sustainability

Article

Underpinning Quality Assurance: Identifying Core Testing Strategies for Multiple Layers of Internet-of-Things-Based Applications

Amer Aljaedi [1,*], Saba Siddique [2], Muhammad Islam Satti [3], Adel R. Alharbi [1], Mohammed Alotaibi [4] and Muhammad Usman [5]

[1] College of Computing and Information Technology, University of Tabuk, Tabuk 71491, Saudi Arabia; aalharbi@ut.edu.sa
[2] Department of Computer Software Engineering, National University of Sciences and Technology, Islamabad 44000, Pakistan; saba.siddique@mcs.nust.edu.pk
[3] Department of Computing (TMUC), Millennium Institute of Technology & Entrepreneurship (MiTE), Karachi 75190, Pakistan; muhammad.islam@tmuc.edu.pk
[4] Department of Management Information Systems, College Business Administration, University of Tabuk, Tabuk 71491, Saudi Arabia; msalotaibi@ut.edu.sa
[5] Department of Computer Science, Edge Hill University, Lancashire L39 4QP, UK; usmanm@edgehill.ac.uk
* Correspondence: aaljaedi@ut.edu.sa

Abstract: The Internet of Things (IoT) constitutes a digitally integrated network of intelligent devices equipped with sensors, software, and communication capabilities, facilitating data exchange among a multitude of digital systems via the Internet. Despite its pivotal role in the software development life-cycle (SDLC) for ensuring software quality in terms of both functional and non-functional aspects, testing within this intricate software–hardware ecosystem has been somewhat overlooked. To address this, various testing techniques are applied for real-time minimization of failure rates in IoT applications. However, the execution of a comprehensive test suite for specific IoT software remains a complex undertaking. This paper proposes a holistic framework aimed at aiding quality assurance engineers in delineating essential testing methods across different testing levels within the IoT. This delineation is crucial for effective quality assurance, ultimately reducing failure rates in real-time scenarios. Furthermore, the paper offers a mapping of these identified tests to each layer within the layered framework of the IoT. This comprehensive approach seeks to enhance the reliability and performance of IoT-based applications.

Keywords: IoT failure causes; layered architecture of IoT; quality assurance; testing framework

check for updates

Citation: Aljaedi, A.; Siddique, S.; Satti, M.I.; Alharbi, A.R.; Alotaibi, M.; Usman, M. Underpinning Quality Assurance: Identifying Core Testing Strategies for Multiple Layers of Internet-of-Things-Based Applications. *Sustainability* **2023**, *15*, 15683. https://doi.org/10.3390/su152215683

Academic Editor: Yang (Jack) Lu

Received: 2 September 2023
Revised: 5 October 2023
Accepted: 10 October 2023
Published: 7 November 2023

1. Introduction

The IoT is not simply a concept but an architectural paradigm that provides the medium for exchanging captured data and the means of integrating physical world and computer systems over a defined network. Independent technologies construct the IoT's fundamental components. The IoT's applications can be found in an array of devices, industries, and settings. The components of the IoT are based on object, communication, and computing modules. The main functionality of object modules is to provide a response to instructions and retrieve data. Communication means the network to be used, and it comprises protocols and technologies that allow the exchange of information or data between physical objects. It might be a Local Area Network (LAN), Wide Area Network (WAN), Metropolitan Area Network (MAN), or cellular network. Computing includes collecting, processing, storing, and manipulating the underlying data. It reflects the overall behavior of the system. The accurate form of computing analyzes user behavior, efficiently makes the right decisions based on user nature, and makes deductions. The IoT is an evolving trend, and each evolving trend needs some sort of affirmation regarding its quality. This outlines the great need for quality assurance in IoT to optimize the ongoing processes and meet user expectations. As each IoT application is a different product and has a different usage, they cannot be categorized as one. Even in the same domain, many different types of IoT applications with several layers exist, and as they evolve very quickly, the software quality assurance process must also be updated. Therefore, the accurate measurement of quality is essential, considering the presence of objects of heterogeneous nature that are bound with one another to build an IoT system. However, performing quality assurance (QA) testing in IoT is precarious as it is a huge network of physical devices and involves testing both the hardware and software, recording the test results, and then sending them back in real time, which is not an easy task to accomplish. There are numerous failure cases of the IoT in real time. One such example is Petnet, an automated pet feeder [1]. This device encountered system failure and was unable to establish reliable communication with connected devices. This system depended on third-party servers which they rented out from Google. The problem started when the servers were not responsive for 10 h and the system had no backup plan. The users affected by this situation lost the ability to set feeding schedules for their pets and were also unable to remotely get hold of the device and command it to feed their pet. This endangered pets' lives as they might have missed meals and starved if the owner was unable to reach home and depended on this device for feeding services. Although Petnet was luckily successful in resolving the issue on time, this raises the question as to what measures are important in making an IoT product a market-winning product that also satisfies user expectations. In this paper, we conduct a literature review to identify the IoT's current trends concerning the quality assurance process. The issues related to QA and their existing solutions are provided in Table 1. The problem area is identified after critically analyzing the reviewed literature and addressed by our proposed framework, in which we have identified basic yet important tests for IoT-based applications to ensure quality. Although these approaches are basic, they still lay the foundation of quality assurance in the IoT. This area needs special attention in IoT development as the idea behind the IoT is smart living. This goal is not achievable without user satisfaction. For user satisfaction, quality is the utmost requirement that needs to be fulfilled. This paper also provides a mapping architecture in which we have mapped all the important tests across the six-layered architecture of the IoT. The basic aim behind this architecture is to enable QA engineers to identify the tests that must be conducted at different levels of the IoT. This paper is structured as follows: We present a literature review in Section 2. Section 3 provides our proposed framework for IoT testing and its implications, whereas Section 4 includes our mapping architecture. Section 5 includes the evaluation of our research. Section 6 provides the conclusions of this study and deliberates some future work based on prior research.

Table 1. QA Evaluation and Analysis Parameters.

Reference	Research Gaps	Proposed Solutions
A. E. Al-Fagih et al. [2]	Challenges related to pricing, resource management, and inter-operability in wireless sensors	Priced PS framework for architectures of IoT for applications related to services in cities to make them smart and the use of utility function of pricing for acquisition of data
J. Kiljander et al. [3]	The devices heterogeneity, for representing their functionality in form of a platform for virtual computing	Architectures for interoperability of semantic level architecture for pervasive IoTs and its computing
J. Zhou et al. [4]	Large amount of data which is sparse, dynamic, hetero geneous, and multi-source in IoT	Use of data fusion for manipulation and management of such kind of data for improvement of efficiency of data and system and for providing advanced level of intelligence
Leal et al. [5]	Sensing which is trustworthy and safe for general public in IoTs which are cloud-centric	Sensing-as-a-Service (S2aaS) enhances safety from public by using sensing services with help of crowd management which is provided by various smartphones having different sensors

Table 1. *Cont.*

Reference	Research Gaps	Proposed Solutions
F. Li et al. [6]	Integration of secured network for integration of wireless sensors of network into IoT	Heterogeneous signcryption scheme which are offline and online as well, for securing of the communication between an internet boot and sensor node
X. Mao et al. [7]	Issue of systems which are cyber-physical and networks which are wireless sensor in IoT	Not Defined
E. S. Reetz et al. [8]	Testing of services which are based on IoT before their deployment into the world	Emulation of resources of IoT interface from the architectural, implementation and semantic perspective
D. Kuemper et al. [9]	Derivation of test for services of IoT which are semantic based	Methodology for enriching of service related to descriptions for derivation of testing which is semi-automated and is required for the adaptions of IoT
P. Gimenez et al. [10]	Faster tests with reduced cost of operation and low risk	using simulator of high quality sensor, SWE simulation and web standards for sensors
J. Fernandes et al. [11]	So-designing issues in IOT	Platform of loT lab for the design framework of loT reference architecture model for creation of an initial design including test-bed components, crowd-sourcing, ability to do federation with other test-beds and virtualization
V. A. Desnitsky et al. [12]	Monitor security components for anomaly detection in components and data in IoT	Use of expert knowledge and elicitation approach and for detecting data anomalies and giving them as an input for automated systems for monitoring of IoT components of security
C. Chuang et al. [13]	Application on quality assurance of composite digital services on Intelligent Transportation System	Framework for integration of end-to-end testing for quality assurance which works on DSRA (Digital Service Reference Architecture) supported by forum of TM
M. Masirap et al. [14]	In-adaptability of protocol like TCP	Transport protocols based on UDP which are UDT, PA UDP and RUBDP for use in application of IoT
S. Sankaran et al. [15]	Increased sensitivity of securing IoTs related to data of user and consumption of high power which is the nature of IoTs	Using cryptography which is identity based and development of security framework which is light in weight for IoTs
A. R. Chandan et al. [16]	Maintaining authenticity, confidentiality and integrity for securing the network of the IoT network	Framework for testing of IoT
D. Kim et al. [17]	Verification and execution of applications of IOT	TITAN is designed in such a way for allowing developers to verify and execute IoT applications preventing from being constrained by the environment, in a development environment
A. Kaiser et al. [18]	Growing of probability and complexity of vulnerabilities and malfunctions in IOT	IoT-Test ware Eclipse for ensuring conformance and robustness of protocol and secured implementations
M. Abdallah et al. [19]	Model for quality measurement, making the measurement process of quality less applicable, less accurate, and more challengeable	Model for the quality of IoT which consists the characteristics of IoT systems, by introduction of quality factors for measuring them
S. Popereshnyak et al. [20]	Difficulties related to the phase of testing of applications and systems of IoT	Features related to testing, based on network of modeling of IoT Application
Kim et al. [21]	Increased complexity and cost of testing because of the large number of variables heterogeneity and scalability of conventional testing of IoT devices	Testing of IoT as a Service called IoT-TaaS, which is a service oriented approach for automation of testing of IoT
K. Papachristou et al. [22]	Routing and runtime verification of the policies for security to enhance quality of networks in IoT	Framework for number of information security policies' verification at run-time, of the network and dynamic routing paths flow

2. Literature Review

Numerous sensor-assisted devices associated together are present in the current era. The topologies made using such devices and other pervasive items give us the Internet of Things (IoT); this is a whole new paradigm that allows all the already existing mechanisms to be equally detectable, controllable, and linked. Multi-attribute quality score computation is a method for evaluating the quality of Internet of Things (IoT) applications based on multiple attributes or criteria. Rohini Temkar et al. [23] proposed an approach that takes into account various factors such as reliability, usability, security, performance, and scalability, among others, and assigns a score to each attribute. The overall quality score is then computed by combining these individual attribute scores based on their relative importance. Multi-attribute quality score computation can be used to compare different IoT applications and select the one that best meets the user's requirements. It can also be used to monitor the quality of an IoT application over time and identify areas for improvement. Various methods such as fuzzy logic, decision trees, and neural networks can be used to implement multi-attribute quality score computation. These methods can handle the uncertainty and imprecision associated with evaluating multiple attributes and can provide accurate and reliable quality scores. Overall, multi-attribute quality score computation is a useful approach for evaluating and comparing the quality of IoT applications based on multiple criteria. It can help ensure that IoT applications meet the user's expectations and provide a satisfactory user experience.

Yair Rivera Julio et al. [24] offered a comprehensive framework for managing software quality in IoT applications. They emphasized its significance and addressed associated challenges in these intricate systems. The framework comprises five stages, covering aspects from requirements engineering to maintenance. Each stage is detailed, including recommended activities, techniques, and tools. The benefits include enhanced software quality, greater efficiency, and cost savings. The authors stressed the importance of a systematic approach to quality management, integrating it into the entire development process. They also acknowledged challenges, including specialized skills, managing quality across layers, and adapting to the dynamic nature of IoT applications.

Noha Medhat et al. [25] emphasized the critical role of testing for ensuring the quality of complex and diverse IoT systems. They discussed a range of testing techniques, including functional, non-functional, integration, and regression testing. Each technique was thoroughly described, along with its advantages and limitations in the context of IoT systems. The paper underscores the significance of testing at every stage of the software development life-cycle, from requirements engineering to maintenance, stressing the need for continuous testing to meet specified quality standards.

A. Sharma and A. K. Sarje [26] stressed the pivotal role of testing for quality, reliability, and security in IoT systems, given their complexity. They detailed various testing techniques, including functional, performance, security, and interoperability testing. Their paper introduces emerging techniques like crowdsourced, automated, and model-based testing, highlighting their potential benefits. Practical implementation guidance is provided. The paper also addresses challenges, including the absence of standardized testing tools, the complexity in testing across layers, and the need for specialized expertise.

R. Kumari and M. K. Soni's study [27], published in 2020 in the *International Journal of Advanced Research in Computer Science and Software Engineering*, comprehensively explored quality assurance techniques for IoT systems. It underscored their vital role in diverse applications due to the complexity and heterogeneity of IoT systems. The paper covers techniques including requirements engineering, testing, verification and validation, fault tolerance, and maintenance, offering detailed descriptions along with their respective advantages and limitations. Emerging techniques like machine-learning-based testing, blockchain-based verification, and edge-computing-based fault tolerance are also discussed, with emphases on their potential benefits and practical implementation guidance in IoT systems.

Shanzhi Chen et al. [28] indicated that an IoT system must have three characteristics: Comprehensive Perception, Reliable Transmission, and Intelligent Processing. Comprehensive Perception means obtaining authentic information anywhere and anytime it is needed. Reliable Transmission means reliable data availability through all radio, Internet, and telecommunication channels. Intelligent Processing such as cloud computing supports the IoT, and it means a huge amount of data to be processed. The IoT must also incorporate other characteristics such as connectivity, enormous scale, sensing, dynamic changes/nature, heterogeneity, and security [19]. For the IoT to be successful, it is mandatory to keep its business models and applications clear.

It is insufficient to keep just one plan in mind. To reduce the risk of failure, business aspects should be considered in the early stages of IoT development to minimize the risk of failure. To guarantee credibility in the IoT framework, the S2aaS scheme with explicit Trustworthy Sensing for Crowd Management (TSCM) for front-end contact with the IoT is used [29]. TSCM gathers distinguishing data grounded on a cloud model and a technique that picks out mobile devices for precise sensing responsibilities, regulating the outgoings to users of the mobile devices that offer data. An enactment assessment of TSCM demonstrated that the power of malevolent users in the crowd-sourced data can be decreased by 75 percent, while the dependability of a malevolent user amounts to under 40 percent. The reduced cost, lessened configuration load, and constricted pairing with the power-driven product make Communicating Power Supplies an exceptional application of the IoT.

In the IoT, when numerous devices are connected, a power consumption problem arises, as it is unknown which device requires how much energy and when it needs it. Most of the devices require energy conversion from an AC source to DC power. CPSs convert AC to DC power supply [15]. The permeating nature of sensors, in addition to the sensitivity of user data, makes it a need for the IoT to ensure immense security. In addition, power constraints are an important limitation of the IoT. Therefore, there is a need to secure the IoT using some lightweight solution. A security framework using identity-based cryptography is a solution to this problem [6].

IoT-based networks usually consist of minute sensor nodes with base stations. Sensor nodes usually have limited power, while a base station is powerful enough that it provides an interface between the user and node. In a TCP/IP suite, the base station acts as a router, and there are security challenges such as secure channel setup and end-to-end authentication. A heterogeneous encryption scheme is used for secure communication between the base station and sensor node [7].

The current research challenges for the IoT are channel assessment, system schemes, resource administration, the assimilation of several schemes, application advancement, network protocol plans, and even changeovers from legacy systems [2]. For maximum utility gain, quality control in the IoT is an important parameter. In Priced Public Sensing (PPS) [3] in the IoT, data delivery schemes are divided into delay-tolerant and delay-sensitive schemes. In the IoT using cheap sensors, quality is ensured using efficient algorithms. Interoperability is an important part to be considered while evaluating IoT systems. Considering the layered approach, connectivity-level interoperability means that the connected devices can transfer data to each other without knowing the actual meaning. Semantic-level interoperability enables devices to understand the meaning of transferred information.

The functionality of these systems and their interactions are tested at design time [9]. Immense knowledge of data types, system dependencies, and the behavior of services is required to derive tests for the IoT. The data regarding test case generation are derived from the stored knowledge of services in a knowledge database. The IoT is a huge paradigm involving billions of devices, which means a huge amount of gigantic, dynamic, and heterogeneous data. For data management and manipulation, data fusion [4,8] is a tool that is important for improved efficiency. The goal of this method is to ensure better quality in obtaining information. Eike Steffen Reetz et al. [11] proposed a semiautomated approach in which test code is injected to enable efficient prototyping, along with the integration of tests for IoT services. If the logical interaction of services and IoT systems is semantically defined, then the knowledge gathered via services might help in the generic resource emulation interface. In this way, a service gives an abstract image of IoT interaction and enables efficient and scalable emulation of IoT systems.

A new extension of IoT testbed groundwork is crowdsourcing, which manages communities online while keeping an eye on why the crowd is showing interest in this domain and what they want. For this purpose, a third party needs to be involved to obtain immediate feedback and record it accordingly. This helps the IoT lab to select important use cases, and then the crowd can again give feedback regarding the selection. This is how testing is made more efficient [5]. For the IoT, testbeds are used for simulation, which is vital but also has challenges, leaving room for improvement. Brazil wants new regulations to be followed in transportation regarding RFID and the IoT. The Brazilian transportation department intends to broaden its research area for research into the IoT [30]. The object name services (ONS) projects aimed at discovering a suitable way of communicating automatically [12]. All the work it does is in addition to radio-frequency identification (RFID) and the IoT. But both of them have crucial security concerns. To assure internal communication security, a repetitive and hard testing scenario is produced so that one can obtain a clear picture regarding the behavior of the system under load.

Mohammad Abdallah et al. [28] proposed a new quality model after studying different existing models for quality and comparing them regarding the factors that play a role in quality measurement that can be used as a basis for finding other factors in the future for improving the quality of IoT systems. As the number of IoT systems is increasing, the complexity of IoT systems is also increasing, and as the complexity is increasing, the need for improved security is also increasing. The solution to this problem lies in detecting anomalies in the data and then assessing where these data are being used as inputs in IoT-based systems [31]. In the IoT, there are certain design requirements that must be fulfilled. These are connectivity, security, sensor, and touch [14].

Security and heterogeneous test integration are features in the IoT that require innovative solutions. IoT-based systems need to maintain the per unit cost of production, so while handling the above-mentioned issues, managing cost-effectiveness is challenging. It is a requirement of the IoT that the wireless connection should be in real time, and for real-time connections, Transport Control Protocol (TCP) is not enough as it has a large header size, slow start, and Additive Increase Multiplicative Decrease (AIMD) congestion control algorithm. Madzirin Masirap et al. [20] tested the UDP using a testbed with two systems connected by an ad hoc network. The evaluation results showed that the UDP is far better than the TCP in terms of speed and resource utilization.

Svitlana Popereshnyak et al. [10] provided features that can be used to test IoT applications and devices, along with the main differences in testing techniques of classical systems and IoT systems. Basically, two types of IoT testing were discussed: user convenience testing and network connection testing. For approving the proposed specifications and selecting an application protocol, an experiment and medical applications were also conducted in order to test a portion of a communication network. The service-oriented testing model architecture was used, following the generic life cycle of software development, consisting of four main phases.

The major part of this model is the automation of testing in every phase of the model, which includes testing the deployed parts of systems, benchmark testing, and integration testing. CoAP is a good protocol for the communication of IoT projects as it is lightweight and fast. It ensures the stable operation of the system, providing reliability and the possibility that the system can be improved even after its release. Testing methods in the IoT should evaluate the developed system and check the non-functional requirements of the client. In the IoT, wireless sensor networks are quite important. A cheap way to add sensors in networks is via virtual sensors.

Simulation tools are sufficient to evaluate a system when checking riskless deployment, and they are efficient and economical. The Sensor Web Enablement (SWE) simulator is a software module that enables the emulation of various sensors, which helps in producing numerous use cases and situations that might be external to those in real time [32]. In an urban IoT system, if proper monitoring of the structural health of a building is required, then numerous sensors are required to be embedded in the building, as well as in its surroundings, to monitor pressure, pollution, and other factors [13]. In this way, it becomes easier and more efficient to maintain the database to ensure the quality of buildings, as well as the system, as compared to humans performing these tasks. The intelligent transportation system (ITS) is a recent advancement in the IoT [16].

To assure quality in the ITS, Information and Communication Technology (ICT) strategies and facilities are utilized. Each device or service we use from ICT has its own Operation Support System (OSS), which limits information exchange from device to device. A framework composed of Digital Service Reference Architecture (DSRA) and TM Forum is feasible to ensure end-to-end quality in ITS. Abhishek R. Chandan et al. [17] proposed a methodology for security testing. Information related to IOT devices and networks was gathered; a better understanding of a device, which assists in threat profiling and testing of the device, depends upon the implemented device and the network. This methodology can be followed during penetration testing to ensure quality in the IoT. Integrity and heterogeneity can be achieved by using proper techniques of encryption, firewalls, and security protocols, along with a lightweight key management system. Proper data management is required to maintain confidentiality and availability. Scalability can be achieved through the enforcement of policies.

Authenticity is very challenging to achieve as per the nature of the IoT. TITAN [18] is a tool that provides a virtual environment to developers where they can efficiently run and test IoT applications in the development process without them being influenced by the physical environment and the behaviors of users. This minimizes the time and effort required for repetitive testing during development. The availability of open testing equipment and how it can have a positive impact on IoT applications is an issue [22]. ETSI (European Telecommunications Standards Institute), oneM2M, and the OPC-Foundation have already been working on this, and they have provided open standards and open-source testing equipment.

Alexander Kaiser et al. [32] also discussed the results of an experiment performed using brokers of MQTT and suggested that protocol implementation must not be considered a trivial task; open-source testing tools must be available to benefit open-source projects. The two types of protocol testing discussed in the paper are protocol conformance and protocol security, both of which have their own impacts on the final product and its vulnerability. Konstantinos Papachristou et al. [21] proposed two frameworks for the verification of many security policies, which relate to information on the dynamic flow of routing paths and networks. The underlying set of concepts is to let the operator control the overall network and define various policies, which are the basis of network demands, as well as use cases for achieving a faster and more secure network.

The input given to the optimization algorithm (multi-objective) is the routing policies and statistics taken in real time. This information then helps in the calculation of policies for routing to estimate the quality of routing decisions. The flow rules created by SDN are compared with the optimal set of flow rules, giving the actual results for verification in the form of a deviation metric. HIUN KIM et al. [33] proposed an IoT testing framework called Service-IoT-TaaS. It works on the basis of the "plug and test" concept, using a service-based approach in order to provide an IoT testing framework that is automated and provides solutions to the traditional software testing issues of costs, coordination, and scalability, in order to use standard-based processes of the development of IoT devices and to explore their implementation and design. IoT-TaaS consists of remotely distributed automated scalable conformance testing, testing to validate semantics, and interoperability testing.

Minhaj Ahmad Khan et al. [34] showed how basic features of blockchain can be used as a key in solving security problems related to the IoT, but still, there needs to be some mechanism more effective than this to avoid attacks. However, an attacker can host the blockchain and, hence, the hashing power of the miner can be a risk, which relates to the consensus mechanism. Private keys that have a little randomness can also be used to attack the accounts of the blockchain.

3. Framework for IoT Testing

For quality assurance, different types of testing techniques are conducted. However, conducting all the tests for a particular software product, especially in IoT applications, is difficult. Therefore, there is a need to define all the necessary tests which must be conducted for quality assurance in the IoT and to ultimately reduce the failure rate in real time. Our proposed framework is illustrated in Figure 1. This framework provides a comprehensive outline regarding all the necessary tests to be conducted during the quality assurance process in the IoT.

3.1. Device-Level Testing

This test involves testing the logging function of devices, as this is the entrance to a device in the IoT. Device-level testing includes the testing of sensors, operating systems to be used, and system hardware and circuits.

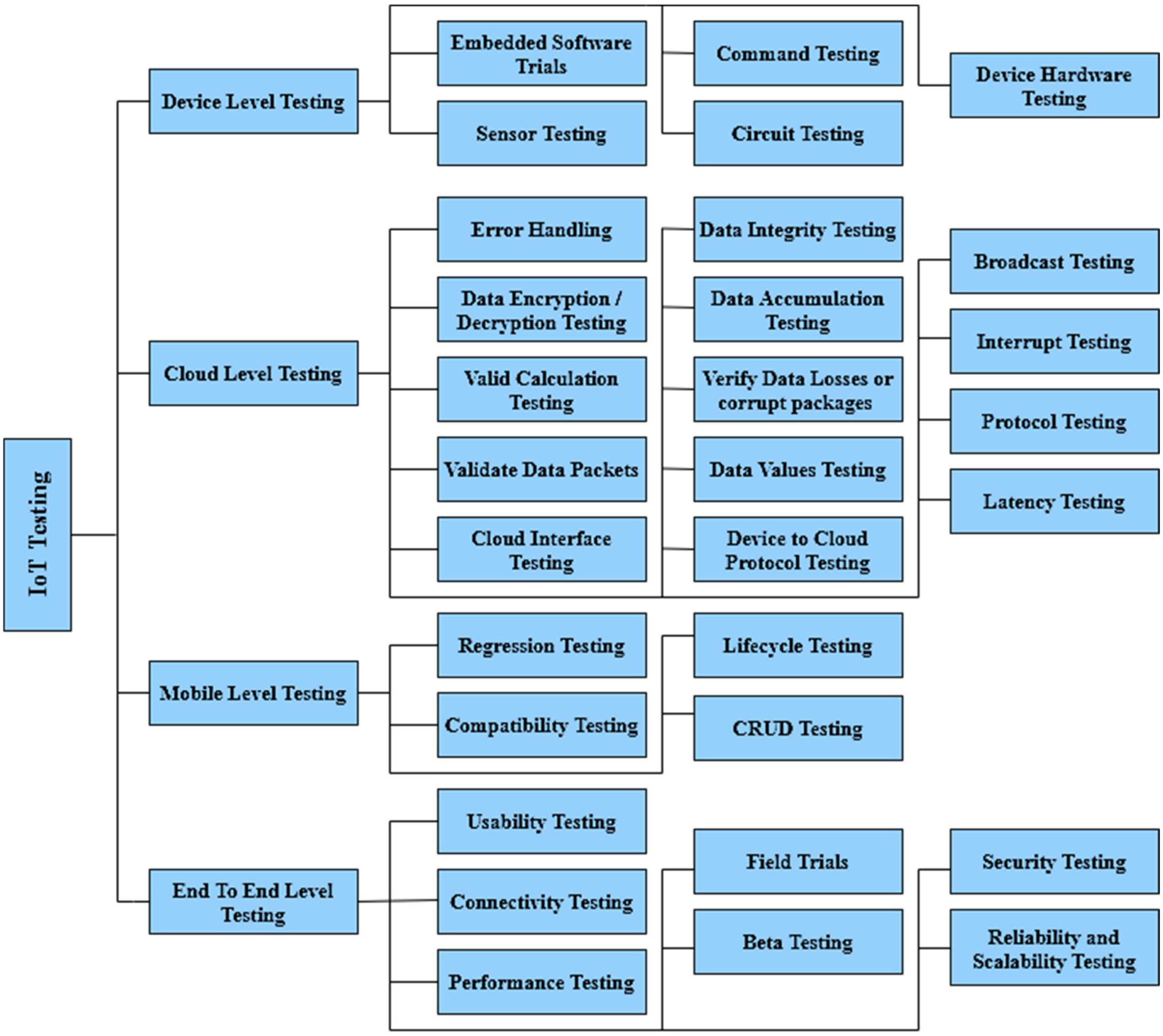

Figure 1. IoT testing framework.

3.2. Cloud-Level Testing

This includes issues regarding the functionality, integration, and cloud API. Another major concern is to ensure high security for user data and all devices connected to the cloud (security test). To ensure security, the IoT must allow access to authentic users, use encryption to protect data from intrusion, and safely store data at a reliable location. As the data consistently move in and out of the cloud in the IoT, to ensure that they are subject to consistent policies, governance tests are conducted; data privacy tests are very much related to security and data governance. Other tests which must be performed here are related to data, packets, different protocols, interruptions, and latency.

3.3. Mobile-Level Testing

Consumers are dependent on mobile devices to interact with the connected devices. Therefore, in mobile testing, it is mandatory to test the mobile app design, its interface, its back-end, and its way of communicating with all connections and the cloud. Compatibility testing identifies what version of the SDK is compatible with the system. When a new version has been launched, the system updates itself to match the major versions. CRUD testing comprises Create, Read, Update, and Delete tests, which are conducted because it is the user who performs all these functions in the account creation and data sharing process [35,36]. Life-cycle testing means actually testing the mobile device in action. Mobile-level testing also includes regression testing. The mobile device is assessed from start to end in its functionality and usually fails at this stage.

3.4. End-to-End Testing

This testing is the most extensive as it checks all the previously mentioned components of the IoT, namely, objects, communication, and computing. This is basically testing from start to finish, involving every component, as shown in Figure 2. In end-to-end testing, not only are the components checked—it is also crucial to check how they interact [37]. Field trials consist of a group of actual users checking the system and giving feedback.

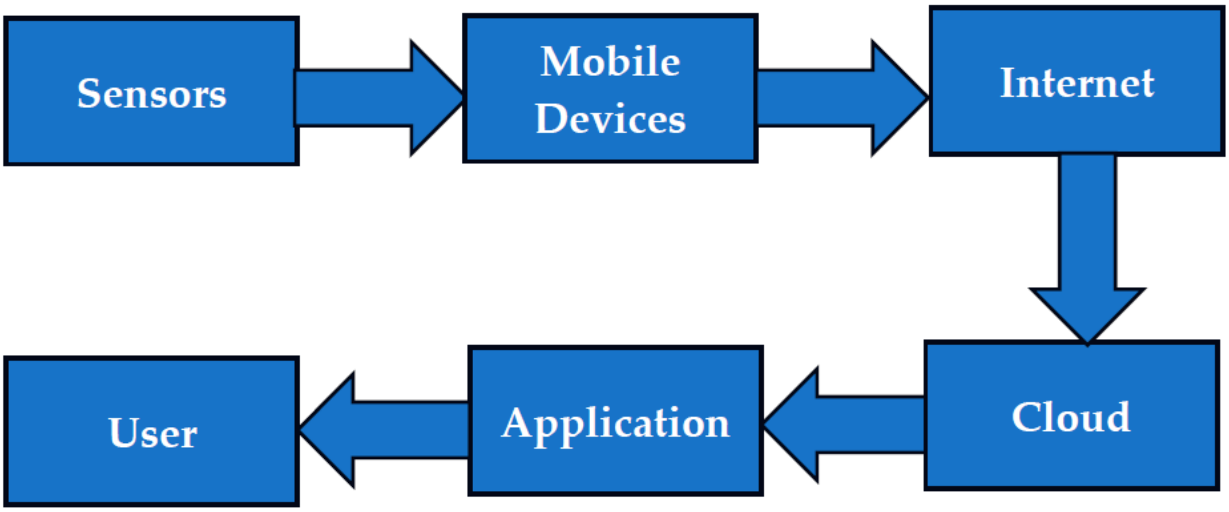

Figure 2. End-to-end test workflow.

Changes are made accordingly, and the system is again released for another field trial. This testing is also concerned with the usability of a device and its necessity in the IoT. It is critical to know how to scale all the available users in the cloud at a time (scalability test), what the effects of higher loads will be on the performance of the application (performance test), and how reliability will be ensured in cases of stress (reliability test). All of these tests must be performed at this level. End-to-end testing also ensures the system's security and connectivity.

4. Mapping of Important Testing Domains at each Layer of the IoT's Layered Architecture

Layered architecture best describes the working of an IoT device as it distributes and distinguishes the responsibilities of each component, either hardware or software, in an IoT device. For instance, the interface through which the user interacts with an IoT device is distinguished and separated from the main business logic of the device on which it works, and the connection media, such as the Internet, comprise a distinguished component. Likewise, storage media are also a separate component. The database layer has nothing to do with the interface through which the user interacts [36]. It is the business layer, which works as middle-ware and makes the data presentable on an interface. Hence, every logical layer is separated, and the security of every layer is preserved. Describing an IoT architecture in terms of a layered architecture best helps in the distribution of responsibilities among the development team. The mapping of different testing domains for the IoT at each layer will help to identify each important test that must be performed at each layer. Figure 3 shows our mapping architecture for the IoT [37].

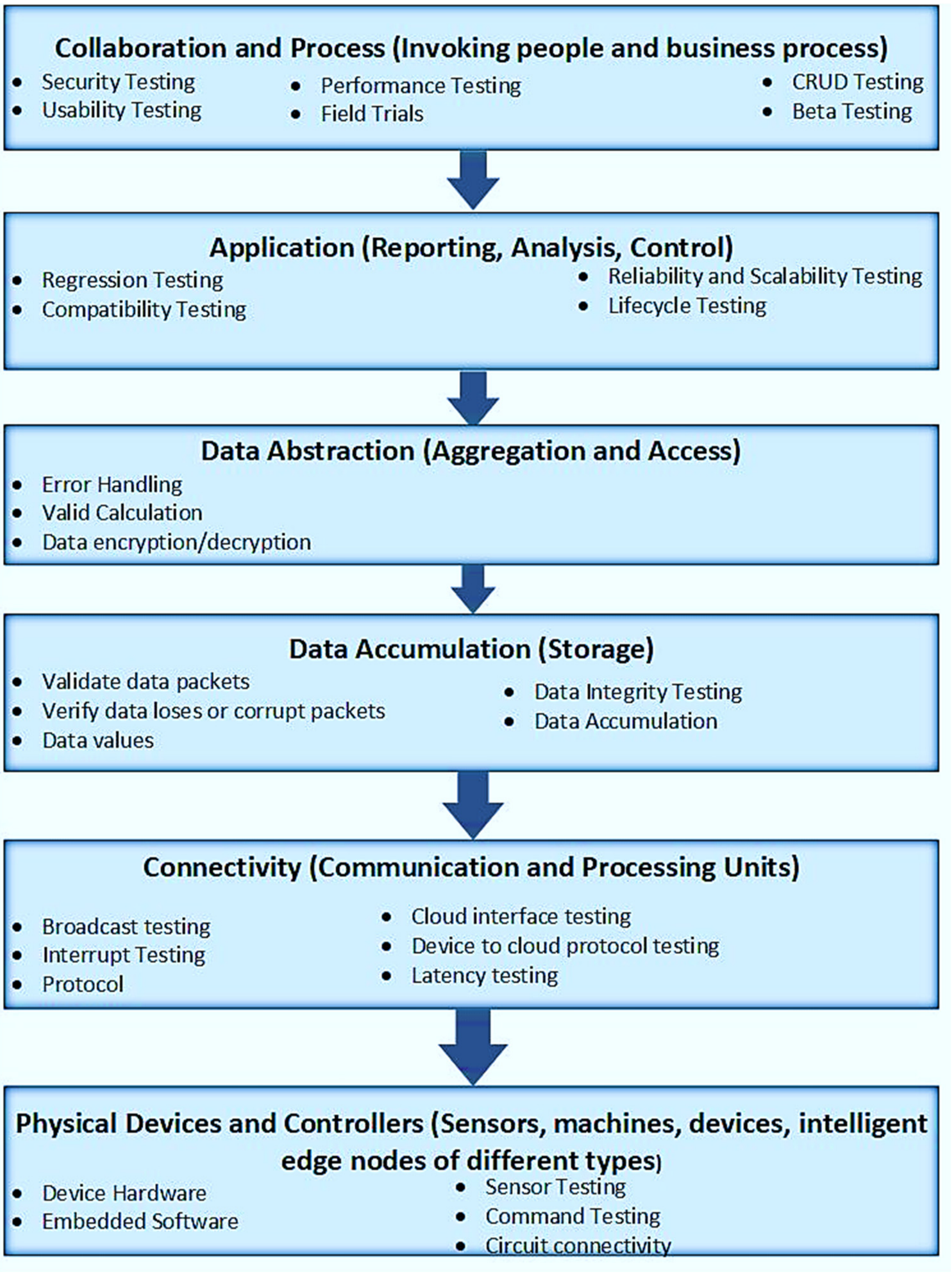
Collaboration and Process (Invoking people and business process)
Security Testing
Usability Testing
Performance Testing
Field Trials
CRUD Testing
Beta Testing
Application (Reporting, Analysis, Control)
Regression Testing
Compatibility Testing
Reliability and Scalability Testing
Lifecycle Testing
Data Abstraction (Aggregation and Access)
Error Handling
Valid Calculation
Data encryption/decryption
Data Accumulation (Storage)
Validate data packets
Verify data loses or corrupt packets
Data values
Data Integrity Testing
Data Accumulation
Connectivity (Communication and Processing Units)
Broadcast testing
Interrupt Testing
Protocol
Cloud interface testing
Device to cloud protocol testing
Latency testing
Physical Devices and Controllers (Sensors, machines, devices, intelligent edge nodes of different types)
Device Hardware
Embedded Software
Sensor Testing
Command Testing
Circuit connectivity

Figure 3. Mapping of important testing domains on the IoT's layered architecture.

It is very necessary to test each component before and after integration in various ways. The whole IoT device can fail due to a single component failure [38–40]. For instance, consider the Internet as the communication medium and issues with connectivity, where data are not able to be sent and received. In the case of a safety-critical system, testing becomes even more important. The following are the types of testing that must be performed at each layer in order to preserve the quality of an IoT device and deliver the best possible product.

4.1. Collaboration and Process Layer

The collaboration and process layer includes processes and people related to business. It involves business, people, their decision-making, and collaboration on the basis of the information that is extracted from the computation of the IoT. The overall interaction of people with the IoT device is validated in this, including mainly the non-functional properties like security, reliability, availability, and performance attributes.

The following are some types of testing to be performed for a quality product:

1. **Security Testing:** Physical and logical threats to the security of an IoT-embedded system are analyzed. One of the best practices for maintaining the security of an IoT device is to train employees on the importance of security on an IoT device. In the case of an outside service provider, their capability of maintaining security should be validated before obtaining any services from them. The software part must be well-protected with login identification and a strong password, keeping it protected from unauthorized access to personal information or manipulation of device usage [41–48].
2. **Performance Testing:** Performance testing deals with testing the overall performance of an IoT device. It includes checking the number of users that can be handled by the device at a time and how it reacts to the situation of excess users. It also deals with the recovery of a system in case of any issue or even if the number of users exceeds the response time of the device being tested at peak loads, as well as during normal working.
3. **Usability Testing:** Usability testing deals with how easy a system is to use by people of every age, culture, and physical or mental capability level. This includes people, contexts, activities, and technologies to be used. Usability testing can be performed on parameters like navigation, affordability, flexibility, consistency, control, recovery, constraints, conviviality, style, and visibility.
4. **CRUD Testing:** This means Creating, Reading, Updating, and Deleting. The response of a system is tested after performing these steps on the system to determine whether it allows new data to be added, created, accessed, and easily read by the user. A matrix is made for easy evaluation of the system. This is also a type of black box testing.
5. **Beta Testing:** Beta testing refers to testing in which a group of people test the product and give their feedback. These people can be any external testing party. This is basically acceptance testing by the user.
6. **Field Trials:** In field trials, the product is given to the user so that they can test it in a real environment rather than through some automation technique or artificial method.

4.2. Application Layer

The application layer comprises the software application, which works among the devices. In the application layer, the delivery of data is checked and reported. How the application manages to control the devices is also tested. This includes the following testing types:

1. **Regression Testing:** Regression testing is performed after adding new functionality to an application of the IoT device by testing the whole system again, validating its results, and checking how changes in the system affected the whole IoT device.
2. **Reliability and Scalability Testing:** Reliability and scalability testing deals with testing the system in terms of 'abilities', i.e., the non-functional attributes of the system.
3. **Compatibility Testing:** In compatibility testing, the compatibility of a software application is checked with other components and software in an IoT device. The compatibility of the operating system is also checked, and the type of database that is compatible with the current system type is kept in mind.
4. **Life-Cycle Testing:** Life-cycle testing includes the validation of every step of the system development life-cycle. It checks whether every step is properly followed or not.

4.3. Data Abstraction Layer

The abstraction of data provides an abstract view, rather than describing how the machine actually handles or stores something. It includes handling the occurrence of errors, as well as the encryption and decryption of data. The following are some types of tests to be performed for a quality product:

1. **Error Handling:** This deals with the detection, resolution, and anticipation of errors in an application and how it reacts in case of these errors, e.g., does it safely shut down and terminate the preprocess?
2. **Encryption and Decryption:** In the case of encryption and decryption, both the data sent and received should be the same; if they are not, then how will the system deal with the error and recovery of information?
3. **Valid Calculation:** The calculations are observed, and their validity is very important. If there is some mistake in the calculations, then the whole encryption and decryption process will be full of errors.

4.4. Data Accumulation Layer

Many problems can occur in the accumulation of data, as the data for the IoT are large, involving issues like variety, velocity, and volume. The aspects of high variety, volume, and velocity of data in motion are also considered in validating data packets. A hot path is needed for fast processing of data, whereas for processing of applications, a cold path is needed. The testing of all the data and their validity is very important. The following are major tests that must be performed:

1. **Validate Data Packets:** The data packets sent to the communication media and received on the other end should be the same with no error or noise.
2. **Data Integrity Testing:** Integrity of data means the quality of data. The data should be accurate, consistent, and complete.
3. **Data Accumulation:** Data accumulation deals with collecting the data and then validating their quality.
4. **Verify Data Loses or Corrupt Packets:** Every packet is observed and checked for the eradication of corrupt or lost data packets. There are few methods and techniques available to extract corrupted information or estimate its values.
5. **Data Values:** The data values must be correctly sent and received.

4.5. Connectivity Layer

The connectivity should be intelligent and secure, having the least possible delay in data transfer. The broadcasting, cloud interface, and protocols are checked in the connectivity layer. The following are types of testing that are conducted for a quality product in this layer:

1. **Broadcast Testing:** Broadcast testing ensures the quality of transmission and broadcasting of data. It is also known as a test pattern, test card, close-down, or start-up testing.
2. **Cloud Interface Testing:** The web traffic on the cloud being used for the IoT device and the function's validity are checked. The scalability, redundancy, and performance are also observed.
3. **Device-to-Cloud Protocol Testing:** The requirements for compatibility of the application of the IoT device are validated, and the compatibility of the application being used to interact with the cloud is also checked. There should not be any defects arising during connection of the cloud and the application.
4. **Latency Testing:** Latency testing checks the amount of time that the system takes to send and receive data. It should be kept to a minimum in order to provide the best-quality interface and services to the user.
5. **Interruption Testing:** In interruption testing, the response of the system in case of all possible interrupts is observed. The system should return to a normal working state in case of any interruption.

4.6. Physical Devices and Controller Layers

The testing of the whole circuit and its connections, along with the workings of sensors and actuators, is very important. The commands given to the devices through embedded software must also be validated. The following are some types of testing to be performed for a quality product:

1. **Sensor Testing:** All the sensors of the IoT device are validated by checking their outputs separately. Every sensor should work properly. If any sensor does not give the correct output, then the whole calculation can go wrong. This is crucial in the case of critical systems.
2. **Command Testing:** The commands given to the processor are validated in command testing. These commands should give the expected output.
3. **Circuit Connectivity:** The proper connectivity of the circuit is very important. If a single wire is detached accidentally or has a wrong connection, there can be a loss of any other component, like actuators, which affects the cost of the project. The connections should be tight and validated by different devices.
4. **Device Testing:** The proper functioning of every device separately attached, for providing services or assisting in them, must be verified.
5. **Embedded Software:** Embedded software testing deals with the testing of the operating system of the processor being used.

5. Results and Discussion

To evaluate our proposed framework, we conducted an online survey. We received 120 responses, out of which 28.3% of respondents were working in software industries, 48.3% were students of Computer Science or relevant degrees, 7.5% were faculty members of Computer Science or relevant departments, and the remaining respondents included freelancers, software developers in government organizations or research-based industries, etc. The foremost question of the survey form was whether people have witnessed failures of IoT-based applications in real time, such as Petnet [1]. About 83.3% of respondents replied 'yes'. The second question was to identify the phase of the software development life-cycle (SDLC) that requires more attention to reduce such failures of IoT-based applications. The results of the obtained answers are illustrated in Figure 4, demonstrated using a pie chart [43]. About 35% of the respondents identified 'Testing' as the most crucial phase of the SDLC which needs more attention to avoid failures of IoT-based applications in real time.

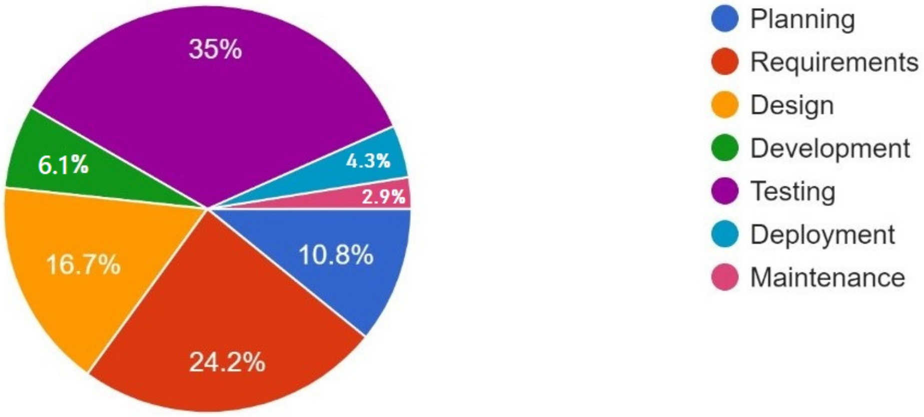

Figure 4. An illustration of the dependency of failures of IoT-based applications.

As customer satisfaction is dependent on quality, the third question was to measure the dependency of quality on the testing phase of the SDLC. Figure 5 shows the obtained results. In all, 13.7% of the respondents were of the view that about 80–100% of the quality is dependent on testing, 45.8% of the respondents selected 50–70%, 25.8% chose 40–60%, and only 5.8% selected 20–40%.

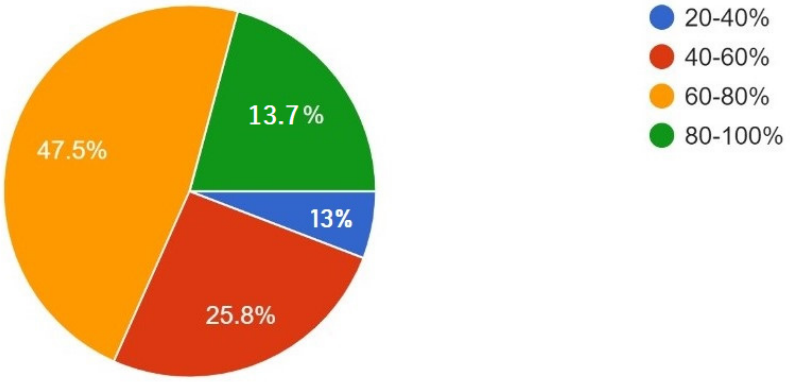

Figure 5. An illustration of the dependency of quality of IoT-based applications.

The respondents were also asked about the percentage of failure minimization if testing is properly performed. The responses of the respondents are illustrated in Figure 6. The largest proportion of respondents thought that 60–80% of the failure can be minimized if testing is properly performed. These obtained responses show the importance of the testing phase of the SDLC, which must be properly performed to maintain quality and customer satisfaction. As the IoT is a huge network of physical objects integrated with sensors, software, and communication technologies, the respondents were asked if conducting all types of tests in the IoT is possible and easy. The obtained results are shown in Figure 7. About 40.8% of respondents replied ‘No’, whereas 40% replied ‘Maybe’, and 19.2% of the respondents replied ‘Yes’.

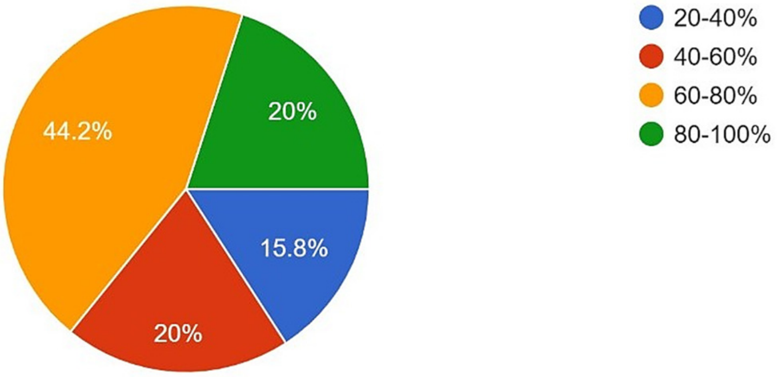

Figure 6. An illustration of the minimization of failures if testing is performed properly.

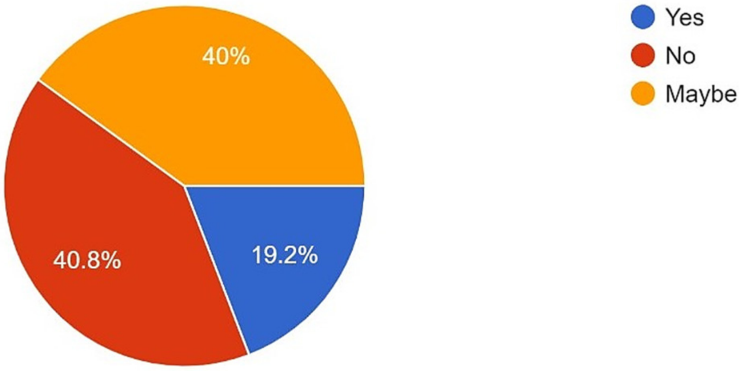

Figure 7. An illustration of the possibility of conducting all types of tests during QA.

The respondents were asked if conducting all types of tests during the QA process of IoT-based applications is time-consuming and decelerates the SDLC by posing many other challenges as well. About 81.2% of the respondents replied ‘Yes’, which shows that conducting all testing techniques for quality in IoT-based applications is not feasible. To measure the need for our proposed framework, we asked the respondents whether there is a need to identify all important testing techniques that must be conducted for the quality assurance of IoT-based applications, thus accelerating the SDLC and reducing cost. The obtained results are illustrated in Figure 8.

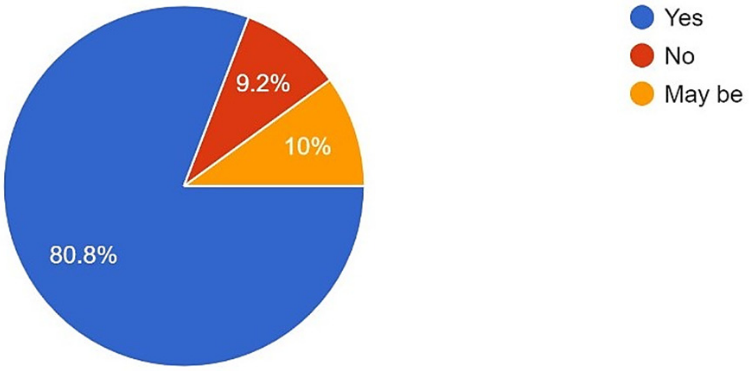

Figure 8. An illustration of the need for a framework that identifies all important tests.

About 80.8% of the respondents replied 'Yes'. This shows that there is a need for some framework that assists quality assurance engineers in identifying all the necessary testing techniques that must be conducted to ensure quality in IoT-based applications. The respondents were also asked how much our proposed framework or idea would assist in improving the quality of IoT-based applications. The results are shown in Figure 9.

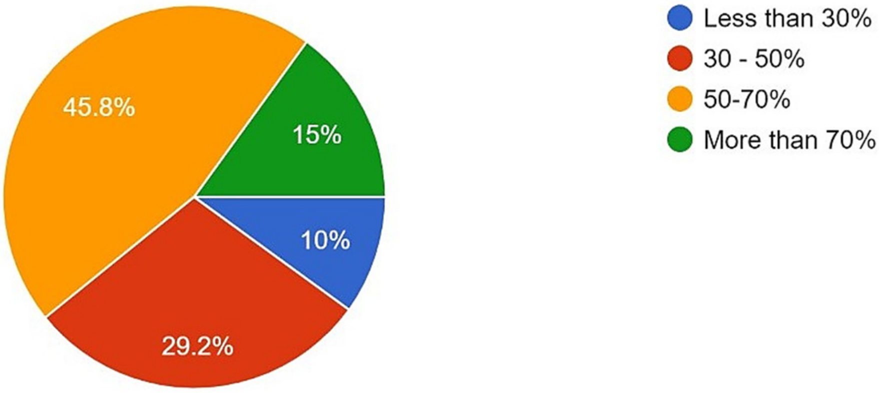

Figure 9. An illustration of how much our proposed framework will improve the quality of IoT-based applications.

Different testing techniques were identified in Figure 1. To evaluate our identified testing techniques, we asked respondents to select the most important techniques for the device level, cloud level, mobile level, and end-to-end level in IoT-based applications. They were also provided with an option to suggest any other techniques that they considered important. The obtained results for device-level testing are illustrated in Figure 10. The largest proportion of respondents considered embedded software trials and sensor testing as the most important testing techniques. Other than the identified testing techniques, the respondents also gave suggestions for communication and security testing, which must be performed at the device level. The results obtained for mobile-level testing are illustrated in Figure 11.

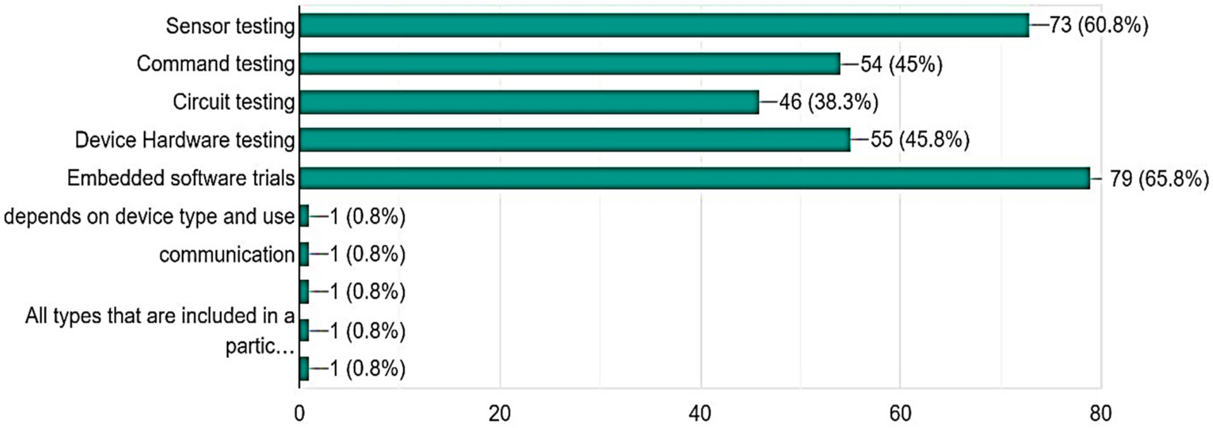

Figure 10. An illustration by percentage of important testing techniques for the device level in IoT-based applications.

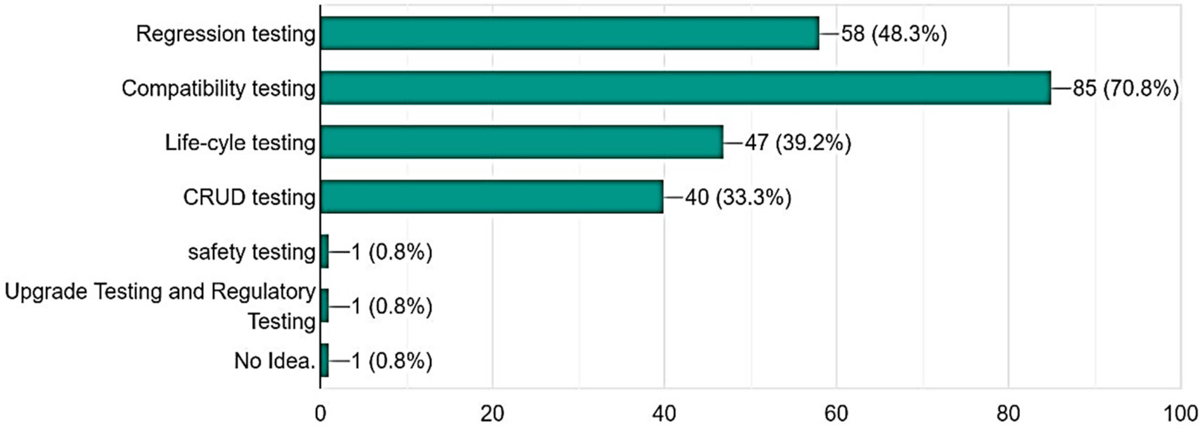

Figure 11. An illustration by percentage of important testing techniques for the mobile level in IoT-based applications.

The largest proportion of respondents considered compatibility testing as the most crucial testing technique for mobile-level testing. Safety testing, upgrade, and regulatory testing were suggested by a few respondents and must be incorporated into our proposed framework. The results obtained for end-to-end-level testing are illustrated in Figure 12.

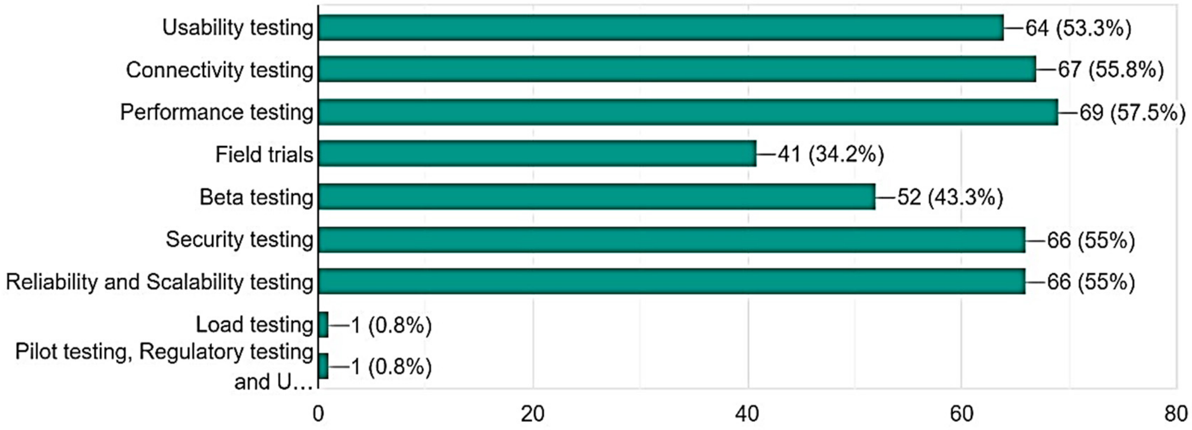

Figure 12. An illustration by percentage of important testing techniques for the end-to-end level in IoT-based applications.

All testing techniques that we incorporated for end-to-end-level testing were considered important by the majority of respondents. Other testing techniques which were suggested by respondents included load testing, pilot testing, upgrade, and regulatory testing. The results obtained for cloud-level testing are illustrated in Figure 13. Data encryption or decryption testing, cloud interface testing, and tests for error handling were considered the most important testing techniques for cloud-level testing in IoT-based applications. No other techniques were suggested by our respondents for this level. From the survey, we found that no testing technique was considered unimportant by our respondents. To improve our framework in the future, we can also include those testing techniques that were suggested by our respondents. By considering and performing all the identified testing techniques, we can improve the quality of IoT-based applications and reduce their failures in real time. Quality is the utmost requirement for customer satisfaction and must be fulfilled.

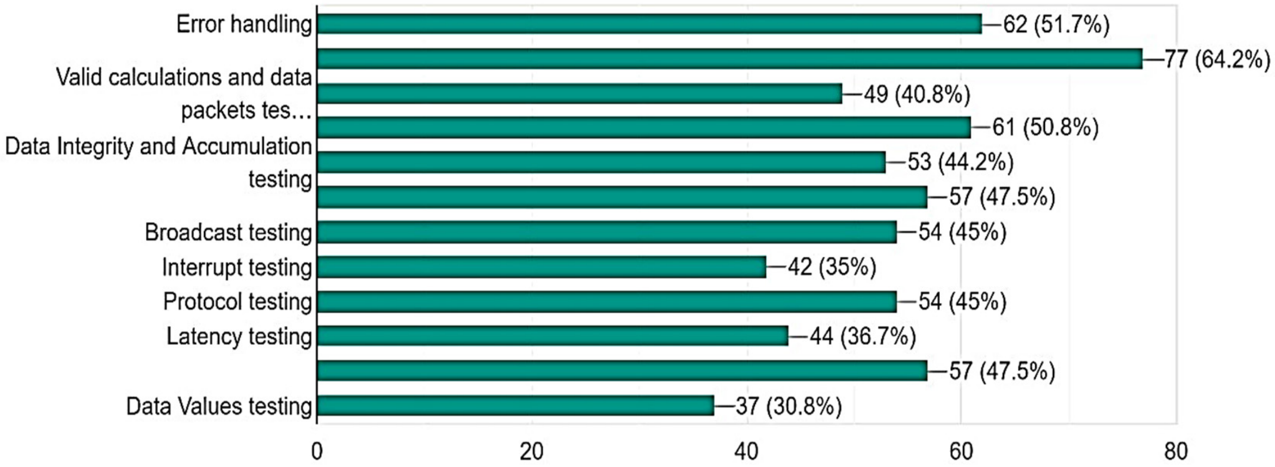

Figure 13. An illustration by percentage of important testing techniques for the cloud level in IoT-based applications.

6. Conclusions and Future Work

Overall, this research contributed to identifying all the important testing techniques for IoT-based applications and proposed a robust framework that assists quality assurance engineers in performing necessary tests during the QA process, ensuring quality with maximum customer satisfaction, reducing failure rates and costs, and, lastly, accelerating the SDLC. We also mapped all the necessary tests across the IoT's layered architecture to help the testers identify tests for each specific layer. Although these approaches are fundamental, they still lay the foundation of quality assurance in the IoT. This area needs special attention in IoT development, as the idea behind the IoT is smart living. This goal is not achievable without user satisfaction, for which quality is the utmost requirement that needs to be fulfilled. Testing should be introduced as early as possible with reviews, inspection, and formal methods. The quality of the IoT is strongly linked with security and safety as these networks are mostly linked to real-time conditions. Thus, it is important to concentrate more on data security and safety. The quality of the hardware being used should also be considered very important, as hardware from different vendors provides different performance and compatibility with IoT-based applications. Compatibility testing, load testing for traffic on the network, and functional testing for both Internet and non-Internet applications must be performed for all three types of clouds, i.e., public, private, and hybrid clouds. Product analytics is also very important as it can harmonize processes and test data. It can improve quality, yield, and productivity significantly. IoT-based applications can be tested using simulators that mimic the real environment to verify performance, usability, and all other concerns. Artificial intelligence (AI) and machine learning (ML) techniques offer the potential to enhance data collection processes, making them more efficient, adaptable, and secure. Integrating customer feedback early on and including human input in the testing loop can further elevate the quality. Considering sustainability, the proposed framework not only enhances reliability but also contributes to reducing resource consumption in IoT applications. In future work, our aim is to employ neural networks for end-to-end testing automation in the IoT, ensuring a sustainable and efficient quality assurance process. This automation will not only mitigate the risk of system failures in real time but also lead to cost savings, as it eliminates the need for human intervention.

Author Contributions: Conceptualization, A.A. and S.S.; methodology, A.A. and M.I.S.; software, S.S., M.I.S., M.A. and M.U.; validation, M.I.S. and A.R.A.; formal analysis, A.R.A.; investigation, A.A., M.A. and A.R.A.; resources, M.U., A.A. and S.S.; data curation, M.I.S.; writing original draft preparation, A.A. and S.S.; writing review and editing, A.A. and S.S.; visualization, M.A. and M.U.; supervision, M.A. and M.U. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Deanship of Scientific Research at University of Tabuk through Research no. S-0151-1443.

Institutional Review Board Statement: Not Applicable.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: The data presented in this study are available in the article.

Acknowledgments: The authors extend their appreciation to the Deanship of Scientific Research at University of Tabuk for funding this work through Research no. S-0151-1443.

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

References

1. Townsend, J. Gradual Transformation to Secure Cloud Operations. 2016. Available online: https://www.ctl.io/blog/post/qawith-the-iot/ (accessed on 1 January 2023).
2. Al-Fagih, A.E.; Al-Turjman, F.M.; Alsalih, W.M.; Hassanein, H.S. A Priced Public Sensing Framework for Heterogeneous IoT Architectures. *IEEE Trans. Emerg. Top. Comput.* **2013**, *1*, 133–147. [CrossRef]
3. Kiljander, J.; D'Elia, A.; Morandi, F.; Hyttinen, P.; Takalo-Mattila, J.; Ylisaukko-Oja, A.; Soininen, J.-P.; Cinotti, T.S. Semantic Interoperability Architecture for Pervasive Computing and Internet of Things. *IEEE Access* **2014**, *2*, 856–873. [CrossRef]
4. Zhou, J.; Hu, L.; Wang, F.; Lu, H.; Zhao, K. An efficient multidimensional fusion algorithm for IoT data based on partitioning. *Tsinghua Sci. Technol.* **2013**, *18*, 369–378. [CrossRef]
5. Leal, A.G.; Santiago, A.; Miyake, M.Y.; Noda, M.K.; Pereira, M.J.; Avanço, L. Integrated environment for testing IoT and RFID technologies applied on the intelligent transportation system in Brazilian scenarios. In Proceedings of the 2014 IEEE Brasil RFID, Sao Paulo, Brasil, 25 September 2014; pp. 22–24.
6. Li, F.; Xiong, P. Practical Secure Communication for Integrating Wireless Sensor Networks into the Internet of Things. *IEEE Sens. J.* **2013**, *13*, 3677–3684. [CrossRef]
7. Mao, X.; Zhou, C.; He, Y.; Yang, Z.; Tang, S.; Wang, W. Guest editorial: Special issue on wireless sensor networks, cyber-physical systems, and internet of things. *Tsinghua Sci. Technol.* **2011**, *16*, 559–560. [CrossRef]
8. Reetz, E.S.; Kuemper, D.; Moessner, K.; Tönjes, R. How to Test IoT-based Services before Deploying them into Real World. In Proceedings of the European Wireless 2013; 19th European Wireless Conference, Guildford, UK, 16–18 April 2013; pp. 1–6.
9. Kuemper, D.; Reetz, E.S.; Tönjes, R. Test derivation for semantically described IoT services. In Proceedings of the 2013 Future Network & Mobile Summit, Lisboa, Portugal, 3–5 July 2013; p. 13851815.
10. Gimenez, P.; Molina, B.; Palau, C.E.; Esteve, M. SWE Simulation and Testing for the IoT. In Proceedings of the 2013 IEEE International Conference on Systems, Man, and Cybernetics, Manchester, UK, 13–16 October 2013; pp. 356–361.
11. Fernandes, J.; Nati, M.; Loumis, N.S.; Nikoletseas, S.; Raptis, T.P.; Krco, S.; Rankov, A.; Jokic, S.; Angelopoulos, C.M.; Ziegler, S. IoT Lab: Towards co-design and IoT solution testusing the crowd. In Proceedings of the 2015 International Conference on Recent Advances in Internet of Things (RIoT), Singapore, 7–9 April 2015; pp. 1–6.
12. Desnitsky, V.A.; Kotenko, I.V.; Nogin, S.B. Detection of anomalies in data for monitoring of security components in the Internet of Things. In Proceedings of the 2015 XVIII International Conference on Soft Computing and Measurements (SCM), St. Petersburg, Russia, 19–21 May 2015; pp. 189–192. [CrossRef]
13. Chuang, C.; Cheng, W.; Hsu, K. A comprehensive composite digital services quality assurance application on the intelligent transportation system. In Proceedings of the 2015 17th Asia-Pacific Network Operations and Management Symposium (APNOMS), Busan, Republic of Korea, 19–21 August 2015; pp. 368–371.

14. Masirap, M.; Amaran, M.H.; Yussoff, Y.M.; Rahman, R.A.; Hashim, H. Evaluation of reliable UDP-based transport protocols for the Internet of Things (IoT). In Proceedings of the 2016 IEEE Symposium on Computer Applications and Industrial Electronics (ISCAIE), Batu Feringghi, Malaysia, 30–31 May 2016; pp. 200–205.
15. Sankaran, S. Lightweight security framework for IoTs using identity based cryptography. In Proceedings of the 2016 International Conference on Advances in Computing, Communications, and Informatics (ICACCI), Communications, Jaipur, India, 21–24 September 2016; pp. 880–886. [CrossRef]
16. Chandan, A.R.; Khairnar, V.D. Security Testing Methodology of IoT. In Proceedings of the 2018 International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, 11–12 July 2018; pp. 1431–1435.
17. Kim, D.; Lee, E.; Kang, S. IJPoster: Expediting IoT Application Testing. In Proceedings of the International Conference on Mobile Systems, Applications, and Services, Seoul, Republic of Korea, 17–21 June 2019; pp. 572–573.
18. Kaiser, A.; Hackel, S. Standards-Based IoT Testing with Open-Source Test Equipment. In Proceedings of the 2019 IEEE 19th International Conference on Software Quality, Reliability and Security Companion (QRS-C), Sofia, Bulgaria, 22–26 July 2019; pp. 435–441. [CrossRef]
19. Abdallah, M.; Jaber, T.; Alabwaini, N.; Alnabi, A.A. A Proposed Quality Model for the Internet of Things Systems. In Proceedings of the 2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT), Amman, Jordan, 9–11 April 2019; pp. 23–27.
20. Popereshnyak, S.; Suprun, O.; Suprun, O.; Wieckowski, T. IoT application testing features based on the modeling network. In Proceedings of the 2018 XIVth International Conference on Perspective Technologies and Methods in MEMS Design (MEMSTECH), Lviv, Ukraine, 18–22 April 2018; pp. 127–131.
21. Kim, H.; Ahmad, A.; Hwang, J.; Baqa, H.; Le Gall, F.; Ortega, M.A.R.; Song, J. IoT-TaaS: Towards a Prospective IoT Testing Framework. *IEEE Access* **2018**, *6*, 15480–15493. [CrossRef]
22. Papachristou, K.; Theodorou, T.; Papadopoulos, S.; Protogerou, A.; Drosou, A.; Tzovaras, D. Runtime and Routing Security Policy Verification for Enhanced Quality of Service of IoT Networks. In Proceedings of the 2019 Global IoT Summit (GIoTS), Aarhus, Denmark, 17–21 June 2019; p. 18833635.
23. Temkar, R.; Chakrabarti, P.; Jena, O.P.; Elngar, A.A.; Margala, M.; Ravi, V. Multi-attribute quality score computation for Internet of Things (IoT) based applications. *Res. Sq.* **2022**. [CrossRef]
24. Julio, Y.R.; Contreras, B.H.; Rivera, S.C.; López, C.C.; Mangonez, A.D.P.; Herazo, H.B. Framework to Manage Software Quality on IIoT Apps. *IOP Conf. Ser. Mater. Sci. Eng.* **2021**, *1154*, 012006. [CrossRef]
25. Medhat, N.; Moussa, S.; Badr, N.; Tolba, M.F. Testing Techniques in IoT-based Systems. In Proceedings of the 2019 Ninth International Conference on Intelligent Computing and Information Systems (ICICIS), Cairo, Egypt, 8–10 December 2019; pp. 394–401. [CrossRef]
26. Sharma, A.; Sarje, A.K. Testing Techniques for IoT Systems: A Review. *Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol.* **2020**.
27. Kumari, R.; Soni, M.K. A Comprehensive Study of Quality Assurance Techniques in the Internet of Things. *Int. J. Adv. Res. Comput. Sci. Softw. Eng.* **2020**.
28. Chen, S.; Xu, H.; Liu, D.; Hu, B.; Wang, H. A Vision of IoT: Applications, Challenges, and Opportunities with China Perspective. *IEEE Internet Things J.* **2014**, *1*, 349–359. [CrossRef]
29. Lanzisera, S.; Weber, A.; Liao, A.; Pajak, D.; Meier, A. Communicating Power Supplies: Bringing the Internet to the Ubiquitous Energy Gateways of Electronic Devices. *IEEE Internet Things J.* **2014**, *1*, 153–160. [CrossRef]
30. Kypus, L.; Vojtech, L.; Hrad, J. Security of ONS service for applications of the Internet of Things and their pilot implementation in academic network. In Proceedings of the 2015 16th International Carpathian Control Conference (ICCC), Szilvasvarad, Hungary, 27–30 May 2015; pp. 271–276. [CrossRef]
31. Marinissen, E.J.; Zorian, Y.; Konijnenburg, M.; Huang, C.T.; Hsieh, P.H.; Cockburn, P.; Delvaux, J.; Rožić, V.; Yang, B.; Singelée, D.; et al. IoT: Source of test challenges. In Proceedings of the 2016 21st IEEE European Test Symposium (ETS), Amsterdam, The Netherlands, 23–27 May 2016; p. 16159898.
32. Zanella, A.; Bui, N.; Castellani, A.; Vangelista, L.; Zorzi, M. Internet of Things for Smart Cities. *IEEE Internet Things J.* **2014**, *1*, 22–32. [CrossRef]
33. Khan, M.A.; Salah, K. IoT security: Review, blockchain solutions, and open challenges. *Future Gener. Comput. Syst.* **2018**, *82*, 395–411. [CrossRef]
34. Power, D.; Curry, D.; Pierson, R.; Lawrence, C. Petnet'sfailure Is a Warning to IoT developers. Available online: http://readwrite.com/2016/08/01/petnet-shows-happens-iot-fails-dl1/ (accessed on 1 January 2023).
35. Naveed, M.; Usman, S.M.; Satti, M.I.; Aleshaiker, S.; Anwar, A. Intrusion Detection in Smart IoT Devices for People with Disabilities. In Proceedings of the 2022 IEEE International Smart Cities Conference (ISC2), Pafos, Cyprus, 26–29 September 2022; p. 22187135. [CrossRef]
36. Allerin.com. IoT Solutions Failure Points that You Should Be Aware of. 2016. Available online: https://www.allerin.com/blog/6-iotfailure-points-that-you-should-be-aware-of (accessed on 1 January 2023).
37. Training an Artificial Neural Network—Intro, solver. 2016. Available online: http://www.solver.com/training-artificial-neural-networkintro (accessed on 1 September 2023).
38. Irshad, A.; Mallah, G.A.; Bilal, M.; Chaudhry, S.A.; Shafiq, M.; Song, H. SUSIC: A Secure User Access Control Mechanism for SDN-Enabled IIoT and Cyber–Physical Systems. *IEEE Internet Things J.* **2023**, *10*, 16504–16515. [CrossRef]

39. Chaudhry, S.A.; Irshad, A.; Khan, M.A.; Khan, S.A.; Nosheen, S.; AlZubi, A.A.; Zikria, Y.B. A lightweight authentication scheme for 6G-IoT enabled maritime transport system. *IEEE Trans. Intell. Transp. Syst.* **2021**, *24*, 2401–2410. [CrossRef]
40. Chaudhry, S.A.; Irshad, A.; Yahya, K.; Kumar, N.; Alazab, M.; Bin Zikria, Y. Rotating behind Privacy: An Improved Lightweight Authentication Scheme for Cloud-based IoT Environment. *ACM Trans. Internet Technol.* **2021**, *21*, 1–19. [CrossRef]
41. Said, G.; Ghani, A.; Ullah, A.; Azeem, M.; Bilal, M.; Kwak, K.S. Light-weight secure aggregated data sharing in IoT-enabled wireless sensor networks. *IEEE Access* **2022**, *10*, 33571–33585. [CrossRef]
42. Hasan, S.S.U.; Ghani, A.; Din, I.U.; Almogren, A.; Altameem, A. IoT Devices Authentication Using Artificial Neural Network. *Comput. Mater. Contin.* **2022**, *70*, 3701–3716. [CrossRef]
43. Haq, M.I.U.; Khalil, R.A.; Almutiry, M.; Sawalmeh, A.; Ahmad, T.; Saeed, N. Robust graph-based localization for industrial Internet of things in the presence of flipping ambiguities. *CAAI Trans. Intell. Technol.* 2023; early view.
44. Irshad, A.; Usman, M.; Chaudhry, S.A.; Naqvi, H.; Shafiq, M. A provably secure and efficient authenticated key agreement schemefor energy internet-based vehicle-to-grid technology framework. *IEEE Trans. Ind. Appl.* **2020**, *56*, 4425–4435.
45. Kipongo, J.; Esenegho, E.; Swart, T.G. Efficient topology discovery protocol using IT-SDN for software-defined wireless sensor network. *Bull. Electr. Eng. Inform.* **2022**, *11*, 256–269. [CrossRef]
46. Esenogho, E.; Djouani, K.; Kurien, A.M. Integrating Artificial Intelligence Internet of Things and 5G for Next-Generation Smart grid: A Survey of Trends Challenges and Prospect. *IEEE Access* **2022**, *10*, 4794–4831. [CrossRef]
47. Alzahrani, B.A.; Irshad, A.; Alsubhi, K.; Albeshri, A. A secure and efficient remote patient-monitoring authentication protocol for cloud-IoT. *Int. J. Commun. Syst.* **2020**, *33*, e4423. [CrossRef]
48. Kantarci, B.; Mouftah, H. Trustworthy Sensing for Public Safety in Cloud-Centric Internet of Things. *IEEE Internet Things J.* **2014**, *1*, 360–368. [CrossRef]

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