EDITOR: Schahram Dustdar, dustdar@infosys.tuwien.ac.at

DEPARTMENT: INTERNET OF THINGS, PEOPLE, AND PROCESSES

Addressing the Faults Landscape in the Internet of Things: Toward Datacentric and System Resilience

Sultan Altarrazi, *King Abdulaziz University, Jeddah, 21589, Saudi Arabia*
Tomasz Szydlo, *Newcastle University, NE1 7RU, Newcastle upon Tyne, U.K.*
Schahram Dustdar, *Vienna University of Technology, 1040 Vienna, Austria*
Satish Narayana Srirama, *University of Hyderbad, Hyderabad, 500046, India*
Rajiv Ranjan, *Newcastle University, NE1 7RU, Newcastle upon Tyne, U.K.*

In the Internet of Things (IoT) context, the landscape of weaknesses in the IoT spectrum sheds light on addressing faults by researchers due to the number of IoT components that unveil immense vulnerabilities to failures. Hence, there is a need to comprehend the faults dynamics to facilitate identifying potential hazards in a developer's design, deliver methodologies to mitigate the risks, and ensure the data quality and resiliency of the IoT's deployment. This article comprehensively aims to analyze faults occurrences in the IoT, their impacts on functionality, and their repercussions on data. It highlights the intricate patterns of data faults by addressing various aspects, such as duration, cause, pitfalls, component, type, and source.

Internet of Things (IoT) systems, due to their inherent integration of diverse heterogeneous components, are often subject to complexity. As previously demonstrated by our research,[1] *osmotic computing* provides several benefits within these diverse IoT ecosystems. Such systems enable a vast flow of data, services, and applications across various domains, such as smart homes, smart cities, and health care. However, IoT components are prone to rapid degradation or malfunction, among other failure types. As suggested by Norris et al.,[2] these faults can occur at any stage of the IoT architecture (Figure 1). Given the intricate correlations that IoT systems generate, there is an increased risk of fault propagation among various components. IoT devices produce voluminous data, which are subsequently analyzed by AI/machine learning (ML) models for critical decision making, heavily relying on the quality of the data. Thus, the nature of faults profoundly impacts the IoT system, involving elements ranging from hardware and software to networks.

Erroneous data can lead to faulty decisions, emphasizing the need for robust methodologies to maintain high-quality and instantaneous decision making. For instance, age of data,[3] a quantifiable measure we introduced, can be used to determine and evaluate the temporal accuracy of data transmission from IoT devices. Our framework, IoT-QWatch,[4] can assess uncertainties in the IoT environment, proposing quality metrics like accuracy and timeliness at various IoT stages. Understanding IoT faults can help with improving the system's reliability, stability, and efficiency as well as with devising resilient methodologies for fault detection and rectification. Hence, a deep-rooted understanding of fault characteristics and types is essential, as highlighted by prior research,[5] which addresses the potential fault range. Our focus in this article on the IoT data faults taxonomy (IoDFT) aims to enable academics and researchers to better understand the intricacies and origins of these faults, fostering a deep comprehension of the faults' nature and origins. We underscore the

1089-7801 © 2023 IEEE
Digital Object Identifier 10.1109/MIC.2023.3300508
Date of current version 6 November 2023.

FIGURE 1. Various layers of the Internet of Things (IoT) architecture, starting from the perception layer where the data are generated, advancing through the communication and computing stages as well as storing data, and culminating in a decision. API: application programming interface.

significance of realizing that the core element of the IoT—*data*—can lead to erroneous decisions when influenced by faults.

IoT systems incorporate many components, with data forming a fundamental pillar. As an example, take urban evolution, which necessitates the deployment of resilient methodologies across interconnected IoT components at various stages. Given the heterogeneity and diversity of IoT components, there is an inevitable escalation in the volume of data flowing within the IoT pipeline. Concerning this multiplicity and complexity, ML techniques can potentially amplify the utility of the aforementioned data pillar, offering significant insights and aiding decision making within the IoT system. The application of ML in the IoT could significantly enhance the functionality of devices not only in homes and cities but also in industries.[6]

This article discusses data faults taxonomy, emphasizing that low-quality data may lead to catastrophic failures in IoT systems. Therefore, a datacentric (DC) approach is essential for the success of the IoT. By leveraging DC strategies, we can efficiently analyze the vast amounts of data IoT devices generate and extract valuable insights. This pursuit stems from a deep awareness of vulnerabilities and the heterogeneous characteristics of the IoT landscape. Additionally, we explicate the facets of IoDFT. Finally, we explore and present possible future research inquiries.

SCENARIO

We discuss a hypothetical scenario regarding the road traffic congestion challenge in a smart city due to increased vehicle demand. Smart city market growth of 24.53%, the compound annual growth rate, is expected to be reached by 2027.[7] As depicted in Figure 2, roadside units (RSUs) support wireless communication between the units and the board on the vehicles. There are typically three communication types in this scenario: vehicle to vehicle, vehicle to RSU, and RSU to RSU. A dysfunctional RSU resulted in misdirection to vehicles, leading to heavy traffic. The RSU is not apprised of the events occurring near the stadium, which may have led to the lack of essential information

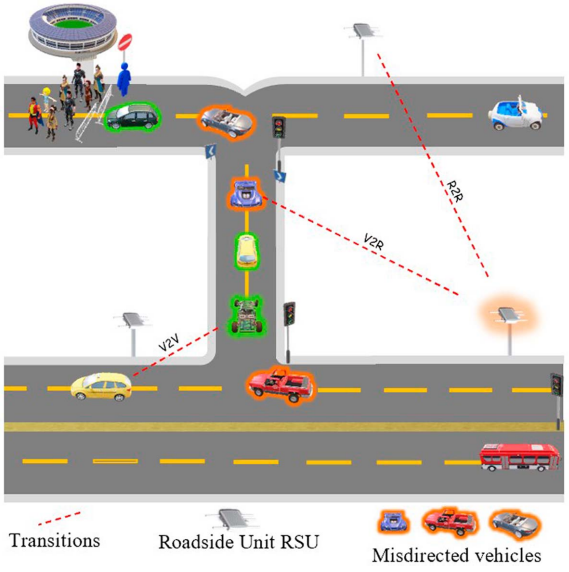

FIGURE 2. In this scenario, upon the end of a football game, a dysfunctional roadside unit (RSU) erroneously incorrectly directed the vehicles. Instead of turning right, vehicles turned left while spectators vacated the stadium, leading to a temporary road closure due to the high pedestrian traffic. (Source for 3-D images: Microsoft; used with permission.) R2R: RSU to RSU; V2R: vehicle to RSU.

or inaccurate data. Failing to attend to the aforementioned problem may result in serious ramifications, such as increased emissions, interruptions to vehicular circulation, and substantial disturbance to the city's transportation infrastructure.

It is of utmost importance that prompt and effective measures are taken to address this issue immediately and prevent any further unfavorable consequences. Multiple underlying factors are possibly contributing to the previous predicament, such as potential malfunctions in the hardware of the onboard unit in the vehicle, updates pertaining to computing, or hindrances in communication regarding packet delivery. However, various ML resilient methodologies have been introduced, in the areas of wireless sensor networks and vehicular ad hoc networks.[8] Utilizing ML algorithms in these scenarios is a substantial matter to lessen potential urban problems. Therefore, data quality is the key point in the traffic control system in a smart city.

IoT FAULTS

The following terms are used to address faults in the IoT: *failures*, *errors*, *flaws*, *anomalies*, *uncertainties*, and *defects*.[9,10,11,12,13] Faults might result from hardware failures, software bugs, environmental conditions, network problems, or human mistakes. Additionally, faults in the IoT realm can range from minor setbacks to significant problems, disrupting the entire system. Likewise, faults can be categorized based on the component of the defect.

We cannot overemphasize the criticality of obtaining trustworthy data readings from sensors in the IoT. It is indispensable to thoroughly comprehend the potential faults that may arise in IoT data, as they play an instrumental role in augmenting the stability and effectiveness of IoT systems. Consequently, this leads to developing robust strategies and solutions that prevent, detect, and rectify faults. Faults can emerge in diverse guises and contexts, underscoring the need for a comprehensive approach to tackle them.[9]

FAULT TAXONOMY

The IoDFT, as depicted in Figure 3, is a taxonomy that systematizes numerous defects that might surface in the IoT. Various articles[14,15] have discussed classification approaches for specific domains. Our taxonomy followed the faceted analysis approach since the IoT includes interacting components, including devices, servers, networks, and analytics software. Each can incur defects that can significantly disrupt the efficacy, reliability, and quality, among other characteristics. Additionally, faults are interrelated and might influence other components to initiate upcoming and unknown cascaded defects, which occur when the fault of one system component causes the failure of other affiliated components. Generally speaking, providing the IoDFT can pinpoint the IoT system fault's nature.

We conducted an exhaustive analysis that included the source of the fault, data pitfalls, form, source, duration, and cause. We have highlighted the most prevalent data pitfalls, e.g., missing data, erroneous data, spikes, and stuck-at. Moreover, we classified the causes associated with the component. Each component has a substantial impact on the IoT system. For instance, physical damage caused by an environmental circumstance leads to a defective sensor. A further discussion of these aspects is provided in the following sections.

The IoDFT analysis requires technical expertise covering three stages: stage one, stage two, and stage three (Figure 4).

The first stage enables the discovery of the initial characteristics of the fault raised in the system. It encompasses four different classes derived from the base *fault*. The *component* class is where the fault originated: *hardware*, *software*, or *network*. The *source* class is the fault's location and scope: either *internally* or *externally* or on a *single* node or *multiple* nodes. The *duration* class pertains to the periodicity and permanence time of the fault in the system. The *type* comprises three distinct elements, each of which illuminates a distinct feature of the underlying attribute: point, contextual, and collective. The second stage of IoDFT analysis reveals several noteworthy data pitfalls, including incomplete or inaccurate data, delayed response times, and power outages. Finally, in the third stage of fault analysis, the potential cause of the malfunction linked to the defective origin component is provided.

DC

The burgeoning adoption of smart sensors in everyday objects is expected to yield a practical framework for both environmental and physical realms, revolutionizing our interaction with these realms in the coming years. Here, the role of the IoT is pivotal, spanning domains from smart homes and agribusiness to economics, all of which heavily rely on data processing of the IoT pipeline. Researchers have proposed various definitions of DC approaches,[16,17,18,19] and there is universal consensus that a DC approach primarily focuses on enhancing data quality for decision-making processes that rely on models. In contrast, a modelcentric[20] approach emphasizes improving the model itself. DC techniques can be implemented during model training,

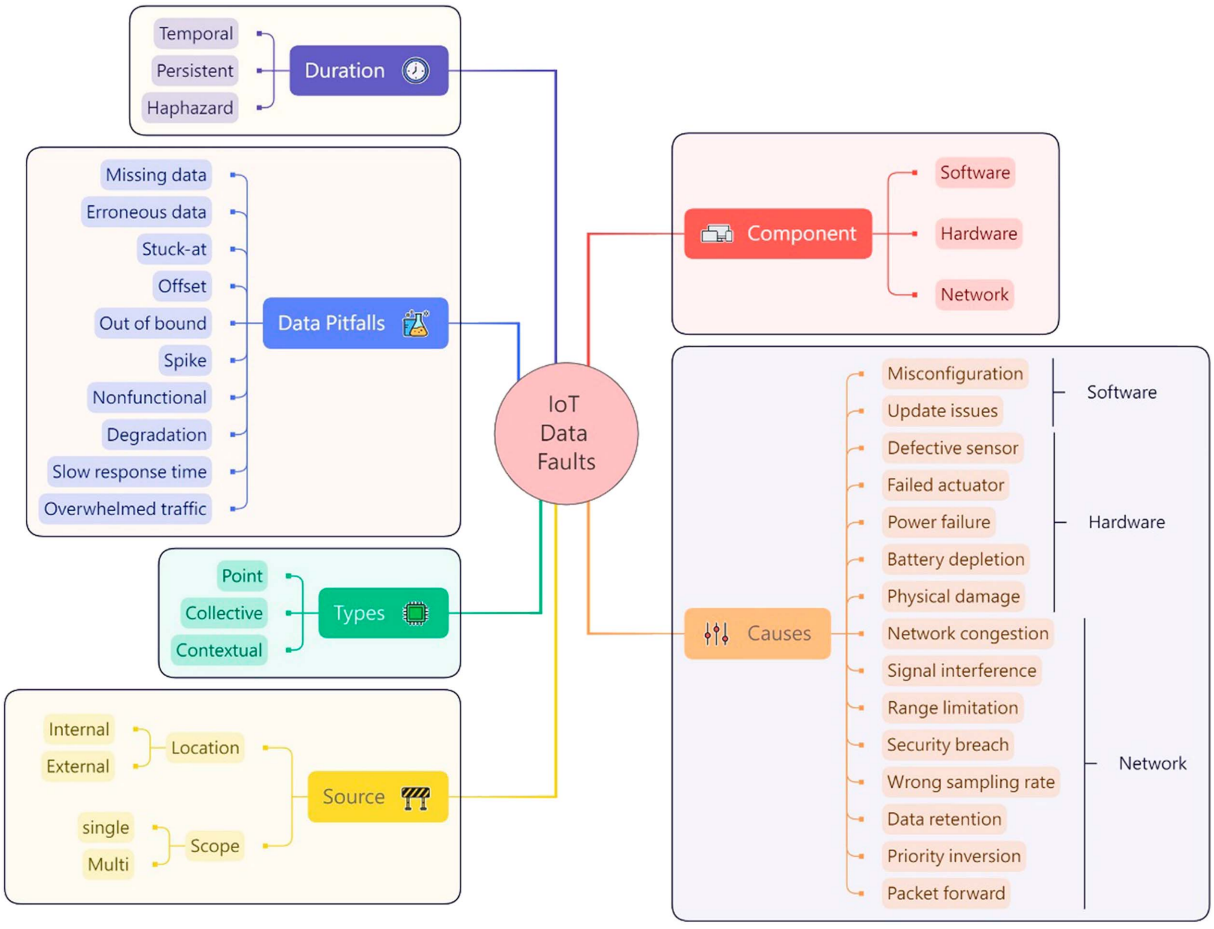

FIGURE 3. The IoT data fault taxonomy (IoDFT).

including augmentation and labeling. The decisions derived from the model can be improved through iterative curation and data maintenance. Continuous data faults necessitate sophisticated strategies, such as DC techniques, to enhance data credibility and integrity. Broadly, DC aims to maximize data quality to improve model performance.

Several factors, including sensor failure, malfunction, malicious attacks, human errors, or software bugs, can introduce uncertainties in the retrieved data. Conducting a thorough analysis of the fault nature can assist in identifying potential causes and developing resilient methodologies for data quality in the IoT. Generally, data are perceived and managed to automate desired functions, such as adjusting the temperature in a smart room or activating a dehumidifier. Incorrect sensor data in these situations can affect the actuators' functionalities (Figure 5).

A report by McKinsey[21] suggests that temperature and humidity are crucial variables affecting a significant portion of the global population. Thus, the importance of erroneous or missing humidity or temperature data cannot be overstated. Normal air humidity falls between 20% and 70%, and average room temperature lies within 18–22 °C, considering seasonal and geographical variations. Data quality issues can arise from readings outside these expected ranges, data transmission delays, or being stuck in a particular state. Various causes, including sensor and actuator malfunction, data transmission issues, and network connectivity problems, can result in faulty data. Observable signs—*pitfalls*—of DC faults include anomalies, slow response time, gaps in data, and unpredictable model behavior, among others. Furthermore, certain data faults, such as temporary ones, are transient, while others, like permanent ones, require a specific solution for seamless operation. Detailed aspects of data faults are discussed in subsequent sections.

Analyzing data faults based on the IoDFT is essential to identifying the root cause; determining whether a fault is temporary or persistent; examining the component involved—hardware, software, or network; and identifying the type of fault, whether point or contextual. Overall, the IoDFT provides a structured view of

Stage 1			
Component	***Source***	***Duration***	***Type***
Hardware	Location: exterrnal	Temporal	Collective
Software	Location: internal	Permanent	Contextual
Network	Scope: single	Haphazrd	Point
	Scope: multi		

Stage 2: Data Pitfall	
Missing	Erroneous
Stuck-at	Slow response time
Nonfunctional	Overwhelmed traffic
Spike	Out of bound
Offset	Degradation

Stage 3: Causes		
Hardware	***Software***	***Network***
Misconfiguration		
Failed actuator	Update stale	Signal interference
Defective sensor	Bug	Network congestion
Power failure		Range limitation
Battery depletion		Security breach
Physical damage		Sampling rate
		Priority inversion
		Packer forward
		Data retention

FIGURE 4. Stages of data fault analysis in the IoT systems based on the IoDFT.

the fault nature, helping to manage the complexity and variety as well as identify the potential cause of a specific fault.

COMPONENT

IoT systems are usually complex and composed of constantly evolving components, such as devices, servers, networks, and analytics software. Faults in the IoT are interconnected and multifaceted. For instance, malfunction (component → hardware) can inhibit a device's ability to process, send, or receive data. Several connectivity issues primarily arise from the network the device utilizes for Internet access. A typical scenario involves the device's inability to find a reliable and available network, such as a local access point, leading to a loss of internet access (component → network).[22] Insufficient configuration may result in constrained functionalities, such as a sudden increase in the network load (component → software).

TYPES

As authors have discussed,[23,24] the fault types can be classified into *point*, *contextual*, or *collective*. Random variables may result from defective sensors or reflect a significant short-term event of interest to the system's operators (type → point). Variables may diverge from the usual records compared to the adjacent records (type → contextual). Moreover, a set of observations may deviate from the norm of much of the data (type → collective). Figure 6 depicts samples of temperature sensor readings regarding the types. We have qualitatively analyzed raw data from Urban Observatory at Newcastle University.[25]

Point type is an outlier manifested in the norm of the data. *Contextual* faults can be identified depending on the data's nature and flow. For instance, the temperature changes during the seasons and abnormal readings are raised. Finally, the *collective* type might be identified if noticeable values are identified on the same weekly timestamp.

DURATION

Temporal faults in IoT systems are transitory, appearing sporadically due to temporary malfunctions, often recovering without inflicting long-term damage. Conversely, *permanent* faults are more critical and usually require intervention for correction. These faults are triggered by hardware or software issues and necessitate replacement or maintenance. Permanent faults may pose a more substantial impact on the functioning of an IoT system.

Haphazard or irrational faults occur randomly and are preventable, triggered by malfunctioning hardware, software bugs, or other unpredictable events. For instance, a sensor that dysfunctions intermittently may transmit erroneous readings at varying timescales.

SOURCE

In the IoT realm, the fault's location can reference either the physical location of the failed component or the position within the network or system where the fault occurred. Fault locations can be categorized as *internal* or *external*, impacting the system's lifecycle differently. Further, some faults render the system ineffective due to their manifested influence, termed "scope," while other flaws impact the system independently.

Internal faults are typically caused by hardware or software defects within the system. Conversely, external faults arise from elements outside the IoT application's framework, such as environmental factors. These faults are often easier to identify, as they may be more conspicuous and traceable. For instance, a device may fail due to extreme temperatures affecting resources such as the CPU or memory.

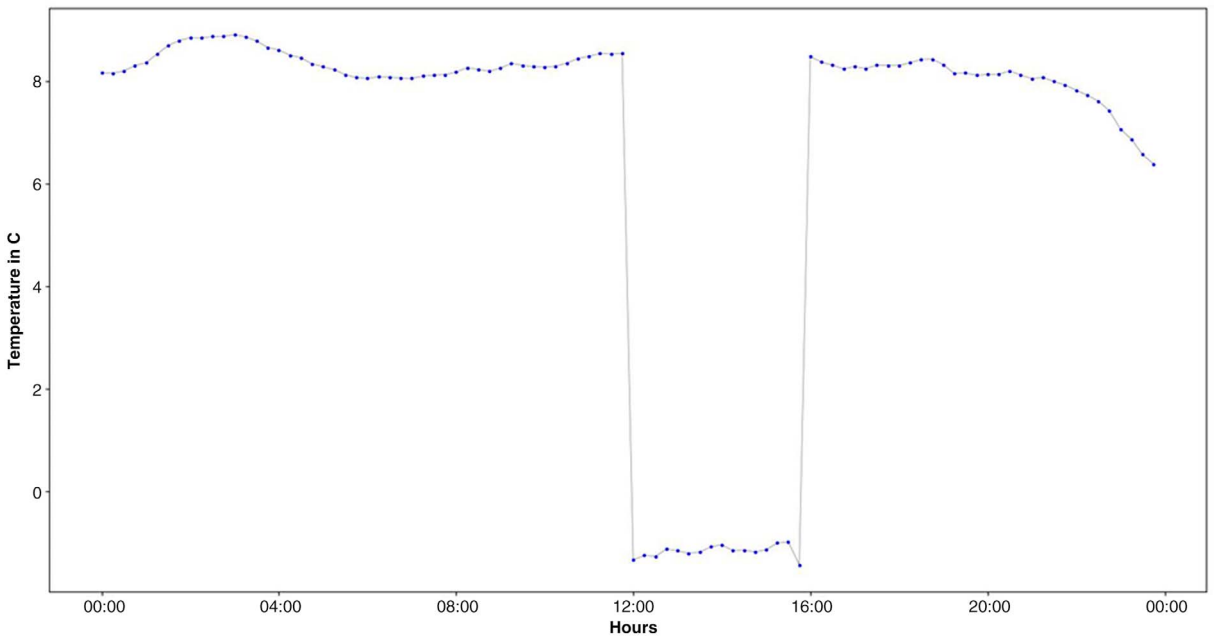

FIGURE 5. In certain circumstances and based on the context, hypothetically, on 3 April 2023, between 12 and 4 p.m. a sensor is transmitting values close to 0 °C. This deviation from the norm is unusual, especially since it is not typical to observe temperatures below 0 °C in April. This type of anomaly in data is categorized as a contextual fault type.

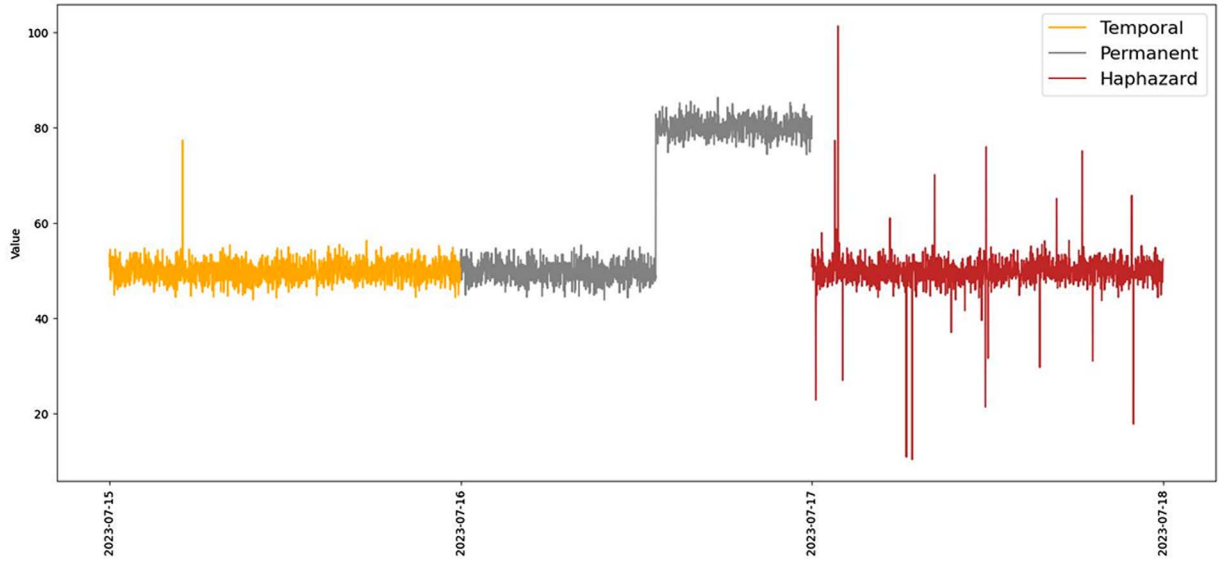

FIGURE 6. Understanding the flaws and the nature of the fault in the IoT data can be challenging, and it is helpful to consider three distinct periods to gain a deeper insight—temporal, permanent, and haphazard. As shown, the temporal state is transitional and only lasts for a short period. On the other hand, the permanent state occurs when multiple incorrect readings are transmitted due to an unknown cause. Lastly, the haphazard state occurs when random shifts between the transmitted readings occur.

Certain faults adversely affect only one component, while others—termed "cascaded"—extend their influence on another component or node within the IoT system.

The IoDFT provides a classification scheme that organizes the myriad faults that might occur in IoT systems. It is a tool for investigating various IoT faults to understand their causes, impacts, and potential solutions.

DATA PITFALLS AND THEIR CAUSES

We argue that faults in the IoT pipeline originate from three main components—*hardware*, *software*, or *network*. For instance, a *nonfunctional device* caused by *physical damage* or *battery depletion* is considered a hardware cause. On the other hand, network defects caused by an *overflow message pack* between components lead to erroneous data or missing data. Generally, as described earlier in the article, IoT systems are prone to faults during the system pipeline and design at any stage. Ultimately, some cause, such as *misconfiguration*, is not limited to being linked to the software; it might also be aligned with the hardware or network.

Given the complex interplay between indicators and causes, fault detection reliant on a single aspect may yield inaccurate results and lead to faulty decisions. Therefore, a comprehensive analysis and examination of various data sources are warranted.

Being dispersed in nature, IoT architectures are susceptible to various faults. Missing data are the most prevalent form of real-time uncertainty. Some faults are permanent, such as those caused by a damaged battery, while others are transient due to environmental changes. These intermittent occurrences may cause data to be sent erroneously. Moreover, communication networks (such as 3G, 4G, and LoRaWAN) are the most common sources of missing data, generally temporary. The LoRaWAN network has a drawback wherein, in the absence of data acknowledgment, some data might get lost in transit, leading to missing data.

Similarly, if a sensor is deployed in a remote location, hardware tends to be the prevalent cause (attributable to poor design, battery failure, memory failure, sensor damage, or environmental conditions). For instance, aggressive battery-preserving algorithms might lead to IoT devices being in sleeping mode most of the time, resulting in data loss pitfalls. Overloading in the messaging queue is another potential cause of missing data. Specific data pitfalls can be identified by considering the peculiarities of the detected data. The causes of data faults are often related to sensor performance or the retrieved data. Erroneous values in the transmitted data, possibly arising from incorrect calibration in a sensing unit, can add a deviation value to the actual detected data, causing deviation data faults. Furthermore, *stuck-at* or *spike* data pitfalls, where the volatility of a sequence of sensed data remains constant or the fluctuation of sensed data deviates from the anticipated change rate, are other potential issues.

AUTOMATED DETECTION AND HEALING

Monitoring an IoT system is essential in handling IoT data failure, ensuring rich data collection for diagnosing and future decision making. It involves observing and recording the system's operational status, data transmission, and real-time connectivity. Existing mechanisms, including IoT-QWatch,[4] could be leveraged to gather data from various IoT devices. Considering that there may be some resource constraint scenarios (i.e., edge computing), reinforcement learning—particularly deep reinforcement learning[26]—can be used to manage and optimize data acquisition from various devices intelligently. It can learn from the environment and decide the relatively better routines for which device's data need to be acquired and when optimizing resources. These data can include both the primary data points and other metrics related to the devices' performance, such as temperature, battery levels, and network signal strength, and it is necessary to preprocess the data procedurally. Processed data may need to be manually annotated with several selected samples or with the help of the deep clustering[27] methods to ensure that the acquired data can be discriminated preliminarily.

The gathered data can be used to analyze and build the intelligent data fault detection pipeline involving state-of-the-art algorithm design and model training. Due to the data being sequential and high dimension as well as containing diverse information, the deep learning model can be highly beneficial here, with the ability to learn complex patterns and identify anomalies or predict fault.[28] On the one hand, a model will be trained by supervised deep learning (e.g., long short-term memory[29] and transformers[30]), which is based on the collected data with annotation of different data faults, and the successfully trained model will be developed into the system for real-time detection. On the other hand, a detection model can be trained unsupervised, acting in the role of anomaly detection,[31] assuming the model is trained on normal (nonfaulty) data from the IoT system. Different models or learning paradigms can be explored in the future, following the two directions to detect the data faults.

After a fault has been detected, the goal is to rectify it, and automated troubleshooting or generative models are the approaches that can be employed. When a fault is detected, automated scripts can be executed to try to resolve the problem. This can range from simple solutions, like restarting the device or cleaning up its memory, to more complex solutions that might require human intervention. The optimal combination routines of executing the recovery scripts can be controlled by an intelligent reinforcement learning agent.[32] Also, generative models like generative adversarial networks,[33] variational autoencoders,[34] or diffusion models[33] can be used to rectify data faults in IoT systems due to their ability to learn the underlying distribution of data. This functionality empowers them to produce high quality and rectify the eccentricity to align standard operational parameters, which may include tasks such as imputing absent data and rectifying erroneous data.

SUMMARY AND OPEN RESEARCH QUESTIONS

This article presents the IoT data fault taxonomy as a tool for addressing data quality in IoT pipeline architectures. We have proposed and analyzed six aspects as depicted in the IoDFT. As one ventures deeper into the IoT landscape, the system inevitably becomes exposed to various faults. A key research goal is to develop effective solutions to mitigate these challenges; hence, the suggestion of leveraging ML to alleviate difficulties in the IoT. Future considerations may include designing a component capable of controlling low data quality using ML techniques and developing a DC metric to identify potential causes affecting data integrity generated from IoT devices. The complexity and integrity of systems and data in the IoT pipeline architecture present numerous open research questions. As IoT systems regularly encounter an array of faults leading to low data quality, researchers are called to devise robust and innovative resilient methodologies to unravel the complexity in the heterogeneous realm of the IoT. Creating an IoT system capable of addressing DC faults will ensure the reliability and accuracy of the system.

REFERENCES

1. M. Villari, M. Fazio, S. Dustdar, O. Rana, D. N. Jha, and R. Ranjan, "Osmosis: The osmotic computing platform for microelements in the cloud, edge, and Internet of Things," *Computer*, vol. 52, no. 8, pp. 14–26, Aug. 2019, doi: 10.1109/MC.2018.2888767.
2. M. Norris et al., "IoTRepair: Flexible fault handling in diverse IoT deployments," *ACM Trans. Internet Things*, vol. 3, no. 3, pp. 1–33, Aug. 2022, doi: 10.1145/3532194.
3. K. Fizza, P. P. Jayaraman, A. Banerjee, D. Geor-Gakopoulos, and R. Ranjan, "Age of data aware Internet of Things applications," in *Proc. IEEE 19th Annu. Consum. Commun. Netw. Conf. (CCNC)*, 2022, pp. 399–404, doi: 10.1109/CCNC49033.2022.9700640.
4. K. Fizza, P. P. Jayaraman, A. Banerjee, N. Auluck, and R. Ranjan, "IoT-QWatch: A novel framework to support the development of quality aware autonomic IoT applications," *IEEE Internet Things J.*, early access, May 2023, doi: 10.1109/JIOT.2023.3278411.
5. A. Gaddam, T. Wilkin, M. Angelova, and J. Gaddam, "Detecting sensor faults, anomalies and outliers in the Internet of Things: A survey on the challenges and solutions," *Electronics*, vol. 9, no. 3, Mar. 2020, Art. no. 511, doi: 10.3390/electronics9030511.
6. B. J. Bell, "What is machine learning?" in *Machine Learning and the City: Applications in Architecture and Urban Design*, S. Carta, Ed. Hoboken, NJ, USA: Wiley, 2022, ch. 18, pp. 409–444.
7. "Smart city market by application, component, and geography - Forecast and analysis 2023-2027." Technavio. Accessed: May 1, 2023. [Online]. Available: https://www.technavio.com/report/smart-cities-market-industry-analysis
8. M. Gillani, H. A. Niaz, and M. Tayyab, "Role of machine learning in WSN and VANETs," *Int. J. Elect. Comput. Eng. Res.*, vol. 1, no. 1, pp. 15–20, Jun. 2021, doi: 10.53375/ijecer.2021.24. [Online]. Available: https://ijecer.org/ijecer/article/view/24
9. A. Chatterjee and B. S. Ahmed, "IoT anomaly detection methods and applications: A survey," *Internet Things*, vol. 19, Aug. 2022, Art. no. 100568, doi: 10.1016/j.iot.2022.100568.
10. A. Alrajhi, K. Roy, L. Qingge, and J. Kribs, "Detection of road condition defects using multiple sensors and IoT technology: A review," *IEEE Open J. Intell. Transp. Syst.*, vol. 4, pp. 372–392, Feb. 2023, doi: 10.1109/OJITS.2023.3237480.
11. L. Xing, "Cascading failures in Internet of Things: Review and perspectives on reliability and resilience," *IEEE Internet Things J.*, vol. 8, no. 1, pp. 44–64, Jan. 2021, doi: 10.1109/JIOT.2020.3018687.
12. Y. Jeong, "Blockchain processing technique based on multiple hash chains for minimizing integrity errors of IoT data in cloud environments," *Sensors*, vol. 21, no. 14, Jul. 2021, Art. no. 4679, doi: 10.3390/s21144679.
13. T. J. OConnor, W. Enck, and B. Reaves, "Blinded and confused: Uncovering systemic flaws in device telemetry for smart-home Internet of Things," in *Proc. 12th Conf. Secur. Privacy Wireless Mobile Netw.*, 2019, pp. 140–150, doi: 10.1145/3317549.3319724.
14. V. C. Farias da Costa, L. Oliveira, and J. de Souza, "Internet of Everything (IoE) taxonomies: A survey and

a novel knowledge-based taxonomy," *Sensors*, vol. 21, no. 2, Jan. 2021, Art. no. 568, doi: 10.3390/s21020568.
15. R. C. Nickerson, U. Varshney, and J. Muntermann, "A method for taxonomy development and its application in information systems," *Eur. J. Inf. Syst.*, vol. 22, no. 3, pp. 336–359, May 2013, doi: 10.1057/ejis.2012.26.
16. P. Zahra, Z. Mani, and R. Masoud, "Data-centric approaches in the Internet of Vehicles: A systematic review on techniques, open issues, and future directions," *Int. J. Commun. Syst.*, vol. 36, no. 3, 2023, Art. no. e5383, doi: 10.1002/dac.5383.
17. M. H. Jarrahi, A. Memariani, and S. Guha, "The principles of data-centric AI (DCAI)," 2022, *arXiv:2211.14611*.
18. N. Polyzotis and M. Zaharia, "What can data-centric AI learn from data and ML engineering?" 2021, *arXiv:2112.06439*.
19. D. Zha, Z. P. Bhat, K. Lai, F. Yang, and X. Hu, "Data-centric AI: Perspectives and challenges," 2023, *arXiv:2301.04819*.
20. A. Ng. *A Chat with Andrew on MLOps: From Model-Centric to Data-Centric AI*. (2021). [Online Video]. Available: https://t.ly/6g1sh
21. McKinsey and Health Institute. "The healthy 23: Drivers of your health and longevity." McKinsey & Company. Accessed: May 1, 2023. [Online]. Available: https://www.mckinsey.com/~/media/mckinsey/mckinsey%20health%20institute/our%20insights/the%20secret%20to%20great%20health%20escaping%20the%20healthcare%20matrix/table-1-the-healthy-23-drivers-of-your-health-and-longevity_janfinal.pdf
22. A. Makhshari and A. Mesbah, "IoT bugs and development challenges," in *Proc. IEEE/ACM 43rd Int. Conf. Softw. Eng. (ICSE)*, 2021, pp. 460–472, doi: 10.1109/ICSE43902.2021.00051.
23. A. Cook, G. Mısırlı, and Z. Fan, "Anomaly detection for IoT time-series data: A survey," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 6481–6494, Jul. 2020, doi: 10.1109/JIOT.2019.2958185.
24. A. Diro, N. Chilamkurti, V. Nguyen, and W. Heyne, "A comprehensive study of anomaly detection schemes in IoT networks using machine learning algorithms," *Sensors*, vol. 21, no. 24, Dec. 2021, Art. no. 8320, doi: 10.3390/s21248320.
25. "Newcastle urban observatory," Newcastle Univ., Newcastle upon Tyne, U.K. Accessed: May 1, 2023. [Online]. Available: https://newcastle.urbanobservatory.ac.uk/
26. J. Xu, Z. Xu, and B. Shi, "Deep reinforcement learning based resource allocation strategy in cloud-edge computing system," *Frontiers Bioeng. Biotechnol.*, vol. 10, Aug. 2022, Art. no. 908056, doi: 10.3389/fbioe.2022.908056.
27. Y. Ren et al., "Deep clustering: A comprehensive survey," 2022, *arXiv:2210.04142*.
28. Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015, doi: 10.1038/nature14539.
29. S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.
30. A. Vaswani et al., "Attention is all you need," 2017, *arXiv:1706.03762*.
31. A. Blázquez-García et al., "Exploratory combinatorial optimization with reinforcement learning," *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 4, pp. 3243–3250, Apr. 2020, doi: 10.1609/aaai.v34i04.5723.
32. I. Goodfellow et al., "Generative adversarial networks," *Commun. ACM*, vol. 63, no. 11, pp. 139–144, Nov. 2020, doi: 10.1145/3422622.
33. J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, vol. 33, pp. 6840–6851.
34. J. An and S. Cho, "Variational autoencoder based anomaly detection using reconstruction probability," *Special Lecture IE*, vol. 2, no. 1, pp. 1–18, Dec. 2015.

SULTAN ALTARRAZI is a lecturer at King Abdulaziz University, Jeddah, 21589, Saudi Arabia, and a Ph.D. candidate at Newcastle University, NE1 7RU, Newcastle upon Tyne, U.K. Contact him at starazi@kau.edu.sa or s.m.m.altarrazi2@newcastle.edu.uk.

TOMASZ SZYDLO is a senior lecturer at the School of Computing, Newcastle University, NE1 7RU, Newcastle upon Tyne, U.K. Contact him at tomasz.szydlo@newcastle.ac.uk.

SCHAHRAM DUSTDAR is a full professor of computer science heading the Distributed Systems Group at the Vienna University of Technology, 1040, Vienna, Austria. Contact him at dustdar@dsg.tuwien.ac.at.

SATISH NARAYANA SRIRAMA is an associate professor at the School of Computer and Information Sciences, University of Hyderabad, Hyderabad, 500046, India, and a visiting professor and the honorary head of the Mobile and Cloud Lab at the Institute of Computer Science, University of Tartu, 50090, Tartu, Estonia. Contact him at satish.srirama@uohyd.ac.in.

RAJIV RANJAN is a chair professor of computing science and the Internet of Things at Newcastle University, NE1 7RU, Newcastle upon Tyne, U.K. Contact him at raj.ranjan@ncl.ac.uk.