria. IDPs maintains the log of such packets inter-arrival time. Cheng Li Zhengrui Qin et al. [302] demonstrated the

possible threats of MitMA on the OpenFlow control channel in Fog based IoT networks. They have proposed a lightweight mechanism to detect MitMA using bloom filters. In this, the bloom filter maintains a copy of the flow of packets, and this copy is used by the controllers. The controllers collect the packet flow from the bloom filter and checks for modification in the packet flow. MitMA is detected and presented by the controller when it found any modification in the packet flow. Now, this implementation is part of the OpenFlow protocol. To ensure the confidentiality and integrity of the data in the nuclear power plant. Mohamed Abdallah Elakrat et al. [303] developed an encryption-based security model on Field Programmable Gate Array (FPGA) to mitigate MitMA. This model uses the AES algorithm with a 128-bit key to encrypt the data. Behavioral analysis of the proposed model analyzed using the CORE9 tool. The FPGA based encryption model has shown better performance compared to the software-based encryption model.

#### 4) Eavesdropping

Stefano Iellamo et al. [304] proposed an intelligent jamming strategy to improve security against eavesdropping. In this, the security capacity of the channel is determined using Shannon's formula for channel capacity. The concept of the neutralized zone and Artificial Noise (AN) are used in this strategy. The neutralized zone is the region where all the eavesdroppers are deactivated, and AN is a noise generated and broadcasted by the IoT-Gateway to make the neutralized zone of some radius centered at the IoT-Gateway. IoT nodes can examine the surroundings and deactivate the eavesdropper falling inside the neutralized zone. In this strategy, potential eavesdroppers are deactivated using the jamming activities achieved using the In-band Full Duplex IoT gateway and cooperative helpers. Xiangyu Liu et al. [305] proposed a SecLight framework to mitigate eavesdropping in IoT. The devices communicate using Visible Light Communication (VLC) and, the required signal is sent through Light Emitting Diode (LED). To increase the security in VLC communication systems, the authors used the multipath redundancy, time reversal, and random choice techniques, which can make transmitted signals automatically focus on the authorized users while interference of the eavesdroppers present in the communication channel. The authors analyzed the effect of channel correlation and estimation error on the system using the additive noise, leaking factor, and time-varying error. This framework ensures the security in both multi-input single output and single input single output VLC. Kun Wang et al. [306] proposed a method to minimize the eavesdropping attack between the sensors and the controllers in the green cyber-physical transportation system. This communication model consists of a jammer, eavesdropper, and a communication channel. In this method, the secrecy of the communication is increased by using the feedback received from the controller, and the sensors are able to adjust its condition using this feedback. This method proved the existence of

Stackelberg Equilibrium between the sensors and jammers. To minimize the eavesdropping attack, the model allows the sensors to use dynamic transmission power.

#### 5) DoS/ DDoS Attack

Hongsong Chen et al. [307] proposed a method to detect DoS attack in low rate ZigBee networks using the Hilbert-Huang Transform (HHT) approach and trust of Intrinsic Mode Function (IMF) components. The trustworthiness of IMF is estimated by combining the correlation co-efficient and Kolmogorov-Smirnov (KS) approaches. Firstly, the HHT approach is applied to the normal traffic (NT) and attack traffic (AT) to obtain the IMF component of the NF and AT, respectively. Secondly, calculate the correlation coefficient and calculate probability using KS of AT with their original traffic. Further, analyze the trust of the IMF component based on the DoS detection rule. Rohan Doshi et al. [308] demonstrated the DoS detection for consumer IoT devices using the ML techniques. In this work, DoS detection was carried out at the packet level. The features like packet size, inter-packet arrival time, type of transport layer protocol, bandwidth, and destination IP address are considered to detect the DoS attack in the IoT network. The authors used five (KNN, LSVM, Decision Tree, Random Forest, and Neural Networks) different ML algorithms to distinguish the normal and DoS packets. Both the decision tree and KNN has shown the accuracy of 99%. Shruti Kajwadkar et al. [309] discussed the CoAP protocol and its DTLS security issues. They proposed an algorithm to detect DoS and DDoS attacks in IoT networks, communicating using CoAP. The detection algorithm was implemented at the 6LoWPAN border router gateway. The algorithm monitors the incoming traffic and decides whether the incoming packet is suspicious or not by using the black and graylisted IPs. The border gateway node extracts the incoming packet header and verifies whether the source IP belongs to blacklisted IP or not. The incoming packets were dropped if the packets belong to the blacklisted IP. The payload is considered malicious if the size of the payload is greater than the threshold payload. The gateway receives the packets from the same source with similar characteristics, then the DoS attack is found and the IP address of the device is added to the black list. DDoS attack is found when the gateway receives the packets from the different sources with the same features, then the different source IP addresses are added to the grey list. Anirudh M et al. [310] proposed a honeypot model to protect IoT devices against the DoS attack. Honeypot acts like a main IoT device and divert the attention of the attackers. In this, all the incoming packets should pass through the IDS to the Server. If IDS find malicious packets, such packets are directed to the honeypots and the information related to the packets (IP address and MAC address) are stored as logs. Each time, information of the packets is examined against the information present in the logs.

#### 6) Tag cloning Attack

Guo Yimin et al. [311] presented a Deterministic Cloned Tag Detection Protocol (DCTDP) to find the cloned tags. This protocol uses the Tree-Based anti-collision method to detect collisions. To detect cloned tags without revealing sensitive information, the protocol uses the pseudonym

update method. For distributed IoT, Prosanta Gope et al. [312] presented an RFID based authentication Architecture. This architecture uses the lightweight RFID-based authentication method, which provides secrecy, traceability, and anonymity of RFID-tags and secure localization. Kuljeet Kaur et al. [313] proposed an authentication protocol to mitigate replaying, tracking, and cloning attacks. This protocol, developed using the Elliptical Curve Cryptography (ECC) launches the mutual authentication between the tags and the servers.

## X. PLATFORMS FOR IOT

Generally, IoT is a collection of resource-constrained and resource-aware devices, which are communicating over the internet [314]. The Operating System (OS) is an essential component of any IoT device; it acts as an interface between the physical world and the user applications [315]. In general, Kernel, System Shell, and Utility Software are part of the OS. The Kernel is the core of the OS, which performs resource management. Using system shell, users can access the kernel. Software Utility is a collection of system software such as Assemblers, Debuggers, and Compilers to perform OS level tasks. The operating system like Raspbian, Ubuntu Mate, Snappy Ubuntu and Windows 10 IoT Core are used in the higher level IoT platforms like RaspberryPi, Beagle bone, OrangePi, Samsung Artik, Intel Edison and Galileo, and many more, which requires a large amount of memory.

### A. Platforms for Constrained Devices

The resource constraint devices make use of the Real-Time Operating System (RTOS), as it supports resource management, data management, safety, and security. The significant challenges like connectivity, memory management, and support for heterogeneous devices must be considered while designing the IoT applications on top of the RTOS.

#### 1) RIOT

RIOT is a microkernel-based operating system designed for IoT devices and embedded systems, which is written in C with an open-source license. RIOT uses the modular internet stack, and it provides a rich and consistent set of interfaces to develop IoT applications. Static & dynamic memory allocation, multi-threading, and tickles scheduling are the main feature of this OS. Tickles scheduler is used to minimize power consumption. It does not have a floating-point unit [315][316].

#### 2) FreeRTOS

This operating system used for real-time processing, which runs on the many of the microcontrollers, including advanced ARM CortexTM - Mx series, and it is written in C. This comes with the various real-time features like - scheduling, inter-process communication, timing analysis, and synchronization [317][318]. FreeRTOS-Memory Protection Unit (MPU) is used to secure the Kernel from the invalid execution of the tasks. It supports TCP/IP and a Lightweight TCP/IP (LwIP) stack, which is built on IPv6

and less power consuming protocols like 6LoWPAN, CoAP, and many more.

#### 3) ContikiOS

It was the first operating system developed for sensor networks with features like dynamic memory allocation, multi-threading, multi-tasking, protothreads, TCP/IP protocol, IPv4, IPv6, simple web server, telnet clients, and many more [319]. Because of these features, ContikiOS is used by many IoT applications such as smart city monitoring, industrial monitoring, alarm systems, construction site monitoring, and many more. This OS also supports the network simulator cooja, which enables developers to create and analyze the network and tests their code.

#### 4) TinyOS

This operating system was developed and managed by the TinyOS alliance [319][320]. TinyOS written in NesC language with an open-source license [321][322]. The size of the OS is around 400 bytes. TinyOS widely used by wireless sensor applications, personal networks, ubiquitous computing [323]. The CPU will be in sleep mode while the scheduler has no tasks to perform, and this facilitates the energy-saving feature. TinyOS consists of components, which eases the application developers to develop diverse applications. The components comprise of three computational entities, such as Commands, Events, and Tasks. A command is a request sent to a component to perform certain tasks. The tasks are consisting of reading and writing the sensor data. An event is an indication of the completion of the service.

#### 5) LiteOS

It is a lightweight OS designed for low power devices and suitable for applications like wearables, connected homes, V2V communications, etc. [324]. This OS was developed by Huawei in 2015, and it claimed that the memory occupied by this OS is less than 10KB [325]. The main features are zero-configuration, auto-discovery, fast boot, and hierarchical file system. The major components of the LiteOS are 1. LiteShell (Used with the user intervention) – provides the set of commands to perform file and process management and debugging 2. Lite Filesystem (LiteFS) – the sensors are considered as the files, and the file system developed using LiteC and 3. Kernel – which supports features like multithreading, scheduling and provides the callback functions to handle the events efficiently [326].

#### 6) VxWorks

This RTOS supports Intel, ARM, and POWER architectures of 32-bit and 64-bit microcontrollers. It supports IPv4/ IPv6 network stack and adds security features like secure booting, kernel hardening, and limited access control to the RTOS. In VxWorks, real-world problems can be solved using the programming languages like C and C++, along with this modern programming language like Python3 is also supported by the RTOS. It also assists the multimedia libraries such as OpenVG,

TABLE XV  
 COMPARISON OF RTOS's

Characteristics	ContikiOS	FreeRTOS	RIOT	TinyOS	LiteOS	VxWorks	Erika	Nut/OS
Version	V3.0, 2015	V10.2.1, 2019	V2.1	V2.1.2, 2012	V2.1, 2018	V7.0	V3.0	V5.2.4
Architecture	Modular	Microkernel RTOS	Microkernel RTOS	Monolithic	Modular	Modular	Modular	Modular
Hardware Support	AVR, TI CC2538, MSP430, MSP430x	ARM Cortex – A5, M3, M4, M7, MSP432, MSP430, Xilinx,	AVR, ARM, Cortex-M, X86, MSP430	Atmel AVR, ATmega1281, TI – MSP430, CC2420	Mica Z and IRIS	Intel X86, Fujitsu FR-V, MIPS	Atmel AVR, ARM Cortex M4, MSP 430, Mica Z	8-bit AVR, AVR32, ARM7, ARM9, ARM Cortex M3
Memory Management	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Power Management	Yes	Yes	Yes	Yes	Yes	-	-	-
Scheduling	Cooperative	Preemptive	Preemptive	Non-Preemptive	On Task Completion	Preemptive	Preemptive & Non-Preemptive	Non-Preemptive
Scheduling Algorithms	Priority (Round Robin)	Priority (Round Robin)	Priority (Round Robin)	FIFO	Priority (Round Robin)	Priority/ Adaptive	Dynamic (Fixed Priority, Earliest Deadline First)	-
Programming Model	Multi-Threading	Multi-Threading	Multi-Threading	Event Driven	Multi-Threading	Multi-Processing	Multi-Threading	Multi-Threading
Concurrency	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Hardware Requirements in KiB	Min-RAM: 2 Min-ROM: 30	Min-RAM: 10 Min-ROM: 12	Min-RAM: 1.5 Min-ROM: 5	Min-RAM: 1 Min-ROM: 4	Min-RAM: 4 Min-ROM: 128	Min-RAM: 2MiB Min-ROM: 512	Min-RAM: 2 Min-ROM:12	Min-RAM: 4 Min-ROM 128
Targeted Device Class	C0 and C1	C1 and C2	C1 and C2	C0 and C1	C0 and C1	-	C0 and C1	C0 and C1
Simulator	Cooja, Netsim	-	IoT-Lab	TOSSIM, PowerTossim	AVRORA	-	-	-
Database	Coffee, Raima	ITTA DB, Raima	Raima	TinyDB, Raima	-	Raima	-	-
IoT Protocol Stack	Yes	Yes	Yes	Yes	Yes	Yes	Partial	Partial
Security	TLS	TLS (WolfSSL)	-	Cipher Block Chaining	-	SSL/ TLS, Cryptography, SecureBOOT	-	TLS (WolfSSL)
Network Stack	uIP, RIME	-	OpenWSN	BLIP	-	Yes	-	Nut/Net (TCP/IP)
Applications	Smart Homes, Smart Agriculture, Smart City, Industry 4.0	Smart Homes, Industry 4.0, Smart City	Wearables, Smart Homes, Smart Agriculture, Smart City, Industry 4.0	Smart Homes, Smart Agriculture, Smart City, Industry 4.0	Smart Homes, V2V Agriculture, Smart City, Industry 4.0	Aerospace, Defense, Industry 4.0, Smart Health, Consumer Grid, Smart Fisheries	Smart Homes, Wearables	-

OpenGL, OpenCV, and audio-video handling libraries [327][328].

### 7) Erika Enterprise

Erika Enterprise is a hard-real-time open-source operating system suitable for low power and memory computers. It is written in C. ISO 17356 API (OSEK/VDX API) has been adopted by the Erika Enterprise, and it supports for 8-bit, 16-bit, and 32-bit microcontrollers. One of the essential features of this RTOS is the sharing of the stack between different tasks; this increases the efficient utilization of the memory by eradicating the use of dedicated stack for individual tasks. The use of RT-Druin with the eclipse plugin allows programmers to develop the graphical user interfaces [329].

### 8) Nut/OS

Nut/OS is an operating system for memory-constrained devices. The programming language like C and Lua scripting languages can be executed interactively. It supports TCP/ IP protocol stack along with the prioritized event handling. The transport and network layer protocols like TCP, UDP, IP, and ICMP are part of the TCP/IP protocol suite. Additionally, the operating system facilitates application layer protocols like FTP, HTTP, DNS, and DHCP. Nut/OS supports the transport layer security by using wolfSSL for embedded security. Currently, networking features like IPv6 and WLAN are not available in this operating system [330].

Table XV depicts the comparison between the RTOS's and Table XVI shows the IoT protocol stack supported by the different RTOS's [331][324][325].

### B. Other IoT Platforms for IoT Devices

These operating systems are designed for resource efficient IoT devices like smartphones, Televisions (TVs), smartwatches, and high-end IoT development boards. These types of OS's provide rich GUI and other libraries to carry out specific tasks.

#### 1) Tizen OS

Tizen is an operating system built from scratch to meet the requirements of the connected devices. Tizen operating system is used in the in-vehicle infotainment system, Smart TVs, smartwatches, mobiles, etc. Tizen IoT is headed or

headless binaries support for the embedded devices. Tizen IoT supports developers to implement IoT applications. Tizen IoT supports RaspberryPi3 and Samsung Artik family development boards [332].

#### 2) Android Things

Android Things is an android based operating system developed by Google, which uses Linux kernel; this helps to android developers to build IoT projects easily and swiftly. The OS uses the 32MB of RAM. Android Things supports a variety of hardware, such as RaspberryPi3, NXP i.MX7D, Qualcomm SDA624, and MediaTek MT8516. The weave is an application layer protocol, which comes with the OS; this protocol can be implemented over IPv6 using BLE, WiFi, Cellular, and thread communication standards [333].

#### 3) Ubuntu Core

Ubuntu Core is the Linux operating system for embedded devices. It facilitates optimized and reliable updates to develop IoT applications with advanced features. It is easy to deploy with the minimum hardware requirement (256MB of RAM and 512MB of flash memory). It is tamper-resistant and hardened against memory corruption. By default, this OS does not provide any GUI. However, Wayland or Mir options can be used. Ubuntu Core supports Artik, Intel, Qualcomm, Beaglebone, Snickerdoodle, and RaspberryPi family devices. Ubuntu Core establishes communication using Ethernet, WiFi, and BLE [334].

Neil Klingensmith et al. [335] proposed a lightweight hypervisor called Hermes to accomplish the runtime necessities of the embedded software. Always, RTOS schedulers do not provide swift responses to the I/O events. Hermes architecture offers the solution for these kinds of problems by placing the abstract layer between the I/O devices and software services. The authors have used the ARM Cortex M7 CPU to measure the Interrupt Service Routine (ISR) latency (Time between starting of ISR execution and starting of user-space data processing) and measuring of the ISR latency captured in FreeRTOS. This architecture has the advantages like enabling and disabling of the interrupt sources and allows the switching between the network port and the serial port. Rafael Raymundo Belleza et al. [336] analyzed the performance of the RTOS

TABLE XVI  
RTOS IoT PROTOCOL STACK

Operating System	IoT Protocol Stack							
	802.14.5	6LowPAN	RPL	MQTT	CoAP	DDS	XMPP	WebSocket
ContikiOS	✓	✓	✓	✓	✓	✗	✗	✗
FreeRTOS	✓	✓	✓	✓	✓	✗	✗	✗
RIOT	✓	✓	✓	✓	✓	✗	✗	✗
TinyOS	✓	✓	✓	✓	✓	✗	✗	✗
LiteOS	✓	✓	✓	✓	✓	✗	✗	✗
VxWorks	✓	-	-	✓	✓	✗	✗	✗

**Note:** Real-time operating systems like Erika Enterprise and Nut/OS provides the Networking stack (USB Stack, IPv6, Wireless LAN, and Security). However, these OS's need to provide IoT protocols stack

like FreeRTOS, RIOT, Zephyr, and μC/OS-II and μC/OS-III. The analysis was performed based on the parameters like task switching time, semaphore passing time, time taken to exchange messages between the tasks, time to get and release memory region, and time is taken to activate a task within the interrupt service routine. After the successful analysis, the authors pointed out the following observations 1. μC/OS-II and μC/OS-III have well suited for safety-critical applications like avionics, industry 4.0 and healthcare devices in smart health, 2. Privacy and security are the primary concern, then the open source RTOS's like FreeRTOS, RIOT and Zephyr are the best choices for the IoT devices, and 3. Zephyr is a good choice for the application, which needs an immediate response.

## XI. OPEN ISSUES AND FUTURE DIRECTIONS

In earlier sections, a comprehensive study on system-level aspects of IoT, such as IoT architectures, communication standards, application layer protocols, computing paradigms, security, and real-time operating systems, are discussed. Attributed to the above study, current research issues and future directions have been addressed in this section.

### A. IoT Architecture

Due to the lack of standards in IoT, many IoT architectures have been developed like Computing, Secure, IoT-SDN, SOA, Middleware Architectures, etc. An in-depth examination of many IoT architectures is unexplored. Several Computing architectures/ platforms have been introduced, but they are not interoperable, and this becomes a big issue when the multiple applications are integrating. Integration becomes difficult due to the use of abundant services from various platforms, protocols, and file formats. Creating seamless connectivity between different applications is of the researcher's interest. IoT devices are increasing exponentially, and these devices are mobile in nature. Developing a scalable and mobility supported architecture for IoT is still a challenging issue. IoT devices may perish due to power constraints and go offline due to infrastructure constraints (Bandwidth constraints). Design and development of self-healing and self-configurable architectures are much needed. Recent days, IoT middleware architectures have gained the researcher's interest. IoT middleware architectures should provide real-time and secure services to the mission control applications like health monitoring, disaster management systems, etc.

### B. Communication

Signaling plays a vital role in the exchanging of information between the devices. The data need to be exchanged between node A and B swiftly and reliably in IoT networks. The communication standards like WiFi and Thread uses the 2.4GHz radio, due to this signal may interfere. This increases the level of noise and weakens the signal strength. IoT devices require a persistent internet connection to share the information at the global level, which consists of Open Ports. Unauthorized access to Open Ports leads to attacks like malicious code execution, DoS,

DDoS, and code injection. Maintaining the bandwidth in IoT networks is essential to provide the QoS to the end-users. The efficient use of available bandwidth in IoT networks is also essential. In the future, the requirement for the bandwidth may increase as the number of devices and data increases. Currently, WiFi is used widely in IoT networks to share the information, which does not use any power management techniques. The low range communication technologies like BLE, Zigbee, and Z-Wave consumes less power as compared to WiFi communication standard. Power management is significant since the majority of the IoT devices are operating in remote areas. Due to the power and bandwidth constraints, sometimes devices may go offline. Continuous monitoring of the IoT devices is still challenging. High signaling overhead in 5G networks increases the energy consumption in IoT devices. The small cells, such as picocells, microcells, and femtocells, bring the QoS capabilities into 5G IoT networks. Also, the use of small cells in 5G IoT networks provides the extended battery life of IoT devices; this is due to the low latency in the novel 5G IoT wireless network environment.

### C. Application Layer Protocols

Study of IoT application layer protocols at different environments (resource-aware and resource-constrained) with different load and diversity of data (Text, Audio, Images, and Videos) is needed. Generally, IoT application layer protocols use Transport Layer Security (TLS and DTLS), which increases the latency, consumes more resources due to the expensive handshaking procedure; TLS and DTLS are easily susceptible to the Man in the Middle attacks. Developing a strong and lightweight authentication mechanism is still challenging. The usage of protocols like QUIC and HTTP 2.0 is increasing; adopting these protocols into the IoT networks is a researcher's concern. Moreover, the computational operations are moving from Cloud to the Edge of the network. This transformation raises challenges like data management and security issues. The huge number of lightweight encryption algorithms are reaching the IoT community. However, testing and validating the performance of these algorithms by incorporating them into the IoT application layer protocols is also a major research concern. Privacy issues like anonymous communication still require major attention. Most of the application layer protocols are compatible with traditional communication standards like WiFi and local area networks, and these protocols are needed to be compatible with all the communication standards. XMPP does not support QoS and enabling the QoS feature in it is a researcher's concern. IoT application layer protocols use First in First out data delivery strategies, but priority-based data delivery strategies need to be incorporated to address the real-time IoT applications. Due to the in-band binary data transfer limitation, XMPP does not support for sharing of the high-resolution images and videos.

### D. Computing Paradigms

Cloud and fog computing architectures suffer from server downtime, limited control on the services, and security.

Making edge node work in cooperation with the cloud and fog computing architectures is a challenging task. Data analytics, immediate response, intelligence, and security are the significant characteristics of edge computing architecture. All the time, edge computing devices are not sophisticated servers. Recent days, ML, DL, and AI algorithms are common in IoT applications. Using these algorithms at the edge level turns the edge node into an intelligent edge node. Implementing energy-efficient ML, DL, and AI algorithms in emerging applications like Autonomous Vehicles, V2V communication, Intelligent robots, Outlier detection, AR & AR, and others to make effective decisions is also a challenging task. Edge computing should support for the various QoS features like scalability, mobility, scheduling, and resource utilization. Improved energy-efficient device management and monitoring techniques that need to be implemented at the edge level are beneficial. Privacy & security is one of the serious issues in IoT, and researchers should focus on developing trust management techniques in edge devices, which reduces the latency in massive IoT networks. IoT users are increasing, and edge computing should take care of user and resource allocation algorithms to provide reliable and scalable services. 5G technical standards are still evolving, and edge computing should create an environment to develop 5G applications.

#### **E. Security**

Numerous security mechanisms have been proposed to protect the ecosystem of IoT. In the real world, it is a tedious task to reduce latency and increase the computation speed of the security algorithms in IoT applications. Most of the IoT applications are controlled using mobile applications, and securing such mobile applications is also a major concern. Many Lightweight cryptographic algorithms have been introduced to protect data from the attackers. The use of hybrid encryption algorithms, simplifying the key generation mechanisms, provides a high level of security and enhances the computational speed. Different middleware technologies were introduced to reduce the data processing and communication overhead in the IoT networks. Authorization, privacy, encryption, securing the web interfaces, and adequate software protection are the major security concerns in the IoT middleware. IoT is a self-organizing network; any device can become a part of the network at any point in time. Design and development of novel privacy-preserving lightweight device authentication schemes are essential to enhance the quality of the services to legitimate devices. Now the trend is moving toward the use of biological characteristics (fingerprints, iris, and facial data) to authenticate the users, which poses new challenges. Employing of ML and DL algorithms are increasing in the area of security to predict DoS and DDoS attacks. These algorithms require authentic IoT traffic datasets to build efficient security solutions. One of the major issues with the IoT deployment is radio jamming since most of the communication is based on wireless standards. An attacker may disrupt the data

transmission by propagating the powerful radio within the proximity of an IoT device. BLE devices in IoT networks may be susceptible to various attacks; this is due to the use of predictable smaller secret keys. Blockchain technology has an impact on applications like supply-chain, Industry 4.0, and smart grids. However, the decentralized architecture of the blockchain results in increasing in high energy depletion, storage, and computation overhead. Energy-efficient blockchain solutions need to be developed for the resource-constrained IoT devices. Future IoT communication is 5G, which poses various security challenges due to its network architecture and heterogeneous access. Design and development of authentication, identity management, trust models, and privacy protection algorithms increase the security features in 5G IoT networks.

#### **F. IoT Platforms**

Most of the RTOS's use the preemptive scheduling algorithm and have the drawbacks 1. The instructions of the high priority tasks may not be executed in the given time, and 2. Sometimes the low priority tasks may wait for an indefinite time to execute. The scheduling algorithms developed for traditional OS's are not suitable for real-time applications. Designing and developing context-aware scheduling algorithms is significant. The recent networking technologies like IPv6, RPL, 6LoWPAN, CoAP, and MQTT are part of most of the OS. Adding new IoT protocols like AMQP, XMPP, DDS, WebSocket, and others to the RTOS is still the researcher's concern. The interoperability issue arises due to the availability of wide range OS's (RIOT, FreeRTOS, ContikiOS, TinyOS, ERIKA, LiteOS, MantisOS, Ubuntu, Android, Microsoft Windows, MACOS and etc.), Data Structures, programming languages, protocols, and IoT architecture. Solving interoperability at the different levels of platforms is also a researcher's concern. The services provided by the schedulers to the I/O events are not deterministic. Enhancing the quality of the services to the I/O events is a researcher's motive. RTOS supports programming languages like C and C++. However, the most widely used programming languages are Java, Python, and JavaScript. Developing the RTOS's compatible with these languages is also one of the prime concerns; this brings the new application features to the IoT applications.

#### **XII. CONCLUSION**

In this survey, a basic understanding of IoT, along with the current research trends in IoT technologies as of 2020, is presented. Next, the characteristics and requirements of the emerging IoT applications are provided. Then, the current challenges in the IoT system design are analyzed. Later, the present research trends in IoT architectures are articulated and also described the issues that are necessary to be addressed in the future. Subsequently, the commonly used communication standards in IoT, along with future research directions, are presented. Further, investigation on the characteristics of IoT application layer protocols and further

requirements of the IoT application layer protocols are provided. Next, the necessity of the computing paradigms in IoT, along with detailed analysis of Cloud, Cloudlets Fog, and Edge computing architecture in detail, are deliberated. Then, the security features of IoT are emphasized and presented a detailed study on security issues, current research, and future scope in IoT security. Further, an analysis of RTOS and the available protocol stack for IoT is provided. At last, a detailed discussion on open research challenges and future directions is provided. From the analysis, it has been noticed that the foremost requirements like reliability, scalability, adaption, context-awareness, interoperability, embedded intelligence, privacy & security issues need to be addressed more precisely. Currently, researcher's focusing on solving the interoperability at the cloud and cloudlets. However, solving the interoperability issues at the edge level brings lots of benefits to the IoT applications. Our goal is to design and develop an energy-efficient, interoperable, and secure reconfigurable edge node for IoT applications.

### Acknowledgment

The authors would like to thank the Indian National Academy of Engineering (INAE), Gurgaon, and R V College of Engineering (RVCE), Bengaluru, for their financial support. The authors would also like to thank the P C Panchariya, Director, CSIR-CEERI, Pilani, for his encouragement and constant support.

## REFERENCES AND FOOTNOTES

### A. REFERENCES

- [1] N. Bălău and S. Utz, "Information sharing as strategic behaviour: the role of information display , social motivation and time pressure," *Behav. Inf. Technol.*, vol. 0, no. 0, pp. 1–17, 2017.
- [2] H. Liu, D. Han, and D. Li, "Fabric-iot: A Blockchain-Based Access Control System in IoT," *IEEE Access*, vol. 8, pp. 18207–18218, 2020.
- [3] P. Sethi and S. R. Sarangi, "Internet of Things: Architectures, Protocols, and Applications," *J. Electr. Comput. Eng.*, vol. 2017, pp. 1–25, 2017.
- [4] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao, "A Survey on Internet of Things: Architecture , Enabling Technologies , Security and Privacy , and Applications," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1125–1142, 2017.
- [5] V. Miori and D. Russo, "Improving Life Quality for the Elderly through the Social Internet of Things ( SLoT )," in *2017 Global Internet of Things Summit (GloTS)*, Geneva, 2017, pp. 1–6.
- [6] A. Rizzardi, "Security in Internet of Things: Networked Smart Objects," UNIVERSITA DEGLI STUDI DELL'INSUBRIA, 2016.
- [7] W. Zhou, Y. Jia, A. Peng, Y. Zhang, and P. Liu, "The effect of IoT New Neatures on Security and Privacy: New Threats, Existing Solutions, and Challenges yet to be Solved," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1606–1616, 2019.
- [8] M. A. A. Da Cruz, J. J. P. C. Rodrigues, J. Al-Muhtadi, V. V. Korotaev, and V. H. C. De Albuquerque, "A Reference Model for Internet of Things Middleware," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 871–883, 2018.
- [9] "GlobalDots: 41.6 Billion IoT Devices will be Generating 79.4 Zettabytes of Data in 2025." [Online]. Available: <https://www.globalsdots.com/blog/41-6-billion-iot-devices-will-be-generating-79-4-zettabytes-of-data-in-2025>. [Accessed: 20 Jan-2019].
- [10] "IEEE Xplore Digital Library," Accessed Dec. 06, 2019. [Online]. Available <https://ieeexplore.ieee.org/Xplore/home.jsp>.
- [11] 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. Surv. Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.
- [12] M. M. Dhanvijay and S. C. Patil, "Internet of Things: A survey of enabling technologies in healthcare and its applications," *Comput. Networks*, vol. 153, pp. 113–131, Apr. 2019.
- [13] J. Ding, M. Nemati, C. Ranaweera, and J. Choi, "IoT Connectivity Technologies and Applications: A Survey," *IEEE Access*, vol. 8, pp. 67646–67673, 2020.
- [14] S. Balaji, K. Nathani, and R. Santhakumar, "IoT Technology, Applications and Challenges: A Contemporary Survey," *Wirel. Pers. Commun.*, vol. 108, no. 1, pp. 363–388, Sep. 2019.
- [15] A. Nauman, Y. A. Qadri, M. Amjad, Y. Bin Zikria, M. K. Afzal, and S. W. Kim, "Multimedia Internet of Things: A Comprehensive Survey," *IEEE Access*, vol. 8, pp. 8202–8250, 2020.
- [16] S. Ravidas, A. Lekidis, F. Paci, and N. Zannone, "Access control in Internet-of-Things: A survey," *J. Netw. Comput. Appl.*, vol. 144, no. July, pp. 79–101, Oct. 2019.
- [17] F. Azzedin and M. Ghaleb, "Internet-of-Things and Information Fusion: Trust Perspective Survey," *Sensors*, vol. 19, no. 8, p. 1929, Apr. 2019.
- [18] C. Puliafito, E. Mingozzi, F. Longo, A. Puliafito, and O. Rana, "Fog Computing for the Internet of Things," *ACM Trans. Internet Technol.*, vol. 19, no. 2, pp. 1–41, Apr. 2019.
- [19] M. El-hajj, A. Fadlallah, M. Chamoun, and A. Serhrouchni, "A Survey of Internet of Things (IoT) Authentication Schemes," *Sensors*, vol. 19, no. 5, p. 1141, Mar. 2019.
- [20] K. A. P. da Costa, J. P. Papa, C. O. Lisboa, R. Munoz, and V. H. C. de Albuquerque, "Internet of Things: A survey on machine learning-based intrusion detection approaches," *Comput. Networks*, vol. 151, pp. 147–157, Mar. 2019.
- [21] J. Hou, L. Qu, and W. Shi, "A survey on internet of things security from data perspectives," *Comput. Networks*, vol. 148, pp. 295–306, Jan. 2019.
- [22] I. Ud Din, M. Guizani, B.-S. Kim, S. Hassan, and M. Khurram Khan, "Trust Management Techniques for the Internet of Things: A Survey," *IEEE Access*, vol. 7, no. c, pp. 29763–29787, 2019.
- [23] G. Mohindru, K. Mondal, and H. Banka, "Internet of Things and data analytics: A current review," *WIREs Data Min. Knowl. Discov.*, vol. 10, no. 3, pp. 1–27, May 2020.
- [24] J. Marietta and B. Chandra Mohan, "A Review on Routing in Internet of Things," *Wirel. Pers. Commun.*, vol. 111, no. 1, pp. 209–233, Mar. 2020.
- [25] Y. Harbi, Z. Aliouat, S. Harous, A. Bentaleb, and A. Refoufi, "A Review of Security in Internet of Things," *Wirel. Pers. Commun.*, vol. 108, no. 1, pp. 325–344, Sep. 2019.
- [26] L. E. Lima, B. Y. L. Kimura, and V. Rosset, "Experimental Environments for the Internet of Things: A Review," *IEEE Sens. J.*, vol. 19, no. 9, pp. 3203–3211, May 2019.
- [27] Y. Lu and L. Da Xu, "Internet of things (IoT) cybersecurity research: A review of current research topics," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2103–2115, 2019.
- [28] C. C. Sabin, "A Survey on Architecture, Protocols and Challenges in IoT," *Wirel. Pers. Commun.*, vol. 112, no. 3, pp. 1383–1429, Jun. 2020.
- [29] B. Li *et al.*, "A wearable IoT aldehyde sensor for pediatric asthma research and management," *Sensors Actuators B Chem.*, vol. 287, no. January, pp. 584–594, May 2019.
- [30] D. Sehrawat and N. S. Gill, "Smart Sensors: Analysis of Different Types of IoT Sensors," in *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*, 2019, no. Icoei, pp. 523–528.
- [31] Qingping Chi, Hairong Yan, Chuan Zhang, Zhibo Pang, and Li Da Xu, "A Reconfigurable Smart Sensor Interface for Industrial WSN in IoT Environment," *IEEE Trans. Ind. Informatics*, vol. 10, no. 2, pp. 1417–1425, May 2014.

- [32] S. Bansal and D. Kumar, "IoT Ecosystem: A Survey on Devices, Gateways, Operating Systems, Middleware and Communication," *Int. J. Wirel. Inf. Networks*, vol. 27, no. 3, pp. 340–364, Sep. 2020.
- [33] J. Koo, S.-R. Oh, and Y.-G. Kim, "Device Identification Interoperability in Heterogeneous IoT Platforms," *Sensors*, vol. 19, no. 6, p. 1433, Mar. 2019.
- [34] H. Aftab, K. Gilani, J. Lee, L. Nkenyereye, S. Jeong, and J. Song, "Analysis of identifiers on IoT platforms," *Digit. Commun. Networks*, no. January, May 2019.
- [35] R. S. Paul J. Leach, Michael Mealling, "A Universally Unique IDentifier (UUID) URN Namespace Status," *The Internet Society*, vol. 42, no. 4, p. 1, 2005.
- [36] "Identifiers in Internet of Things (IoT)," *ALLIANCE FOR INTERNET OF THINGS INNOVATION*, 2018. [Online]. Available: <https://euagenda.eu/upload/publications/identifiers-in-internet-of-things-iot.pdf>. [Accessed: 29-Aug-2020].
- [37] Y. Bin Zikria, S. W. Kim, O. Hahm, M. K. Afzal, and M. Y. Aalsalem, "Internet of Things (IoT) Operating Systems Management: Opportunities, Challenges, and Solution," *Sensors*, vol. 19, no. 8, p. 1793, Apr. 2019.
- [38] K. Bierzyński, A. Escobar, and M. Eberl, "Cloud, fog and edge: Cooperation for the future?," *2017 2nd Int. Conf. Fog Mob. Edge Comput. FMEC 2017*, pp. 62–67, 2017.
- [39] B. Umar, "Evaluation of IoT Device Management Tools," in *ACCSE 2018 : The Third International Conference on Advances in Computation, Communications and Services*, 2018, pp. 15–21.
- [40] T. Perumal, S. K. Datta, and C. Bonnet, "IoT Device Management Framework for Smart Home Scenarios," *2015 IEEE 4th Glob. Conf. Consum. Electron.*, pp. 54–55, 2015.
- [41] L. M. Xing Xiaojiang, Wang Jianli, "Services and Key Technologies of the Internet of Things," *ZTE Communication*, 2010. [Online]. Available: [https://www.zte.com.cn/global/about/magazine/zte-communications/2010/2/en\\_127/186201.html](https://www.zte.com.cn/global/about/magazine/zte-communications/2010/2/en_127/186201.html). [Accessed: 11-Aug-2020].
- [42] E. Buenrostro, D. Cyrus, T. Le, V. Emamian, and E. Buenrostro, "Security of IoT Devices Security of IoT Devices," *J. Cyber Secur. Technol.*, vol. 00, no. 00, pp. 1–13, 2018.
- [43] R. Bonetto, N. Bui, V. Lakkundi, A. Olivereau, A. Serbanati, and M. Rossi, "Secure Communication for Smart IoT Objects : Protocol Stacks , Use Cases and Practical Examples," *2012 IEEE Int. Symp. a World Wireless, Mob. Multimed. Networks*, pp. 1–7, 2012.
- [44] A. G. Khairnar and D. A. Birari, "IoT ('Connected Life') and its Use in Different Applications : A Survey," *MVP J. Eng. Sci.*, vol. 1, no. June, 2018.
- [45] G. Mokhtari and Q. Zhang, "A New Layered Architecture for Future Big Data-driven Smart Homes," *IEEE Access*, vol. PP, p. 1, 2019.
- [46] N. Barodawala, B. Makwana, Y. Punjabi, and C. Bhatt, "Home Automation Using IoT," in *Springer International Publishing AG 2018*, vol. 30, no. January, 2018, pp. 219–242.
- [47] A. Tiele, S. Esfahani, and J. Covington, "Design and Development of a Low-Cost, Portable Monitoring Device for Indoor Environment Quality," *J. Sensors*, vol. 2018, pp. 1–14, 2018.
- [48] B. Min and S. J. Park, "A Smart Indoor Gardening System Using IoT Technology," in *Lecture Notes in Electrical Engineering*, vol. 474, 2018, pp. 683–687.
- [49] A. Based, I. Froiz-míguez, T. M. Fernández-caramés, P. Fraguallas, and L. Castedo, "Design , Implementation and Practical Evaluation of Sensor Nodes," *Sensors*, pp. 1–42, 2018.
- [50] S. Pandya *et al.*, "Smart Home Anti-Theft System: A Novel Approach for Near Real-Time Monitoring and Smart Home Security for Wellness Protocol," *Appl. Syst. Innov.*, vol. 1, no. 4, p. 42, Oct. 2018.
- [51] S. Y. Y. Tun, S. Madanian, and F. Mirza, "Internet of things (IoT) applications for elderly care: a reflective review," *Aging Clin. Exp. Res.*, no. 0123456789, Apr. 2020.
- [52] S. Majumder *et al.*, "Smart Homes for Elderly Healthcare—Recent Advances and Research Challenges," *Sensors*, vol. 17, no. 11, p. 2496, Oct. 2017.
- [53] A. Abdalgawad, K. Yelamarthi, and A. Khattab, "IoT-Based Health Monitoring System for Active and Assisted Living," in *Lecture Notes of the Institute for Computer Sciences*, no. October, 2017, pp. 11–20.
- [54] S. B. Baker, W. Xiang, and I. Atkinson, "Internet of Things for Smart Healthcare: Technologies, Challenges, and Opportunities," *IEEE Access*, vol. 5, pp. 26521–26544, 2017.
- [55] B. Klímová and K. Kuča, "Internet of things in the assessment, diagnostics and treatment of Parkinson's disease," *Health Technol. (Berl.)*, vol. 9, no. 2, pp. 87–91, Mar. 2019.
- [56] P. Kakria, N. K. Tripathi, and P. Kitipawang, "A Real-Time Health Monitoring System for Remote Cardiac Patients Using Smartphone and Wearable Sensors," *Int. J. Telemed. Appl.*, vol. 2015, pp. 1–11, 2015.
- [57] M. Haghi, K. Thurow, and R. Stoll, "Wearable Devices in Medical Internet of Things: Scientific Research and Commercially Available Devices," *Healthc. Inform. Res.*, vol. 23, no. 1, p. 4, 2017.
- [58] S. I. Lakkis, "IoT based Emergency and Operational Services in Medical Care Systems," in *2017 Internet of Things Business Models, Users, and Networks, Copenhagen*, 2017.
- [59] S. Velten, J. Leventon, N. Jager, and J. Newig, "What Is Sustainable Agriculture? A Systematic Review," *Sustainability*, vol. 7, no. 6, pp. 7833–7865, Jun. 2015.
- [60] A. Borthakur and P. Singh, "Agricultural research in india: an exploratory study," *Int. J. Soc. Sci. Interdiscip. Res.*, vol. 1, no. 9, pp. 59–74, 2012.
- [61] P. Agriculture, "Precision Agriculture: Tomorrow 's Technology for Today 's Farmer," *Food Process. Technol.*, vol. 6, no. 8, pp. 8–13, 2015.
- [62] M. Reinecke and T. Prinsloo, "The influence of drone monitoring on crop health and harvest size," in *2017 1st International Conference on Next Generation Computing Applications (NextComp)*, 2017, pp. 5–10.
- [63] S. Vatari, A. Bakshi, and T. Thakur, "Green House by using IOT and Cloud computing," *2016 IEEE Int. Conf. Recent Trends Electron. Inf. Commun. Technol.*, pp. 246–250, 2016.
- [64] L. Pan, M. Xu, L. Xi, and Y. Hao, "Research of Livestock Farming IoT System Based on RESTful Web Services," *2016 5th Int. Conf. Comput. Sci. Netw. Technol.*, pp. 113–116, 2016.
- [65] A. Zear, P. Kumar Singh, and Y. Singh, "Intelligent Transport System: A Progressive Review," *Indian J. Sci. Technol.*, vol. 9, no. 32, Aug. 2016.
- [66] D. Jaisinghani and P. A. M. Bongale, "Real-Time Intelligent Traffic Light Monitoring and Control System to Predict Traffic Congestion Using Data Mining and WSN," *Int. J. Eng. Res. Technol.*, vol. 4, no. 12, pp. 33–38, 2015.
- [67] P. Kumar, V. Singh, and D. Reddy, "Advanced Traveler Information System for Hyderabad City," *IEEE Trans. Intell. Transp. Syst.*, vol. 6, no. 1, pp. 26–37, Mar. 2005.
- [68] C.-H. Liao and C. Chen, "Use of Advanced Traveler Information Systems for Route Choice: Interpretation Based on a Bayesian Model," *J. Intell. Transp. Syst.*, vol. 19, no. 3, pp. 316–325, Jul. 2015.
- [69] B. Singh, A. Gupta, and S. Suman, "Advanced Traveller Information System for Chandigarh City using GIS," in *Transportation Planning and Implementation Methodologies for Developing Countries, IIT Bombay*, 2014, no. December.
- [70] B. Sreelatha, E. Alakananda, B. Bhavya, and S. H. Prasad, "IOT APPLIED TO LOGISTICS USING INTELLIGENT CARGO," *Int. Res. J. Eng. Technol.*, vol. 5, no. 4, pp. 191–193, 2018.
- [71] Y. Tsang, K. Choy, C. Wu, G. Ho, H. Lam, and P. Koo, "An IoT-based cargo monitoring system for enhancing operational effectiveness under a cold chain environment," *Int. J. Eng. Bus. Manag.*, vol. 9, p. 184797901774906, Jan. 2017.
- [72] K. Naseer, Q. Universiti, H. Abdullah, and U. Teknologi, "A Survey on Intelligent Transportation Systems," *Middle-East J. Sci. Res.*, no. July, pp. 629–642, 2015.
- [73] V. D. Khairna, "V2V COMMUNICATION SURVEY - ( WIRELESS TECHNOLOGY )," *Int.J.Computer Technol. Appl.*,

- [74] S. Kuttu, S. Fallah, K. Katsaros, M. Dianati, F. McCullough, and A. Mouzakitis, "A Survey of the State-of-the-Art Localization Techniques and Their Potentials for Autonomous Vehicle Applications," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 829–846, 2018.
- [75] S. M. Kumar, "ENHANCING ROAD SAFETY USING VEHICLE TO VEHICLE," *IJCESR*, vol. 5, no. 4, pp. 13–18, 2018.
- [76] Epicor, "What is Industry 4.0—the Industrial Internet of Things (IIoT)?".
- [77] A. Badri, B. Boudreau-Trudel, and A. S. Souissi, "Occupational health and safety in the industry 4.0 era: A cause for major concern?", *Saf. Sci.*, vol. 109, no. August 2017, pp. 403–411, Nov. 2018.
- [78] H. Lasi, P. Fettke, H.-G. Kemper, T. Feld, and M. Hoffmann, "Industry 4.0," *Bus. Inf. Syst. Eng.*, vol. 6, no. 4, pp. 239–242, Aug. 2014.
- [79] K. Santos, E. Loures, F. Piechnicki, and O. Canciglieri, "Opportunities Assessment of Product Development Process in Industry 4.0," *Procedia Manuf.*, vol. 11, no. June, pp. 1358–1365, 2017.
- [80] H. Fatorachian and H. Kazemi, "Impact of Industry 4.0 on supply chain performance," *Prod. Plan. Control*, vol. 0, no. 0, pp. 1–19, Jan. 2020.
- [81] M. Owonibi, E. Petzold, and B. Koenig-ries, *Innovations in Enterprise Information Systems Management and Engineering*, vol. 245. Cham: Springer International Publishing, 2016.
- [82] D. Bigos, "From reactive to proactive quality management with IoT," *IBM*. [Online]. Available: <https://www.ibm.com/blogs/internet-of-things/quality-management-iot/#:~:text=With%20IoT%2C%20the%20ability%20to,specifications%20promises%20significant%20cost%20reductions>. [Accessed: 25-Jun-2020].
- [83] M. A. Al Mamun and M. R. Yuce, "Sensors and Systems for Wearable Environmental Monitoring Toward IoT-Enabled Applications: A Review," *IEEE Sens. J.*, vol. 19, no. 18, pp. 7771–7788, Sep. 2019.
- [84] G. Mois, S. Folea, and T. Sanislav, "Analysis of Three IoT-Based Wireless Sensors for Environmental Monitoring," *IEEE Trans. Instrum. Meas.*, vol. 66, no. 8, pp. 2056–2064, Aug. 2017.
- [85] M. Singh and R. Kaur, *International Conference on Innovative Computing and Communications*, vol. 55, no. January 2019. Singapore: Springer Singapore, 2019.
- [86] S.-C. Hsia, Y.-J. Chang, and S.-W. Hsu, "Remote monitoring and smart sensing for water meter system and leakage detection," *IET Wirel. Sens. Syst.*, vol. 2, no. 4, pp. 402–408, Dec. 2012.
- [87] R. K. Kodali and P. S. Ramakrishna, "Modern sanitation technologies for smart cities," in *2017 IEEE Region 10 Humanitarian Technology Conference (R10-HTC)*, 2017, vol. 2018-Janua, pp. 706–709.
- [88] H. Zheng, K. Xiong, P. Fan, and Z. Zhong, "Data Analysis on Outdoor–Indoor Air Quality Variation: Buildings' Producing Dynamic Filter Effects," *IEEE Syst. J.*, vol. 13, no. 4, pp. 4386–4397, Dec. 2019.
- [89] J. J. Wellington and P. Ramesh, "Role of Internet of Things in disaster management," in *2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*, 2017, vol. 2018-Janua, pp. 1–4.
- [90] A. Sinha, P. Kumar, N. P. Rana, R. Islam, and Y. K. Dwivedi, "Impact of internet of things (IoT) in disaster management: a task-technology fit perspective," *Ann. Oper. Res.*, vol. 283, no. 1–2, pp. 759–794, Dec. 2019.
- [91] A. J. Meijer, J. R. Gil-Garcia, and M. P. R. Bolívar, "Smart City Research," *Soc. Sci. Comput. Rev.*, vol. 34, no. 6, pp. 647–656, Dec. 2016.
- [92] B. N. Silva, M. Khan, and K. Han, "Towards sustainable smart cities: A review of trends, architectures, components, and open challenges in smart cities," *Sustain. Cities Soc.*, vol. 38, no. January, pp. 697–713, Apr. 2018.
- [93] R. I. Ogie, P. Perez, and V. Dignum, "Smart infrastructure: an emerging frontier for multidisciplinary research," *Proc. Inst. Civ. Eng. - Smart Infrastruct. Constr.*, vol. 170, no. 1, pp. 8–16, Mar. 2017.
- [94] A. M. Selim, P. H. A. Yousef, and M. R. Hagag, "Smart infrastructure by (PPPs) within the concept of smart cities to achieve sustainable development," *Int. J. Crit. Infrastructures*, vol. 14, no. 2, p. 182, 2018.
- [95] S. Durga, S. Surya, and E. Daniel, "SmartMobiCam: Towards a New Paradigm for Leveraging Smartphone Cameras and IaaS Cloud for Smart City Video Surveillance," in *2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI)*, 2018, no. Icoei, pp. 1035–1038.
- [96] C. Arcadius Tokognon, B. Gao, G. Y. Tian, and Y. Yan, "Structural Health Monitoring Framework Based on Internet of Things: A Survey," *IEEE Internet Things J.*, vol. 4, no. 3, pp. 619–635, Jun. 2017.
- [97] M. A. Adegbeye, W.-K. Fung, and A. Karnik, "Recent Advances in Pipeline Monitoring and Oil Leakage Detection Technologies: Principles and Approaches," *Sensors*, vol. 19, no. 11, p. 2548, Jun. 2019.
- [98] Z. Chen, X. Zhou, X. Wang, L. Dong, and Y. Qian, "Deployment of a Smart Structural Health Monitoring System for Long-Span Arch Bridges: A Review and a Case Study," *Sensors*, vol. 17, no. 9, p. 2151, Sep. 2017.
- [99] T. O. Olasupo, "Wireless Communication Modeling for the Deployment of Tiny IoT Devices in Rocky and Mountainous Environments," *IEEE Sensors Lett.*, vol. 3, no. 7, pp. 1–4, Jul. 2019.
- [100] X. Zhao *et al.*, "Active health monitoring of an aircraft wing with an embedded piezoelectric sensor/actuator network: II. Wireless approaches," *Smart Mater. Struct.*, vol. 16, no. 4, pp. 1218–1225, Aug. 2007.
- [101] Jishnu Sekhar Mukherje and Maheswara Poothireddy, "IoT: Shaping the Next Epoch of the Aviation Industry," *HCL Technologies Ltd.*, 2017. [Online]. Available: <https://www.hcltech.com/blogs/iot-shaping-next-epoch-aviation-industry>. [Accessed: 03-Jul-2020].
- [102] E. Pantano and H. Timmermans, "What is Smart for Retailing?," *Procedia Environ. Sci.*, vol. 22, pp. 101–107, 2014.
- [103] M. Hauser, S. A. Günther, C. M. Flath, and F. Thiesse, "Towards Digital Transformation in Fashion Retailing: A Design-Oriented IS Research Study of Automated Checkout Systems," *Bus. Inf. Syst. Eng.*, vol. 61, no. 1, pp. 51–66, Feb. 2019.
- [104] T. Chatzidimitris *et al.*, "A location history-aware recommender system for smart retail environments," *Pers. Ubiquitous Comput.*, Feb. 2020.
- [105] S. Kakran and S. Chanana, "Smart operations of smart grids integrated with distributed generation: A review," *Renew. Sustain. Energy Rev.*, vol. 81, no. July 2017, pp. 524–535, Jan. 2018.
- [106] L. Fabbiano, G. Vacca, and G. Dinardo, "Smart water grid: A smart methodology to detect leaks in water distribution networks," *Measurement*, vol. 151, p. 107260, Feb. 2020.
- [107] J. Saebi and D. Thanh Nguyen, "Distributed demand response market model for facilitating wind power integration," *IET Smart Grid*, vol. 3, no. 3, pp. 394–405, Jun. 2020.
- [108] G. Tang, Y. Han, J. Wang, and Q. Zhao, "Smart Meter Data Analytics for Distribution Network," in *2018 37th Chinese Control Conference (CCC)*, 2018, vol. 2018-July, no. 4, pp. 8882–8887.
- [109] Shanzhi Chen, Hui Xu, Dake Liu, Bo Hu, and Hucheng Wang, "A Vision of IoT: Applications, Challenges, and Opportunities With China Perspective," *IEEE Internet Things J.*, vol. 1, no. 4, pp. 349–359, Aug. 2014.
- [110] J. Silva, J. Rodrigues, J. Al-Muhtadi, R. Rabélo, and V. Furtado, "Management Platforms and Protocols for Internet of Things: A Survey," *Sensors*, vol. 19, no. 3, p. 676, Feb. 2019.
- [111] F. Ullah, M. A. Habib, M. Farhan, S. Khalid, M. Y. Durrani, and S. Jabbar, "Semantic interoperability for big-data in heterogeneous IoT infrastructure for healthcare," *Sustain. Cities*

- Soc.*, vol. 34, pp. 90–96, Oct. 2017.
- [112] “47 Per cent of IoT Problems Can Be Solved If We Unlock Interoperability,” 2019. [Online]. Available: <https://iot.electronicsforu.com/content/whats-new/iot-problems-solution-unlock-interoperability/>.
- [113] “Specifications,” *Open Connectivity Foundation (OFC)*. [Online]. Available: <https://www.etsi.org/standards-search#page=1&search=Interoperability&title=1&etsiNumber=0&content=0&version=0&onApproval=1&published=1&historicAl=1&startDate=1988-01-15&endDate=2020-08-24&harmonized=0&keyword=IoT&TB=&stdType=&frequency=&mandate=&collectio>. [Accessed: 24-Aug-2020].
- [114] A. Bröring *et al.*, “Enabling IoT Ecosystems through Platform Interoperability,” *IEEE Softw.*, vol. 34, no. 1, pp. 54–61, 2017.
- [115] H. Derhamy, J. Eliasson, and J. Delsing, “IoT Interoperability - On-Demand and Low Latency Transparent Multiprotocol Translator,” *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1754–1763, 2017.
- [116] S. Yang and R. Wei, “Tabdoc approach: An information fusion method to implement semantic interoperability between IoT devices and users,” *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1972–1986, 2019.
- [117] “IEC role in the IoT,” *International Electrotechnical Commission*, 2016. [Online]. Available: [https://www.iec.ch/about/brochures/pdf/technology/iec\\_role\\_IoT.pdf](https://www.iec.ch/about/brochures/pdf/technology/iec_role_IoT.pdf). [Accessed: 24-Aug-2020].
- [118] “OMA Device Management Protocol,” *Open Mobile Alliance*, 2016. [Online]. Available: [http://www.openmobilealliance.org/release/DM/V1\\_3-20160524-A/OMA-TS-DM\\_Protocol-V1\\_3-20160524-A.pdf](http://www.openmobilealliance.org/release/DM/V1_3-20160524-A/OMA-TS-DM_Protocol-V1_3-20160524-A.pdf). [Accessed: 24-Aug-2020].
- [119] J. Kang, H. Ju, M. Choi, and J. W. Hong, “OMA DM Based Remote RF Signal Monitoring of Mobile Devices for QoS Improvement,” 2007, pp. 76–87.
- [120] G. Yoon, D. Choi, J. Lee, and H. Choi, “Management of IoT Sensor Data Using a Fog Computing Node,” *J. Sensors*, vol. 2019, pp. 1–9, 2019.
- [121] M. Haus, A. Y. Ding, and J. Ott, “Managing IoT at the Edge: The Case for BLE Beacons,” in *SMARTOBJECTS 2017 - Proceedings of the 3rd Workshop on Experiences with the Design and Implementation of Smart Objects, co-located with MobiCom 2017*, 2017, pp. 41–46.
- [122] H. Cai, B. Xu, L. Jiang, and A. V. Vasilakos, “IoT-Based Big Data Storage Systems in Cloud Computing: Perspectives and Challenges,” *IEEE Internet Things J.*, vol. 4, no. 1, pp. 75–87, Feb. 2017.
- [123] M. Marjani *et al.*, “Big IoT Data Analytics: Architecture , Opportunities , and Open Research Challenges,” *IEEE Access*, vol. 5, pp. 5247–5261, 2017.
- [124] M. Abu-Elkheir, M. Hayajneh, and N. Ali, “Data Management for the Internet of Things: Design Primitives and Solution,” *Sensors*, vol. 13, no. 11, pp. 15582–15612, Nov. 2013.
- [125] L. Lengyel, P. Ekler, T. Ujj, T. Balogh, and H. Charaf, “SensorHUB : An IoT Driver Framework for Supporting Sensor Networks and Data Analysis,” *Int. J. Distrib. Sens. Networks*, vol. 11, no. 7, p. 454379, Jul. 2015.
- [126] M. A. Abbasi, Z. A. Memon, T. Q. Syed, J. Memon, and R. Alshboul, “Addressing the Future Data Management Challenges in IoT: A Proposed Framework,” *IJACSA*, vol. 8, no. 5, pp. 197–207, 2017.
- [127] N. S. S, “Repeated Data Management Framework for IoT: A Case Study on Weather Monitoring and Forecasting,” *2018 4th Int. Conf. Recent Adv. Inf. Technol.*, pp. 1–7, 2018.
- [128] M. Amadeo, C. Campolo, A. Molinaro, and G. Ruggeri, “IoT Data Processing at the Edge with Named Data Networking,” in *European Wireless 2018, 24th European Wireless Conference, Catania, Italy*, 2018, pp. 38–43.
- [129] L. Feng, “A Multi-tier Data Reduction Mechanism for IoT Sensors,” 2017.
- [130] M. U. Iftikhar, G. S. Ramachandran, P. Bollansee, D. Weyns, and D. Hughes, “DeltaloT: A Self-Adaptive Internet of Things Exemplar,” in *2017 IEEE/ACM 12th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS)*, 2017, pp. 76–82.
- [131] J. Lee, M. Debnath, A. Patki, M. Hasan, and C. Nicopoulos, “Hardware-Based Online Self-Diagnosis for Faulty Device Identification in Large-Scale IoT Systems,” in *2018 IEEE/ACM Third International Conference on Internet-of-Things Design and Implementation (IoTDI)*, 2018, pp. 96–104.
- [132] H. Huang, L. Zhao, H. Huang, and S. Guo, “Machine Fault Detection for Intelligent Self-Driving Networks,” *IEEE Commun. Mag.*, vol. 58, no. 1, pp. 40–46, Jan. 2020.
- [133] M. Singh and G. Baranwal, “Quality of Service (QoS) in Internet of Things,” in *2018 3rd International Conference On Internet of Things: Smart Innovation and Usages (IoT-SIU)*, 2018, pp. 1–6.
- [134] B. Montruccio, S. A. Moraru, C. Pozna, and J. Vales Alonso, “Mobility for the Internet of Things,” *Mob. Inf. Syst.*, vol. 2018, 2018.
- [135] T. N. Gia, A. M. Rahmani, T. Westerlund, P. Liljeberg, and H. Tenhunen, “Fog computing approach for mobility support in internet-of-things systems,” *IEEE Access*, vol. 6, pp. 36064–36082, 2018.
- [136] S. M. Chun and J. T. Park, “Mobile CoAP for IoT mobility management,” in *2015 12th Annual IEEE Consumer Communications and Networking Conference, CCNC 2015*, 2015, pp. 283–289.
- [137] M. B. Yassein, S. Aljawarneh, and W. Al-Sarayrah, “Mobility management of Internet of Things: Protocols, challenges and open issues,” in *Proceedings - 2017 International Conference on Engineering and MIS, ICEMIS 2017*, 2018, vol. 2018-Janua, pp. 1–8.
- [138] D. Wu, D. I. Arkhipov, E. Asmare, Z. Qin, and J. A. McCann, “UbiFlow: Mobility management in urban-scale software defined IoT,” in *Proceedings - IEEE INFOCOM*, 2015, vol. 26, pp. 208–216.
- [139] H. L. Fu, P. Lin, H. Yue, G. M. Huang, and C. P. Lee, “Group mobility management for large-scale machine-to-machine mobile networking,” *IEEE Trans. Veh. Technol.*, vol. 63, no. 3, pp. 1296–1305, 2014.
- [140] W. Bin Qaim and O. Ozkasap, “DRAW: Data replication for enhanced data availability in iot-based sensor systems,” in *Proceedings - 2018 IEEE 16th International Conference on Dependable, Autonomic and Secure Computing, IEEE 16th International Conference on Pervasive Intelligence and Computing, IEEE 4th International Conference on Big Data Intelligence and Computing and I*, 2018, pp. 762–769.
- [141] Q. Wei, Z. Jin, L. Li, and G. Li, “Lightweight semantic service modelling for IoT: An environment-based approach,” *Int. J. Embed. Syst.*, vol. 8, no. 2–3, pp. 164–173, 2016.
- [142] J. Yao and N. Ansari, “Fog Resource Provisioning in Reliability-Aware IoT Networks,” *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8262–8269, 2019.
- [143] H. M. Al-Kadhim and H. S. Al-Raweshidy, “Energy efficient and reliable transport of data in cloud-based IoT,” *IEEE Access*, vol. 7, pp. 64641–64650, 2019.
- [144] J. Yuan and X. Li, “A Reliable and Lightweight Trust Computing Mechanism for IoT Edge Devices Based on Multi-Source Feedback Information Fusion,” *IEEE Access*, vol. 6, pp. 23626–23638, 2018.
- [145] J. Adu Ansere, G. Han, H. Wang, C. Choi, and C. Wu, “A Reliable Energy Efficient Dynamic Spectrum Sensing for Cognitive Radio IoT Networks,” *IEEE Internet Things J.*, vol. 6, no. 4, pp. 6748–6759, 2019.
- [146] D. Arellanes and K. K. Lau, “Decentralized Data Flows in Algebraic Service Compositions for the Scalability of IoT Systems,” in *IEEE 5th World Forum on Internet of Things, WF-IoT 2019 - Conference Proceedings*, 2019, pp. 668–673.
- [147] H. Guo, J. Ren, D. Zhang, Y. Zhang, and J. Hu, “A Scalable and Manageable IoT Architecture based on Transparent Computing,” *J. Parallel Distrib. Comput.*, 2017.
- [148] J. Canedo and A. Skjellum, “Adding scalability to Internet of Things gateways using parallel computation of edge device data,” in *2016 IEEE High Performance Extreme Computing Conference, HPEC 2016*, 2016, pp. 1–5.

- [149] A. Tiurlikova, N. Stepanov, and K. Mikhaylov, "Method of Assigning Spreading Factor to Improve the Scalability of the LoRaWan Wide Area Network," *Int. Congr. Ultra Mod. Telecommun. Control Syst. Work.*, vol. 2018-Novem, pp. 1–4, 2019.
- [150] M. A. Razzaq, M. A. Qureshi, and S. Ullah, "Security Issues in the Internet of Things ( IoT ): A Comprehensive Study," vol. 8, no. 6, 2017.
- [151] V. Gazis *et al.*, "Short Paper: IoT: Challenges , Projects , Architectures," *2015 18th Int. Conf. Intell. Next Gener. Networks*, pp. 145–147, 2015.
- [152] C. Zhong, Z. Zhu, and R. Huang, "Study on the IOT Architecture and Gateway Technology," *2015 14th Int. Symp. Distrib. Comput. Appl. Bus. Eng. Sci.*, pp. 196–199, 2015.
- [153] K. K. Goyal, "A Literature Survey on Internet of Things ( IoT )," *Int. J. Adv. Netw. Appl.*, vol. 09, no. 6, pp. 3663–3668, 2018.
- [154] A. B. Ostermaier, "The Web as an Interface to the Physical World," Technische Universität München, Germany, 2015.
- [155] B. Vogel and D. Gkouskos, "An Open Architecture Approach: Towards Common Design Principles for an IoT Architecture," in *Proceedings ofECSA '17, Canterbury*, pp. 85–88.
- [156] S. Cirani *et al.*, "A scalable and self-configuring architecture for service discovery in the internet of things," *IEEE Internet Things J.*, vol. 1, no. 5, pp. 508–521, 2014.
- [157] P. Hu, H. Ning, L. Chen, and M. Daneshmand, "An Open Internet of Things System Architecture Based on Software-Defined Device," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2583–2592, Apr. 2019.
- [158] Y. Xu and A. Helal, "Scalable Cloud-Sensor Architecture for the Internet of Things," *IEEE Internet Things J.*, vol. 3, no. 3, pp. 285–298, 2016.
- [159] C. Sarkar, A. U. Akshay, R. V. Prasad, A. Rahim, R. Neisse, and G. Baldini, "DIAT: A scalable distributed architecture for IoT," *IEEE Internet Things J.*, vol. 2, no. 3, pp. 230–239, 2015.
- [160] Aqeel-ur-Rehman, K. Mehmood, and A. Baksh, "Communication Technology That Suits IoT - A Critical Review," in *Communications in Computer and Information Science*, vol. 366, no. December, Springer Berlin Heidelberg, 2013, pp. 14–25.
- [161] W. E. I. Yu *et al.*, "A Survey on the Edge Computing for the Internet of Things," *IEEE Access*, vol. 6, pp. 6900–6919, 2018.
- [162] M. S. M. Gismalla, M. Faiz, and L. Abdullah, "Device to Device Communication for Internet of Things Ecosystem: An overview," *Int. J. Integr. Eng.*, vol. 9, no. 4, pp. 118–123, 2017.
- [163] N. S. Swamy, S. Nayak, and V. M. N, "Analysis on IoT Challenges, Opportunities, Applications and Communication Models," *Int. J. Adv. Eng. Manag. Sci.*, vol. 2, no. 4, 2016.
- [164] "Internet of Things: IEEE Standards Enabling Products with Real-World Applications," *IEEE*. [Online]. Available: <https://standards.ieee.org/initiatives/iot/stds.html>. [Accessed: 24-Aug-2020].
- [165] "The Internet of Things," *IETF*. [Online]. Available: <https://www.ietf.org/topics/iot/#:~:text=Several IETF working groups%2C spanning,build and specify interoperable systems>. [Accessed: 24-Aug-2020].
- [166] L. Chettri and R. Bera, "A Comprehensive Survey on Internet of Things (IoT) Toward 5G Wireless Systems," *IEEE Internet Things J.*, vol. 7, no. 1, pp. 16–32, Jan. 2020.
- [167] V. Cherneva and J. L. Trahan, "A Secure and Efficient Parallel-Dependency RFID Grouping-Proof Protocol," *IEEE J. Radio Freq. Identif.*, vol. 4, no. 1, pp. 14–23, Mar. 2020.
- [168] F. Chetouane, "An Overview on RFID Technology Instruction and Application," *IFAC-PapersOnLine*, vol. 48, no. 3, pp. 382–387, 2015.
- [169] L. Li, H. Xiaoguang, C. Ke, and H. Ketai, "The applications of WiFi-based Wireless Sensor Network in Internet of Things and Smart Grid," in *2011 6th IEEE Conference on Industrial Electronics and Applications*, 2011, pp. 789–793.
- [170] K. E. Jeon, J. She, P. Soonsawad, and P. C. Ng, "BLE Beacons for Internet of Things Applications : Survey , Challenges , and Opportunities," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 811–828, 2018.
- [171] M. R. C, M. E. P. Computing, and M. E. P. Computing, "STUDY ON ZIGBEE TECHNOLOGY," *2011 3rd Int. Conf. Electron. Comput. Technol.*, vol. 6, pp. 297–301.
- [172] M. B. Yassein, W. Mardini, and A. Khalil, "Smart Homes Automation using Z-wave Protocol," *2016 Int. Conf. Eng. MIS*, pp. 1–6, 2016.
- [173] H. Haas, "LiFi is a paradigm-shifting 5G technology," *Rev. Phys.*, vol. 3, no. October 2017, pp. 26–31, Nov. 2018.
- [174] "Li-Fi, the New Frontier in Communications," *OpenMind, BBVA Group*, 2016. [Online]. Available: <https://www.bbvaopenmind.com/en/technology/digital-world/li-fi-the-new-frontier-in-communications/>. [Accessed: 19-Aug-2020].
- [175] P. Ferrari, A. Flammini, D. Marioli, S. Rinaldi, and E. Sisinni, "On the Implementation and Performance Assessment of a WirelessHART Distributed Packet Analyzer," *IEEE Trans. Instrum. Meas.*, vol. 59, no. 5, pp. 1342–1352, May 2010.
- [176] J. Kim and J. Song, "A Secure Device-To-Device Link Establishment Scheme for LoRaWAN," *IEEE Sens. J.*, vol. 18, no. 5, pp. 2153–2160, 2018.
- [177] K. Mekki, E. Bajic, F. Chaxel, and F. Meyer, "Overview of Cellular LPWAN Technologies for IoT Deployment: Sigfox, LoRaWAN, and NB-IoT," in *IEEE International Conference on Pervasive Computing and Communications Workshops, PerCom Workshops 2018*, 2018, pp. 197–202.
- [178] C. B. Mwakwata, H. Malik, M. M. Alam, Y. Le Moullec, S. Parand, and S. Mumtaz, "Narrowband Internet of Things (NB-IoT): From Physical (PHY) and Media Access Control (MAC) Layers Perspectives," *Sensors*, vol. 1, pp. 1–34, 2019.
- [179] Y. Li, X. Cheng, Y. Cao, D. Wang, and L. Yang, "Smart choice for the smart grid: Narrowband internet of things (NB-IoT)," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 1505–1515, 2018.
- [180] M. Bembe, A. Abu-Mahfouz, M. Masonta, and T. Ngqondi, "A survey on low-power wide area networks for IoT applications," *Telecommun. Syst.*, vol. 71, no. 2, pp. 249–274, Jun. 2019.
- [181] "NB-Fi – The IoT Standard," *WavIOT*. [Online]. Available: <https://waviot.com/technology/nb-fi-specification/>. [Accessed: 19-Aug-2020].
- [182] M. Petru and J. Vysock, "Proposall and Implementation of Probe for Sigfox Technology," in *AETA 2018 - Recent Advances in Electrical Engineering and Related Sciences: Theory and Application*, vol. 554, Springer International Publishing, 2020, pp. 844–849.
- [183] F. C. De Oliveira, J. J. P. C. Rodrigues, R. A. L. Rabelo, and S. Mumtaz, "Performance delay comparison in random access procedure for NB-IoT, LoRa, and SigFox IoT protocols," in *Proceedings of the 2019 IEEE 1st Sustainable Cities Latin America Conference, SCLA 2019*, 2019, no. ii, pp. 0–5.
- [184] A. Lavric, A. I. Petrariu, and V. Popa, "Long Range SigFox Communication Protocol Scalability Analysis under Large-Scale, High-Density Conditions," *IEEE Access*, vol. 7, pp. 35816–35825, 2019.
- [185] B. S. Chaudhari, M. Zennaro, and S. Borkar, "LPWAN Technologies: Emerging Application Characteristics, Requirements, and Design Considerations," *Futur. Internet*, vol. 12, no. 3, p. 46, Mar. 2020.
- [186] "Building data-driven cities on smart streetlight infrastructure," *Telensa*. [Online]. Available: <https://www.telensa.com/>. [Accessed: 19-Aug-2020].
- [187] C.-X. Wang, J. Bian, J. Sun, W. Zhang, and M. Zhang, "A Survey of 5G Channel Measurements and Models," *IEEE Commun. Surv. Tutorials*, vol. 20, no. 4, pp. 3142–3168, 2018.
- [188] K. Sharma and N. Dhir, "A Study of Wireless Networks : WLANs , WPANs , WMANs , and WWANs with Comparison," vol. 5, no. 6, pp. 7810–7813, 2014.
- [189] P. Sciencedirect, N. Manoj, and P. Kumar, "ScienceDirect The Internet of Things : Insights into the building blocks , component interactions , and architecture layers," *Procedia Comput. Sci.*, vol. 132, pp. 109–117, 2018.
- [190] M. A. A. Cruz, J. J. P. C. Rodrigues, P. Lorenz, P. Solic, J. Almuhtadi, and V. H. C. Albuquerque, "solutions A proposal for Bridging Application Layer Protocols to HTTP on IoT

- Solutions,” *Futur. Gener. Comput. Syst.*, 2019.
- [191] “OASIS Committee Categories: IoT/M2M,” *OASIS*. [Online]. Available: [https://www.oasis-open.org/committees/tc\\_cat.php?cat=iot](https://www.oasis-open.org/committees/tc_cat.php?cat=iot). [Accessed: 24-Aug-2020].
- [192] “CoAP Protocol Binding,” *oneM2M*, 2016. [Online]. Available: [https://www.onem2m.org/images/files/deliverables/TS-0008-CoAP\\_Protocol\\_Binding-V1\\_3\\_2.pdf](https://www.onem2m.org/images/files/deliverables/TS-0008-CoAP_Protocol_Binding-V1_3_2.pdf). [Accessed: 01-Sep-2020].
- [193] “HTTP Protocol Binding,” *oneM2M*, 2016. [Online]. Available: [https://www.onem2m.org/images/files/deliverables/TS-0009-HTTP\\_Protocol\\_Binding-V1\\_5\\_1.pdf](https://www.onem2m.org/images/files/deliverables/TS-0009-HTTP_Protocol_Binding-V1_5_1.pdf). [Accessed: 01-Sep-2020].
- [194] “MQTT Protocol Binding,” *oneM2M*, 2019. [Online]. Available: [https://www.onem2m.org/images/files/deliverables/Release3/TS-0010-MQTT\\_protocol\\_binding-V3\\_0\\_1.pdf](https://www.onem2m.org/images/files/deliverables/Release3/TS-0010-MQTT_protocol_binding-V3_0_1.pdf). [Accessed: 01-Sep-2020].
- [195] J. H. Huh, D. H. Kim, and J.-D. Kim, “Extension of OneM2M Protocol Binding Considering Service Mapping,” 2016, pp. 1273–1281.
- [196] B. N. Silva, M. Khan, and K. Han, “Internet of Things : A Comprehensive Review of Enabling Technologies , Architecture , and Challenges,” *IETE Tech. Rev.*, vol. 0, no. 0, pp. 1–16, 2017.
- [197] A. Luoto and K. Systä, “Fighting network restrictions of request-response pattern with MQTT,” *IET Softw.*, vol. 12, no. 5, pp. 410–417, Oct. 2018.
- [198] Shinho Lee, Hyeonwoo Kim, Dong-kweon Hong, and Hongtaek Ju, “Correlation analysis of MQTT loss and delay according to QoS level,” in *The International Conference on Information Networking 2013 (ICOIN)*, 2013, pp. 714–717.
- [199] M. Collina, M. Bartolucci, A. Vanelli-coralli, and G. E. Corazza, “Internet of Things Application Layer Protocol Analysis over Error and Delay prone Links,” in *2014 7th Advanced Satellite Multimedia Systems Conference and the 13th Signal Processing for Space Communications Workshop (ASMS/SPSC)*, 2014, pp. 398–404.
- [200] D. Soni and A. Makwana, “A Survey on MQTT : A Protocol of Internet of Things (IoT),” in *International Conference on Telecommunication, Power Analysis and Computing Techniques (ICTPACT - 2017), Bharath Institute of Higher Education and Research*, 2017, no. April.
- [201] IBM and Eurotech, “MQTT V3.1 Protocol Specification.” pp. 1–42, 2010.
- [202] A. Banks and R. Gupta, “Mqtt-v3.1.1- Specification,” *OASIS Standard*, no. October. 2014.
- [203] R. Bryce and G. Srivastava, “The Addition of Geolocation to Sensor Networks,” in *ICSOFT 2018 - Proceedings of the 13th International Conference on Software Technologies*, 2019, pp. 762–768.
- [204] M. Asim, “A Survey on Application Layer Protocols for Internet of Things ( IoT ),” *Int. J. Adv. Res. Comput. Sci.*, vol. 8, no. 3, pp. 996–1000, 2017.
- [205] V. Karagiannis, P. Chatzimisios, F. Vazquez-gallego, and J. Alonso-zarate, “A Survey on Application Layer Protocols for the Internet of Things Research motivation,” *Trans. IoT Cloud Comput.*, pp. 1–10, 2015.
- [206] R. A. Rahman, “Security analysis of IoT protocols : A focus in CoAP,” *2016 3rd MEC Int. Conf. Big Data Smart City*, pp. 1–7, 2016.
- [207] J. Granjal, E. Monteiro, and J. Sa Silva, “Security for the internet of things: A survey of existing protocols and open research issues,” *IEEE Commun. Surv. Tutorials*, vol. 17, no. 3, pp. 1294–1312, 2015.
- [208] G. Tanganeli, C. Vallati, and E. Mingozi, “CoAPthon : Easy Development of CoAP-based IoT Applications with Python,” *2015 IEEE 2nd World Forum Internet Things*, pp. 63–68, 2015.
- [209] and R. T. Peter Saint-Andre, Kevin Smith, *XMPPT: The Definitive Guide, Building Real-Time Applications with Jabber Technologies*, First Edit. O'Reilly Media, 2009.
- [210] I. E. Kho, “OPEN SOURCE XMPP PERFORMANCE ANALYSIS: USING LARGE VOLUME TRAFFIC FROM honeypot Sensor,” in *Proceedings of the International Conference on Innovation, Entrepreneurship and Technology, BSD City, Indonesia*, 2016, no. November 2015.
- [211] P. Saint-Andre, “Streaming XML with Jabber/XMPP,” *IEEE Internet Comput.*, vol. 9, no. 5, pp. 82–89, Sep. 2005.
- [212] H. Chu and W. Chen, “A Novel Design of Instant Messaging Service Extended from Short Message Service with XMPP,” *Int. J. Comput. Networks Appl.*, vol. 2, no. 1, pp. 35–40, 2015.
- [213] P. Nie, “An open standard for instant messaging : eXtensible Messaging and Presence Protocol ( XMPP ),” in *TKK T-110.5190 Seminar on Internetworking*, 2006.
- [214] B. Almadani, M. N. Bajwa, S. Yang, and A. A. Saif, “Performance Evaluation of DDS-Based Middleware over Wireless Channel for Reconfigurable Manufacturing Systems,” *Int. J. Distrib. Sens. Networks*, vol. 11, no. 7, p. 863123, Jul. 2015.
- [215] “DDS: An Open Standard for Real-Time Applications,” *Real-Time Innovations*. [Online]. Available: <https://www.rti.com/products/dds-standard>. [Accessed: 25-Aug-2020].
- [216] P. Cui, “Comparison of IoT Application Layer Protocols,” Auburn University, 2017.
- [217] D. Glaroudis, A. Iossifides, and P. Chatzimisios, “Survey , Comparison and Research Challenges of IoT Application Protocols for Smart Farming,” *Comput. Networks*, p. 107037, 2019.
- [218] T. Sultana and K. A. Wahid, “Choice of Application Layer Protocols for Next Generation Video Surveillance Using Internet of Video Things,” *IEEE Access*, vol. 7, pp. 41607–41624, 2019.
- [219] Y. Chen and T. Kunz, “Performance evaluation of IoT protocols under a constrained wireless access network,” in *2016 International Conference on Selected Topics in Mobile and Wireless Networking, (MoWNet)*, 2016, pp. 1–7.
- [220] J. Dizdarević, F. Carpio, A. Jukan, and X. Masip-Bruin, “A Survey of Communication Protocols for Internet of Things and Related Challenges of Fog and Cloud Computing Integration,” *ACM Comput. Surv.*, vol. 51, no. 6, pp. 1–29, Feb. 2019.
- [221] S. Raza, H. Shafagh, K. Hewage, R. Hummen, and T. Voigt, “Lithe: Lightweight secure CoAP for the internet of things,” *IEEE Sens. J.*, vol. 13, no. 10, pp. 3711–3720, 2013.
- [222] L. Malina, P. Dzurenda, and J. Hajny, “A Secure Publish / Subscribe Protocol for Internet of Things,” in *ARES '19*, 2019.
- [223] H. Wang, D. Xiong, P. Wang, and Y. Liu, “A Lightweight XMPP Publish/Subscribe Scheme for Resource-Constrained IoT Devices,” *IEEE Access*, vol. 5, pp. 16393–16405, 2017.
- [224] H. Elazhary, “Internet of Things (IoT), mobile cloud, cloudlet, mobile IoT, IoT cloud, fog, mobile edge, and edge emerging computing paradigms: Disambiguation and research directions,” *Journal of Network and Computer Applications*, vol. 128, no. October 2018. Elsevier Ltd, pp. 105–140, 2019.
- [225] P. Hu, S. Dhelim, H. Ning, and T. Qiu, “Author ' s Accepted Manuscript Technologies , Applications and Open Issues Reference ;,” *J. Netw. Comput. Appl.*, 2017.
- [226] S. Taherizadeh, A. C. Jones, I. Taylor, Z. Zhao, and V. Stankovski, “The Journal of Systems and Software Monitoring self-adaptive applications within edge computing frameworks : A state-of-the-art review,” *J. Syst. Softw.*, vol. 136, pp. 19–38, 2018.
- [227] T. Francis, “A Comparison of Cloud Execution Mechanisms : Fog , Edge and Clone Cloud Computing,” no. September, pp. 19–21, 2017.
- [228] S. Yi, Z. Hao, Z. Qin, and Q. Li, “Fog Computing : Platform and Applications,” *2015 Third IEEE Work. Hot Top. Web Syst. Technol.*, pp. 73–78, 2015.
- [229] K. Imagane, K. Kanai, J. Katto, and T. Tsuda, “Computing for Multimedia Data Processing,” *2016 IEEE 5th Glob. Conf. Consum. Electron.*, pp. 1–2, 2016.
- [230] J. Zhang, B. Chen, Y. Zhao, X. Cheng, and F. Hu, “Data Security and Privacy-Preserving in Edge Computing Paradigm : Survey and Open Issues,” *IEEE Access*, vol. 6, no. Idc, pp. 18209–18237, 2018.
- [231] S. Zhang, Y. Liu, S. Li, Z. Tan, X. Zhao, and J. Zhou, “FIMPA: A Fixed Identity Mapping Prediction Algorithm in Edge

- Computing Environment," *IEEE Access*, vol. 8, pp. 17356–17365, 2020.
- [232] S. Nadim-tehrani, "A Taxonomy for Management and Optimization of Multiple Resources in Edge Computing," pp. 1–20.
- [233] T. Penner, A. Johnson, B. Van Slyke, M. Guirguis, and Q. Gu, "Transient Clouds : Assignment and Collaborative Execution of Tasks on Mobile Devices," *2014 IEEE Glob. Commun. Conf.*, pp. 2801–2806, 2014.
- [234] Md. Zia Uddin, "A wearable sensor-based activity prediction system to facilitate edge computing in smart healthcare system," *J. Parallel Distrib. Comput.*, vol. 123, pp. 46–53, 2019.
- [235] X. Zhao *et al.*, "Deep learning based mobile data offloading in mobile edge computing systems," *Futur. Gener. Comput. Syst.*, vol. 99, pp. 346–355, Oct. 2019.
- [236] D. C. Klonoff, "Fog Computing and Edge Computing Architectures for Processing Data From Diabetes Devices Connected to the Medical Internet of Things," *J. Diabetes Sci. Technol.*, vol. 11, no. 4, pp. 647–652, Jul. 2017.
- [237] Q. Duy, M. V Ngo, T. Quang, T. Q. S. Quek, and H. Shin, "Enabling intelligence in fog computing to achieve energy and latency reduction," *Digit. Commun. Networks*, vol. 5, no. 1, pp. 3–9, 2019.
- [238] W. Kim and S. Chung, "User incentive model and its optimization scheme in user-participatory fog computing environment," *Comput. Networks*, vol. 145, pp. 76–88, 2018.
- [239] M. Aazam, S. Zeadally, and K. A. Harras, "Fog Computing Architecture , Evaluation , and Future Research Directions," *IEEE Commun. Mag.*, vol. 56, no. May, pp. 46–52, 2018.
- [240] G. Rathee, R. Sandhu, H. Saini, M. Sivaram, and V. Dhasarathan, "A trust computed framework for IoT devices and fog computing environment," *Wirel. Networks*, vol. 3, 2019.
- [241] G. Yoon, D. Choi, J. Lee, and H. Choi, "Management of IoT Sensor Data Using a Fog Computing Node," *J. Sensors*, pp. 1–9, 2019.
- [242] S. Tuli *et al.*, "HealthFog: An ensemble deep learning based Smart Healthcare System for Automatic Diagnosis of Heart Diseases in integrated IoT and fog computing environments," *Futur. Gener. Comput. Syst.*, vol. 104, pp. 187–200, 2020.
- [243] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The Case for VM-Based Cloudlets in Mobile Computing," *IEEE Pervasive Comput.*, vol. 8, no. 4, pp. 14–23, Oct. 2009.
- [244] Z. Pang, L. Sun, Z. Wang, E. Tian, and S. Yang, "A Survey of Cloudlet Based Mobile Computing," in *2015 International Conference on Cloud Computing and Big Data (CCBD)*, 2015, pp. 268–275.
- [245] J. Ren, D. Zhang, S. He, Y. Zhang, and T. Li, "A Survey on End-Edge-Cloud Orchestrated Network Computing Paradigms," *ACM Comput. Surv.*, vol. 52, no. 6, pp. 1–36, Jan. 2020.
- [246] R. Islambouli and S. Sharafeddine, "Optimized 3D Deployment of UAV-Mounted Cloudlets to Support Latency-Sensitive Services in IoT Networks," *IEEE Access*, vol. 7, pp. 172860–172870, 2019.
- [247] S. Lai *et al.*, "FairEdge: A Fairness-Oriented Task Offloading Scheme for Iot Applications in Mobile Cloudlet Networks," *IEEE Access*, vol. 8, pp. 13516–13526, 2020.
- [248] S. Sengupta and S. S. Bhunia, "Secure Data Management in Cloudlet assisted IoT Enabled e-Health Framework in Smart City," *IEEE Sens. J.*, vol. XX, no. XX, pp. 1–1, 2020.
- [249] A. Kobusinska, C. Leung, C. Hsu, S. Raghavendra, and V. Chang, "Emerging trends , issues and challenges in Internet of Things , Big Data and cloud computing," *Futur. Gener. Comput. Syst.*, vol. 87, pp. 416–419, 2018.
- [250] P. Kumari and P. Kaur, "A survey of fault tolerance in cloud computing," *J. King Saud Univ. - Comput. Inf. Sci.*, 2018.
- [251] M. De Donno, K. Tange, and N. Dragoni, "Foundations and Evolution of Modern Computing Paradigms: Cloud, IoT, Edge, and Fog," *IEEE Access*, vol. 7, pp. 1–1, 2019.
- [252] R. Buyya, "Introduction to the IEEE Transactions on Cloud Computing," *IEEE Trans. Cloud Comput.*, vol. 1, no. 1, pp. 3–21, 2013.
- [253] S. Goyal, "Public vs Private vs Hybrid vs Community - Cloud Computing: A Critical Review," *Int. J. Comput. Netw. Inf. Secur.*, vol. 6, no. 3, pp. 20–29, Feb. 2014.
- [254] T. D. Lee, B. M. Lee, and W. Noh, "Hierarchical Cloud Computing Architecture for Context-Aware IoT Services," *IEEE Trans. Consum. Electron.*, vol. 64, no. 2, pp. 222–230, 2018.
- [255] I. R. Chen, J. Guo, D. C. Wang, J. J. P. Tsai, H. Al-Hamadi, and I. You, "Trust-Based Service Management for Mobile Cloud IoT Systems," *IEEE Trans. Netw. Serv. Manag.*, vol. 16, no. 1, pp. 246–263, 2019.
- [256] D. E. Kouicem, A. Bouabdallah, and H. Lakhlef, "Internet of things security: A top-down survey," *Comput. Networks*, vol. 141, pp. 199–221, 2018.
- [257] X. W. Wu, E. H. Yang, and J. Wang, "Lightweight security protocols for the Internet of Things," *IEEE Int. Symp. Pers. Indoor Mob. Radio Commun. PIMRC*, vol. 2017-Octob, pp. 1–7, 2018.
- [258] F. Hussain, R. Hussain, S. A. Hassan, and E. Hossain, "Machine Learning in IoT Security: Current Solutions and Future Challenges," *IEEE Commun. Surv. Tutorials*, no. c, pp. 1–1, 2020.
- [259] Y. Yang, L. Wu, G. Yin, L. Li, and H. Zhao, "A Survey on Security and Privacy Issues in Internet-of-Things," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1250–1258, 2017.
- [260] F. Meneghelli, M. Calore, D. Zucchetto, M. Polese, and A. Zanella, "IoT: Internet of Threats? A Survey of Practical Security Vulnerabilities in Real IoT Devices," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8182–8201, 2019.
- [261] "300,000 Obeying Devices: Hajime is Conquering the Internet of Things World," 2017. [Online]. Available: [https://www.kaspersky.com/about/press-releases/2017\\_300000-obeying-devices-hajime-is-conquering-the-internet-of-things-world](https://www.kaspersky.com/about/press-releases/2017_300000-obeying-devices-hajime-is-conquering-the-internet-of-things-world). [Accessed: 03-Feb-2020].
- [262] "NIST Releases Draft Security Feature Recommendations for IoT Devices," *NIST*. [Online]. Available: <https://www.nist.gov/news-events/news/2019/08/nist-releases-draft-security-feature-recommendations-iot-devices>. [Accessed: 24-Aug-2020].
- [263] "Good Practices for Security of IoT - Secure Software Development Lifecycle," *The European Union Agency for Cybersecurity*, 2019. [Online]. Available: <https://www.enisa.europa.eu/publications/good-practices-for-security-of-iot-1>. [Accessed: 01-Sep-2020].
- [264] I. Brass, L. Tanczer, M. Carr, M. Elsden, and J. Blackstock, "Standardising a Moving Target: The Development and Evolution of IoT Security Standards," in *Living in the Internet of Things: Cybersecurity of the IoT - 2018*, 2018, vol. 2018, no. CP740, pp. 24 (9 pp.)-24 (9 pp.).
- [265] "oneM2M: Security Solutions (ETSI TS 118 103)," *oneM2M*, 2016. [Online]. Available: [https://www.etsi.org/deliver/etsi\\_ts/118100\\_118199/118103/02.04.01\\_60/ts\\_118103v020401p.pdf](https://www.etsi.org/deliver/etsi_ts/118100_118199/118103/02.04.01_60/ts_118103v020401p.pdf). [Accessed: 01-Sep-2020].
- [266] O. El Mouaaatmid, M. Lahmer, M. Belkasmi, O. El Mouaaatmid, M. Lahmer, and M. Belkasmi, "Internet of Things Security : Layered classification of attacks and possible Countermeasures," *Electron. J. Inf. Technol.*, vol. 9, no. 9, pp. 66–80, 2016.
- [267] X. Liu, M. Zhao, S. Li, F. Zhang, and W. Trappe, "A Security Framework for the Internet of Things in the Future Internet Architecture," *Futur. Internet*, vol. 9, no. 3, p. 27, Jun. 2017.
- [268] Z. Zhang, Y. Wang, and L. Xie, "A Novel Data Integrity Attack Detection Algorithm Based on Improved Grey Relational Analysis," *IEEE Access*, vol. 6, pp. 73423–73433, 2018.
- [269] T. Jourdan, A. Boutet, and C. Frindel, "Toward privacy in IoT mobile devices for activity recognition," in *EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (MobiQuitous '18)*, November 5-7, 2018, pp. 155–165.
- [270] R. Amin, N. Kumar, G. P. Biswas, R. Iqbal, and V. Chang, "A light weight authentication protocol for IoT-enabled devices in distributed Cloud Computing environment," *Futur. Gener. Comput. Syst.*, vol. 78, pp. 1005–1019, 2018.
- [271] S. Echeverria, G. A. Lewis, D. Klinedinst, and L. Seitz,

- [272] "Authentication and Authorization for IoT Devices in Disadvantaged Environments," in *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)*, 2019, pp. 368–373.
- [273] S. Shin and T. Kwon, "A Privacy-Preserving Authentication, Authorization, and Key Agreement Scheme for Wireless Sensor Networks in 5G-Integrated Internet of Things," *IEEE Access*, vol. 8, pp. 67555–67571, 2020.
- [274] H. A. Khattak, M. A. Shah, S. Khan, I. Ali, and M. Imran, "Perception layer security in Internet of Things," *Futur. Gener. Comput. Syst.*, vol. 100, pp. 144–164, Nov. 2019.
- [275] W. Trappe, R. Howard, and R. S. Moore, "Low-energy security: Limits and opportunities in the internet of things," *IEEE Secur. Priv.*, vol. 13, no. 1, pp. 14–21, 2015.
- [276] A. K. C. Bormann, M. Ersue, "Terminology for Constrained-Node Networks," 2014.
- [277] J. King and A. I. Awad, "A distributed security mechanism for Resource-Constrained IoT Devices," *Inform.*, vol. 40, no. 1, pp. 133–143, 2016.
- [278] O. Toshihiko, "Lightweight cryptography applicable to various IoT devices," *NEC Tech. J.*, vol. 12, no. 1, pp. 67–71, 2017.
- [279] M. Katagi and S. Moriai, "Lightweight cryptography for the Internet of Things," *Sony Corp.*, pp. 7–10, 2008.
- [280] S. Singh, P. K. Sharma, S. Y. Moon, and J. H. Park, "Advanced lightweight encryption algorithms for IoT devices: survey, challenges and solutions," *J. Ambient Intell. Humaniz. Comput.*, vol. 0, no. 0, pp. 1–18, 2017.
- [281] B. J. Mohd and T. Hayajneh, "Lightweight block ciphers for IoT: Energy optimization and survivability techniques," *IEEE Access*, vol. 6, pp. 35966–35978, 2018.
- [282] Y. Li, P. Zhang, and R. Huang, "Lightweight Quantum Encryption for Secure Transmission of Power Data in Smart Grid," *IEEE Access*, vol. 7, pp. 36285–36293, 2019.
- [283] M. Usman, I. Ahmed, M. I. Aslam, S. Khan, and U. A. Shah, "SIT: A Lightweight Encryption Algorithm for Secure Internet of Things," *Int. J. Adv. Comput. Sci. Appl.*, vol. 8, no. 1, pp. 1–10, 2017.
- [284] Y. Shi, X. Wang, and H. Fan, "Light-weight white-box encryption scheme with random padding for wearable consumer electronic devices," *IEEE Trans. Consum. Electron.*, vol. 63, no. 1, pp. 44–52, 2017.
- [285] J. Deogirikar and A. Vidhate, "Security attacks in IoT: A survey," in *Proceedings of the International Conference on IoT in Social, Mobile, Analytics and Cloud, I-SMAC 2017*, 2017, pp. 32–37.
- [286] K. Daimi, "Computer and network security essentials," *Comput. Netw. Secur. Essentials*, pp. 1–618, 2017.
- [287] Y. M. Armknecht F, Moriyama D, Sadeghi AR, "Towards a Unified Security Model for Physically Unclonable Functions," in *CT-RSA 2016: Topics in Cryptology - CT-RSA 2016*, 2016, vol. 9610, pp. 271–287.
- [288] R. A. Priyanka Mall, Md Zakirul Alam Bhuiyan, "A Lightweight Secure Communication Protocol for IoT Devices Using Physically Unclonable Function," in *Security, Privacy, and Anonymity in Computation, Communication and Storage. SpACCS 2019. Lecture Notes in Computer Science*, 2018, vol. 11611, pp. 26–35.
- [289] B. S. Muhammad Naveed Aman, Kee Chaing Chau, "Mutual Authentication in IoT Systems Using Physical Unclonable Functions," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1327–1340, 2017.
- [290] Z. A. Khan and P. Herrmann, "Recent advancements in intrusion detection systems for the internet of things," *Secur. Commun. Networks*, vol. 2019, 2019.
- [291] M. F. Elrawy, A. I. Awad, and H. F. A. Hamed, "Intrusion detection systems for IoT-based smart environments: a survey," *J. Cloud Comput.*, vol. 7, no. 1, p. 21, Dec. 2018.
- [292] K. Letou, D. Devi, and Y. Jayanta Singh, "Host-based Intrusion Detection and Prevention System (HIDPS)," *Int. J. Comput. Appl.*, vol. 69, no. 26, pp. 28–33, 2013.
- [293] V. Kumar, A. K. Das, and D. Sinha, "UIDS: a unified intrusion detection system for IoT environment," *Evol. Intell.*, no. 0123456789, 2019.
- [294] F. Y. Leu, K. L. Tsai, Y. T. Hsiao, and C. T. Yang, "An internal intrusion detection and protection system by using data mining and forensic techniques," *IEEE Syst. J.*, vol. 11, no. 2, pp. 427–438, 2017.
- [295] and N. M. Ayyaz-ul-Haq Qureshi, Hadi Larjani, Jawad Ahmad, "A Heuristic Intrusion Detection System for Internet-of-Things (IoT)," in *Intelligent Computing, CompCom 2019, Advances in Intelligent Systems and Computing*, 2019, pp. 86–98.
- [296] P. S. Chatterjee and M. Roy, "Lightweight cloned-node detection algorithm for efficiently handling SSDF attacks and facilitating secure spectrum allocation in CWSNs," *IET Wirel. Sens. Syst.*, vol. 8, no. 3, pp. 121–128, 2018.
- [297] M. Jamshidi, M. Esnaashari, A. M. Darwesh, and M. R. Meybodi, "Using Time-Location Tags and Watchdog Nodes to Defend Against Node Replication Attack in Mobile Wireless Sensor Networks," *Int. J. Wirel. Inf. Networks*, 2019.
- [298] T. Dimitriou, E. A. Alrashed, M. H. Karaata, and A. Hamdan, "Imposter detection for replication attacks in mobile sensor networks," *Comput. Networks*, vol. 108, pp. 210–222, 2016.
- [299] M. Mavani and K. Asawa, "Resilient against spoofing in 6LoWPAN networks by temporary-private IPv6 addresses," *Peer-to-Peer Netw. Appl.*, 2019.
- [300] N. Wang, L. Jiao, P. Wang, M. Dabaghchian, and K. Zeng, "Efficient Identity Spoofing Attack Detection for IoT in Mm-Wave and Massive MIMO 5G Communication," in *2018 IEEE Global Communications Conference (GLOBECOM)*, 2018, no. 1, pp. 1–6.
- [301] M. Hasan and S. Mohan, "Protecting Actuators in Safety-Critical IoT Systems from Control Spoofing Attacks," in *IoT S&P' 19, Proceedings of the 2nd International ACM Workshop on Security and Privacy for the Internet-of-Things*, 2019, pp. 8–14.
- [302] M. Elhadi and M. Shakshuki, "A Detection and Prevention Technique for Man in the Middle Attack in Fog Computing," in *Procedia Computer Science*, 2018, vol. 141, pp. 24–31.
- [303] C. Li, Z. Qin, E. Novak, and Q. Li, "Securing SDN Infrastructure of IoT-Fog Networks From MitM Attacks," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1156–1164, Oct. 2017.
- [304] M. A. Elakrat and J. C. Jung, "Development of field programmable gate array based encryption module to mitigate man-in-the-middle attack for nuclear power plant data communication network," *Nucl. Eng. Technol.*, vol. 50, no. 5, pp. 780–787, 2018.
- [305] S. Iellamo, R. Guiazon, M. Coupechoux, and K. Wong, "Securing IoT Uplink Communications Against Eavesdropping," *2018 3rd Cloudification Internet Things*, no. 612361, pp. 1–8, 2013.
- [306] X. Liu, X. Wei, L. Guo, and Y. Liu, "SecLight: A New and Practical VLC Eavesdropping-Resilient Framework for IoT Devices," *IEEE Access*, vol. 7, pp. 19109–19124, 2019.
- [307] K. Wang, S. Member, L. Yuan, T. Miyazaki, and S. Member, "Jamming and Eavesdropping Defense in Green Cyber – Physical Transportation Systems Using a Stackelberg Game," *IEEE Trans. Ind. Informatics*, vol. 14, no. 9, pp. 4232–4242, 2018.
- [308] H. Chen, C. Meng, Z. Shan, Z. Fu, and B. K. Bhargava, "A novel low-rate denial of service attack detection approach in zigbee wireless sensor network by combining hilbert-huang transformation and trust evaluation," *IEEE Access*, vol. 7, pp. 32853–32866, 2019.
- [309] R. Doshi, N. Aptorpe, and N. Feamster, "Machine Learning DDoS Detection for Consumer Internet of Things Devices," in *Proceedings - 2018 IEEE Symposium on Security and Privacy Workshops, SPW 2018*, 2018, no. MI, pp. 29–35.
- [310] S. Kajwadkar, "A Novel Algorithm for DoS and DDoS attack detection in Internet Of Things," *2018 Conf. Inf. Commun. Technol.*, pp. 1–4, 2018.
- [311] M. Anirudh, S. A. Thileeban, and D. J. Nallathambi, "Use of honeypots for mitigating DoS attacks targeted on IoT networks," in *2017 International Conference on Computer, Communication and Signal Processing (ICCCSP)*, 2017, pp. 1–4.
- [312] Z. Sufang, G. Yimin, L. Shundong, and D. Jiawei, "Deterministic cloned tag detection protocol for anonymous

- [312] P. Gope, R. Amin, S. K. H. Islam, N. Kumar, and V. K. Bhalla, "Lightweight and privacy-preserving RFID authentication scheme for distributed IoT infrastructure with secure localization services for smart city environment," *Futur. Gener. Comput. Syst.*, 2017.
- [313] K. Kaur, S. Member, N. Kumar, and M. Singh, "Lightweight Authentication Protocol for RFID-enabled Systems based on ECC," *2016 IEEE Glob. Commun. Conf.*, no. Id, pp. 1–6, 2016.
- [314] M. Hussein, "Real Time Operating Systems for the Internet of Things ; Vision , Architecture , and Research Directions," pp. 106–111, 2016.
- [315] C. Sabri, K. Lobna, and S. L. Azzouz, "Comparison of IoT constrained devices operating systems : A Survey," 2017.
- [316] E. Baccelli, O. Hahm, M. Gunes, M. Wahlsch, and T. Schmidt, "RIOT OS: Towards an OS for the Internet of Things," in *2013 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2013, pp. 79–80.
- [317] N. He & H. W. Huang, "Use of FreeRTOS in Teaching Real-time Embedded Systems Design Course," in *121st ASEE Annual Conference & Exposition*, 2014.
- [318] J. Pereira *et al.*, "Co-Designed FreeRTOS Deployed on FPGA," in *2014 Brazilian Symposium on Computing Systems Engineering*, 2014, pp. 121–125.
- [319] L. Ciabattoni, A. Freddi, S. Longhi, A. Monteriù, L. Pepa, and M. Prist, "An Open and Modular Hardware Node for Wireless Sensor and Body Area Networks," *J. Sensors*, vol. 2016, pp. 1–16, 2016.
- [320] P. Budhwar, "TinyOS : An Operating System for Wireless Sensor Networks," *IJCST*, vol. 8491, pp. 141–145, 2015.
- [321] P. Levis *et al.*, "TinyOS: An Operating System for Sensor Networks," in *Ambient Intelligence*, Berlin/Heidelberg: Springer-Verlag, 2005, pp. 115–148.
- [322] D. Gay, P. Levis, R. von Behren, M. Welsh, E. Brewer, and D. Culler, "The nesC language," in *Proceedings of the ACM SIGPLAN 2003 conference on Programming language design and implementation - PLDI '03*, 2003, p. 1.
- [323] B. C. Söllner, "Comparison of Operating Systems TinyOS and Contiki," in *Chair for Network Architectures and Services, Department of Computer Science, Technische Universität München, 2012*, 2012, no. August, pp. 7–13.
- [324] M. H. Outgut, A. Al-Sakran, F. Almasalha, and H. S. Hassanein, "Comprehensive survey of the IoT opensource OSs," *IET Wirel. Sens. Syst.*, vol. 8, no. 6, pp. 323–339, 2018.
- [325] O. Hahm, E. Baccelli, H. Petersen, and N. Tsiftes, "Operating Systems for Low-End Devices in the Internet of Things: A Survey," *IEEE Internet Things J.*, vol. 3, no. 5, pp. 720–734, 2016.
- [326] S. Jinfan, "Huawei LiteOS: A Heavyweight in IoT Connectivity," 2018.
- [327] "Wind River Redefines Embedded Software Development with New VxWorks Release," *Wind*.
- [328] "VxWorks 6.0 Hardware Considerations Guide," *Wind River Systems, Inc.*, 2004..
- [329] A. Avanzini, P. Valente, D. Faggioli, and P. Gai, "Integrating Linux and the real-time ERIKA OS through the Xen hypervisor," in *10th IEEE International Symposium on Industrial Embedded Systems (SIES)*, 2015, pp. 1–7.
- [330] "Nut/OS Memory Considerations," *egnite GmbH*, 2007. [Online]. Available: <http://www.ethernut.de/pdf/enmem21e.pdf>. [Accessed: 17-Aug-2020].
- [331] Y. Li, M. Potkonjak, and W. Wolf, "Real-time operating systems for embedded computing," *Proc. - IEEE Int. Conf. Comput. Des. VLSI Comput. Process.*, no. December 1998, pp. 388–392, 1997.
- [332] "Tizen IoT," *Linux Foundation*, 2020. [Online]. Available: <https://docs.tizen.org/iot/get-started/overview/>. [Accessed: 18-Aug-2020].
- [333] A. Grush, "Project Brillo is Google's new Internet of Things OS," *Android Authority*, 2015. [Online]. Available: <https://www.androidauthority.com/project-brillo-google-io-612159/>. [Accessed: 17-Aug-2020].
- [334] "Ubuntu Core Documentation," *Canonical Ltd.* [Online]. Available: <https://core.docs.ubuntu.com/en/>. [Accessed: 18-Aug-2020].
- [335] N. Klingensmith and S. Banerjee, "HERMES: A Real Time Hypervisor for Mobile and IoT Systems," in *HotMobile 2018 - Proceedings of the 19th International Workshop on Mobile Computing Systems and Applications*, 2018, vol. 2018-Febru, pp. 101–106.
- [336] R. Raymundo Belleza and E. Pignaton de Freitas, "Performance study of real-time operating systems for internet of things devices," *IET Softw.*, vol. 12, no. 3, pp. 176–182, Jun. 2018.

**NARASIMHA SWAMY S**

He received the bachelor's and Master of Technology from Visvesvaraya Technological University (VTU) in 2010 and 2013, respectively. He is currently working as Assistant Professor with the Rashtreeya Vidyalaya College of Engineering (R. V College of Engineering) and pursuing the Ph. D degree in IoT area with the Academy of Scientific and Innovative Research (AcSIR), Ghaziabad at CSIR-Central Electronics Engineering Research Institute, Pilani, Rajasthan. His research interest includes IoT protocols, Edge Intelligence, IoT Security, Reconfigurable Computing, and NoSQL Databases.

**KOTA SOLOMON RAJU**

He is working as a Sr. Principal Scientist and Group Leader, Societal Electronics Group, CSIR-Central Electronics Engineering Research Institute, Pilani and Professor, Academy of Scientific and Innovative Research, Ghaziabad. He received the B.E. from Andhra University in 1997, M.E. from BITS, Pilani in 2003, and Ph. D. from IIT Roorkee in 2008. He has twenty-two years of experience in R&D and his research work mainly focused on Advanced Electronic Systems Engineering includes Reconfigurable Computing, CPS, Edge Intelligence, Industry 4.0, 5G & Beyond Communication Technologies, IoT Security, Development of Hardware for AI and SHM & Securing Critical Infrastructure.

