Self-Adaptation in IoT Systems for Smart Cities

Bruno C. Nogueira, Rebeca C. Motta, Flávia C. Delicato
Institute of Computing - UFF
Niterói, Brazil
brunocn@id.uff.br, becacmotta@gmail.com, fdelicato@gmail.com

Thais V. Batista
Informatics and Applied Mathematics Dept. - UFRN
Natal, Brazil
thaisbatista@gmail.com

***Abstract* - Smart Cities (SC) leverage digital technologies to enhance the effectiveness of urban services and improve the quality of life. Since it is a recent field of development, several challenges are afoot. An important part of this challenge concerns adaptive behavior, which is paramount to SC solutions, to respond to contextual changes and carry out tasks with minimal human interaction. Given this relevance, we performed a literature review to characterize the platforms, application types, and adaptation aspects for SC and discuss the challenges to building self-adaptive systems SC. The reviews resulted in 23 papers analyzed, from which the findings of the field's research potential, with a particular emphasis on developing SC solutions and the difficulties that must be resolved before they can be built.**

Keywords - Smart Cities, Internet of Things, Self-Adaptation, Adaptive behavior

I. INTRODUCTION

Population growth in urban areas brings great challenges to urbanization, such as traffic congestion, air pollution, and inadequate infrastructure. In this context, technology can improve citizens' quality of life, facilitate city administration and ensure that the resources and services offered by the city are used efficiently and sustainably [1]. The concept of Smart Cities (SC) comes as an alternative, focused on using digital technologies to improve the quality and efficiency of urban services like transportation, energy, healthcare, and traffic management [2]. Simultaneously, it strives to lower the overall cost of delivering these services.

SC concepts have gained popularity among administrative authorities and researchers worldwide. With growing urban populations and the arising challenges, there is a pressing need for intelligent solutions to improve citizens' lives, enhance service delivery, and address disaster mitigation [1]. However, integrating heterogeneous data and services to produce value-added information and actionable insights is challenging. As an enabler for new technical solutions, the Internet of Things (IoT) is envisioned to integrate the physical world into software-based systems [4]. IoT supports a scenario where interconnected everyday objects can behave as autonomous entities and cooperate to reach some common goal with minimum human intervention. Such smart things must be able to sense the environment, analyze the collected information, take decisions and actions to achieve their goals [3]. The IoT [5] has been widely explored in platforms for SC [6] as they form a highly heterogeneous ecosystem through which the various application domains can communicate to ensure the data collected by the sensors bring benefits to the city.

Both the IoT and SC are ecosystems characterized by high degrees of heterogeneity and dynamics. Dynamic contextual changes influence the SC systems behavior, which must be able to adapt to different situations, e.g., altering traffic lights during rush hour [7]. These features have gained notoriety in recent years concerning the system's ability to self-adapt to uncertainties at runtime based on the context in which they are inserted. Considering the perspective of pervasive computing [8], where various types of sensors and actuators are interconnected and can share their data between platforms, the connected objects should also have the ability to learn and think about both physical, virtual and social worlds by themselves [9]. Therefore, to perform actions with minimum human intervention, the self-adaptive approach becomes a requirement to guarantee this vision.

As initially proposed, self-adaptation is a behavior that should be implemented to allow software systems to autonomously adapt to changes in the environment, ensuring the maintenance of the required Quality of Service (QoS) [10]. These systems can detect environmental changes and respond appropriately, all while ensuring that user goals and QoS are met. There are already some initiatives to deal with self-adaptation behavior, such as the feedback loop that collects data about the system it is part of and the environment it is monitoring [11]. Another solution is reactive behavior, typically implemented to deal with unforeseen events [7]. Another form of adaptation is done through the analysis of periodic events since they follow a pattern in the city due to the repetitive interaction of its entities [12]. Another application of adaptation is self-configuration, which allows distributed systems to adapt to dynamic environments by automatically initializing and reconfiguring themselves [3].

Considering the importance of adaptation and the many opportunities for SC [13], this paper aims at contributing to the research on IoT systems able of autonomically adapt to environmental changes in the context of SC. In this direction, motivated by the relevance of the topic and the lack of surveys, we systematically mapped the literature to gather information about platforms that support adaptive IoT systems for SC and their applications. Our review aims to answer the following question: **What are the challenges to build adaptive IoT systems for SC?** To answer this question, we investigated platforms (leading to how the systems are built), applications (leading to which systems are built), and adaptation aspects (what to adapt). Therefore, we broke down the main question into the following:

RQ1. What are the platforms used for Adaptive IoT systems? The goal is answering: RQ1.a Were existing platforms designed to adapt? RQ1.b What are their components/functionalities? RQ1.c What technologies are used for adaptation?

RQ2. What type of SC applications are supported?

RQ3. What aspects of adaptation are addressed?

The goal is to discuss features of each platform and learn how it deals with adaptive behavior in SC.

II. LITERATURE REVIEW

To perform this review we designed a protocol following the guidelines established in [14]. In favor of transparency and to share the full extension of the review, the protocol is available online as a replication package[1]. Per the guidelines, our first step before undertaking the review was to verify the need for this study. In our initial searches, we did not find any secondary studies specific on adaptive IoT systems for SC. To address this gap, our research goal is to analyze SC with the purpose of characterizing the challenges to building adaptive IoT systems for SC from the point of view of software engineering researchers in the context of the technical literature. The steps of the review are divided into planning, execution and results.

Planning. In this phase we created the study protocol, encompassing the study objectives, search criteria, selection procedure, and extraction form. Considering the defined RQs, we aim to select research-based peer-reviewed studies that discuss the challenges to build adaptive IoT systems for SC. As secondary questions, we identify platforms, technologies and tools as enablers to implement adaptive behavior in SC. In Applications, we consider the domains and use cases (concrete examples) of adaptive behavior in SC. Aspect concerns what part of the IoT system is adapted (layer, protocol, architecture). For challenges we aim to identify open issues and research gaps.

Execution. In this step, four researchers acted as reviewers and conducted the review between May and July of 2023. As a database, the search was performed in Scopus[2] since it indexes several peer-reviewed databases and provides a well-known balance for coverage and relevance. After defining the search string, the search resulted in 754 articles. After the screening process of Title (584 papers removed) and Abstract (116 papers removed), 54 papers remained for a full reading. The final set comprises 23 papers from which we extracted relevant information and provided the basis for our findings. All details can be analyzed in our protocol.

Results. The selected papers range from 2014 to 2022 for publication year, with 43% journal articles and 57% conference papers. Figure 1 presents an overview of the papers by Publisher, which can lead us to the main events and communities interested in the SC topic. Springer was the Publisher with the higher number of selected papers.

Figure 2 depicts the most cited technologies. Multi-agent systems are the most frequently used. It is understandable since it is hard to anticipate every scenario and designate the system's behavior in advance. Therefore, agents are widely used since they can learn from and adapt to their environment. Cloud/Edge/Fog computing, Simulation, and ML appear as enablers for adaptation. Osmotic Computing is frequently associated with Edge Computing and Agents, gaining notoriety for its abstraction purposes. MAPE-K is one of the most used models for self-adaptive systems.

III. ANSWERS TO RESEARCH QUESTIONS

A qualitative analysis was conducted on the extracted data to identify patterns, trends, similarities, and differences. The primary findings are summarized as follows.

Fig. 1. Number of Papers by Publisher.

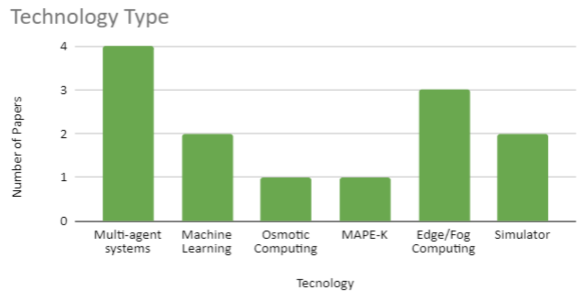

Fig. 2. Mapping of Technology Types.

A. *RQ1. What are the platforms used for Adaptive IoT systems?*

For this question, the sub-questions explore whether the platform was designed for adaptability, its components and functionalities, and which technologies were used in the adaptation. The selected papers present various frameworks and platforms for developing smart environments and applications in the context of adaptive IoT systems. These frameworks aim to enhance IoT devices intelligence, adaptability, and scalability, allowing them to respond to environmental changes and user needs efficiently.

Among the 23 articles selected, only two [9] and [15] deal with platforms exclusively designed for adaptive IoT systems in SC. The Rainbow platform [9] adopts a distributed layer of agents over the physical layer of the IoT, addressing heterogeneity and complexity. It includes adaptive, decentralized algorithms for large-scale cyber-physical applications such as SC. It uses emergent behaviors resulting from agent interactions, increasing adaptation and self-configuration capabilities. Article [15] proposes Sapparchi, a platform for running applications at multiple computational levels (Edge, Fog, and Cloud) typically present in the SC environment. It focuses on scalability by distributing the workload across computational tiers using osmotic computing to equalize workflows. A point that draws attention in both works is the use of Edge/Fog computing technology in their architecture, enabling higher platform scalability and lower bandwidth consumption.

Two articles [3] and [7] present versatile platforms that can be used in several smart domains. iSapiens [3] is a Java-based, agent-oriented system that models intelligent

[1] The replication package is available online: http://bit.ly/SIoT
[2] Scopus has the broadest coverage of interdisciplinary citation databases, making the odds of missing key publications reduced. More information: https://www.scopus.com

objects as agents in a multi-agent system. Agents collaborate to achieve specific goals, providing reasoning capabilities and intelligence to IoT devices. This approach leverages edge computing, processing data closer to devices, reducing costs, and improving efficiency. The SitOPT platform [7] provides means to identify, process, and react to situations captured by heterogeneous sensors. Its three-tier architecture uses sensors and situational awareness mechanisms to model and detect relevant situations and uses adaptive workflows as a reactive mechanism to such situations.

Articles [16] and [17] use the DeltaIoT simulator to develop their adaptive systems projects in order to model SC applications. In [16] DingNet, a simulator for self-adaptation research that uses a physical IoT configuration to model SC applications, is created. Applications connect with geographically distributed gateways that can interact through a wireless network with motes equipped with sensors and actuators. In [17], the focus was on exploring the DeltaloT.v1 and DeltaloT.v2 versions to face the exhaustive analysis of large adaptation spaces using online machine learning. The approach enhances the traditional MAPE-K feedback loop (Monitor- Analyzer- Planner - Executor - Knowledge) with a learning module that supports the analyzer in selecting relevant adaptation options.

One article [18] presented a framework with great potential for adaptive systems. SLASH (Self-Learning and Adaptive Smart Home) is a system for smart homes. It leverages ML to enhance the intelligence of sensors, enabling them to autonomously detect situations, learn the inhabitant's behavioral patterns, respond and control household functionalities with minimal human intervention.

B. RQ2. What type of SC applications are supported?

Adaptive IoT applications are in the core of SC systems, as they provide resilient and efficient solutions for several domains. These solutions benefit the population, the government, and the environment, offering smarter decision-making and technologies to improve life in cities. The application of adaptive technologies in the IoT scenario is essential to meet modern cities' complex and ever-changing demands, making them more sustainable, efficient, and friendly to their inhabitants. For this RQ, from the 23 articles analyzed, only nine describe applications in adaptive IoT systems for SC.

Giallonardo *et al.* [19] proposes an architecture based on semantics and ontology for self-adaptive reactive systems. A case study is carried out to design an intelligent building system for preserving cultural heritage. Several sensors/actuators, such as infrared, presence, thermal cameras, and on/off sensors, are used. Resilience is ensured by reconfiguring the model when sensors fail, replacing them with equivalent logic sensors, which combine observations from other sensors to maintain the original functions.

The Rainbow framework [9] describes three SC applications: mapping noise pollution, controlling urban drainage networks to reduce environmental impacts in heavy rains, and monitoring air quality. SitOPT [7] presents applications that react to events captured by heterogeneous sensors. In a server room, a temperature sensor can trigger actions such as notifying the administrator or starting climate control. The events could be handled by human intervention or with context-adaptive workflows.

Medvedev *et al.* [20] describes an architecture to store and index context in SC applications. The example is solid waste management, where sensors in dumps provide information to optimize collection routes, reducing fuel consumption and improving service quality. Article [21] presents BPMN4SAS, an extension of BPMN (Business Process Modeling and Notation) to manage adaptation from the modeling phase, with pre-defined criteria to optimize performance. It presents an example of an accident manager using BPMN4SAS to model activities with their respective QoS restrictions, such as performance, availability, security, etc. It supports adaptive and efficient management of emergencies, making the city safer and more resilient in the face of unexpected events. A case study for an IoT network dedicated to online games is proposed in [23]. Such a network requires an intelligent and adaptable environment, in order to enable the user to form a direct contract with the gaming company, thus not having to choose a communication provider. The network must support sensors to provide a realistic gaming experience and autonomous management based on stakeholder policies and user context.

For Sapparchi [15], a case study is implemented for an intelligent parking application that maps available spaces in the city in real time. As for SLASH [18], the case presented is for a smart home; with machine learning, sensors can react and control the house autonomously. The system learns from the inhabitant's behavior and supports their needs.

Paper [24] presents a proposal for an adaptive data rate (ADR) mechanism for LoRaWAN networks, in the context of Industrial Internet of Things (IIoT).

C. RQ3. What aspects of adaptation are addressed?

Six articles are examples of truly adaptive systems in which the system architecture has its structure and/or behavior dynamically changed. We also observed several instances of adaptation at the communication level. Four articles deal with the application's adaptive behavior.

Agent technologies allow building truly self-adaptive systems. However, some proposals based on agents only implement adaptation at the application level. In [9], agents use technologies to control a city's drainage system. Agents monitor the water load and, through actuators, control the gates depending on the context, including unpredictable events such as heavy rainfall, to avoid flooding. It is an example in which the system is not adaptive but only the actions performed in the environment, thus it is a typical **adaptive application**, but instead of using predefined rules to guide the adaptive behavior, autonomous agents infer, and act according to the context. In [22], IoT-based intelligent environments are envisioned as systems where devices, considered as agents, act autonomously and dynamically according to a collective behavior that facilitates the user's life. The proposal builds on spontaneous configuration by setting levels of light and heat, depending on various conditions. Regarding **what to adapt**, the proposal adapts the environment according to the user's goals and current state. Again, it is an **adaptive application**, not a system. In [3], smart objects are modeled as agents, running on a multi-agent system, and cooperate to achieve specific goals.

Agents operate according to an analyze-decide-act cognitive cycle, similar to the classic feedback loop of the Autonomic Computing paradigm. Both physical and virtual systems aspects can be dynamically adapted thus, regarding what to adapt, it is the whole system's structure and behavior.

The proposal in [26] is aligned with the vision of autonomic computing and deals with a challenging issue, which is to decentralize the coordination in the execution phase of the adaptation. As for **what to adapt**, with the support of the adaptation manager, **any behavior** of the application can be dynamically changed. Paper [10] focuses on the self-healing property of autonomic systems, which denotes the system's ability to automatically detect, diagnose, and repair software and hardware issues. They introduce the WoT component agent, responsible for executing an operation in an application, communicating with peers, and applying fault-tolerance mechanisms. A WoT agent may retry or replace its operation, retry the communication with another WoT agent, or replace a WoT agent in the WoT application. The replacement mechanism is an example of both behavioral and structural adaptation.

A solid waste management system with on-demand adaptive garbage collection from IoT-enabled bins is illustrated in [20]. This is another example where adaptation does not refer to the system and its components, but to **application actions** (the behavioral logic). The framework proposed in [18] supports the design and implementation of smart home systems endowed with adaptive and self-learning capabilities. It provides management functions to controlling behaviors already foreseen in an automated house, executing actions defined by the user. The system also controls the execution of actions that must be executed upon the occurrence of events previously defined. It also manages automated actions resulting from the perception of situations that the user does not explicitly define. The system cannot be considered strictly adaptive since it does not modify its structure or behavior. However, when using the framework, the system is endowed with self-learning.

In [11], the authors illustrate uncertainties to which IoT systems are subject to and how they hinder efficient operation, demanding reconfigurations that ideally should be done automatically. They claim that self-management in IoT must be guided by high-level adaptation goals, such as reducing system energy consumption. Actions must be performed automatically, considering the environment uncertainties. At the Things layer, uncertainties include inaccuracies inherent in sensor data readings and device failures. Since communication is an energy-intensive operation, the network must be carefully configured to optimize its use and increase the system's lifetime. In this sense, **what to adapt** can be illustrated by the choice of routing protocol or the routing in a multi-hop configuration. This is an example of adapting the system itself, whether at the behavioral (configuration of routes) or structural (changing what implements the routing protocol) level.

In [16], mobile motes dynamically adapt their communication settings to ensure reliable and energy-efficient communication. **What to adapt** concerns the configuration of the communication, which encompasses many possibilities such as the devices' transmission power and sampling rate (parameter adaptation), the destination gateway to which sending the sensed data or the adopted communication protocol (structural adaptation).

Paper [25] focuses on adapting network-level protocols. Motivated by the typical heterogeneity in IoT, it uses autonomic principles to facilitate the integration of heterogeneous devices with M2M gateways. They allow the network protocol stack's dynamic configuration according to the services' requirements. Details of low-level network technologies and protocols do not need to be exposed to higher levels. They achieve this goal by using semantic descriptions (ontologies). Article [28] proposes an adaptive routing protocol based on Reinforcement Learning (RL). It has the ability to detect the level of mobility at different points of time so that each individual node can update routing metrics. Article [24] suggests DROB (downlink rate optimization for class B), an adaptive data rate mechanism to enhance the operation for downlink and class B devices. It helps adapt to channel conditions by changing devices' data rate.

An innovative and relevant approach for SC is described in [12]. Since a large number of software services may degrade performance of service discovery, it proposes a self-adaptive service model to support discovery. The model adapts the organization of service information according to city events. A self-adaptive architecture keeps track of discovery metrics and moves information about services between registries to keep discovery efficiency. **What to adapt** refers to metadata (data describing services). Paper [27] proposes using semantically annotated resources via a semantic interworking proxy to dynamically discover new kinds of information and automatically translate data between a given source and target IoT platforms (at runtime). Therefore, the aspect of **what to adapt** refers to the data format.

In [15], applications run as a graph of MicroElements (MELs) that can migrate between the Cloud/Edge dynamically following QoS parameters. They build on the concept of Osmotic computing to promote the scalability of SC applications. **What to adapt** refers to the physical tier where application components run. Article [21] proposes an IoT adaptive system that can repair itself if any execution problems occur and complete its execution while meeting QoS requirements. Using a BPMN extension it addresses adaptation in the design phase. It also addressed the optimization phase by proposing an architecture to ensure self-adaptation at run time. Components collect data on uncertainties during execution and adjust to changing conditions, meeting QoS constraints. If a working service is about to fail or its QoS degrades, the system re-compose it with the proper replacement service. It is a classic example of structural adaptation. In [23], the focus of the adaptation is to adapt the system behavior, in terms of data processing and its location, to the different stakeholders and their context, including physical conditions such as heart rate. Adaptation strategies encompass updating the system knowledge base with new data, new subscriptions that dynamically become available, and the respective components and protocols for dealing with such data.

IV. CHALLENGES

What are the challenges to build adaptive IoT systems for SC? Considering the papers analyzed and the answers to the RQs, we organized the findings to give an overview and highlight the challenges. Despite the availability of several platforms to build adaptive SC systems, as discussed in RQ1, some challenges remain, such as (i) the need to infer high-level contextual aspects and turn them into measurable and actionable ones to handle changes properly, and (ii) the need to ensure system reliability to support reactiveness, mainly concerning the solution scale.

Regarding (i), finding a suitable abstraction level and the proper mechanisms to enable adaptation is often related to the aspects reported in RQ3. To address how to adapt, a platform may be integrated into another, they may interact at the same level, or a common integration level may be built on top, depending on the SC and its implemented services. For this reason, **interoperability** is a challenge since the degree of compatibility and homogeneity that may be attained determines how valuable platform connections are [27]. Another challenge related to *how* concerns **Technology Maturity** [18]. As reported in RQ2, some solutions are still in the pilot phase or are proof of concepts and need a level of maturity, which limits the generalization and the conclusion of how to be used in other applications.

Regarding (ii), **reliability** is affected by the uncertainties that can happen with SC applications. For example, network interference is caused by outside elements like weather and traffic and changes in the data traffic generated by devices that can enter and exit the network at any time [16]. **Scalability** is still an issue [12], for example, because most self-adaptive approaches either advocate for completely distributed architectures with difficult service management or centralized designs with single points of failure, both of which have their impact in the case of SC.

These challenges provide opportunities and gaps for research and development to advance in SC solutions and provide adequate strategies for handling the complexity of settings that are constantly changing.

V. CONCLUSION

The findings presented in this paper shed light on the current state of the art of adaptive IoT systems for SC, identifying platforms designed for adaptation, types of applications for SC, from environmental monitoring to urban infrastructure optimization, as well as analyzing the main aspects of adaptation addressed in such systems. Findings revealed that while some aspects of IoT systems are dynamically adapted, challenges such as inferring high-level contextual aspects, ensuring interoperability, and dealing with system reliability and scalability present promising avenues for future research and development to advance the field of adaptive IoT systems for SC.

Acknowledgements. This work was partially supported by funding agencies FAPERJ (grant number 200.972/2022) and CNPq. Flavia Delicato and Thais Batista are CNPq Fellows.

REFERENCES

[1] N. Nezamoddini et al. "A Survey of Adaptive Multi-Agent Networks and Their Applications in Smart Cities," vol. 5, no. 1, 2022.

[2] S. S. Samant, et al."Towards Quality-Assured Data Delivery in Cloud-Based IoT Platforms for Smart Cities," in IEEE 3rd Int. Conf. on Collaboration and Internet Computing, Oct. 2017, pp. 291–298.

[3] F. Cicirelli et al. "A Scalable Agent-Based Smart Environment for Edge-Based Urban IoT Systems," in Interoperability, Safety and Security in IoT, in Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol. 242. Cham: Springer, 2018, pp. 53–59.

[4] Y. Dong, et al. "A Feedback-Based Adaptive Service-Oriented Paradigm for the Internet of Things," in Service-Oriented Computing – ICSOC2017 Workshops, LNCS10797. 2018, pp. 137–148

[5] A. Al-Fuqaha et al. "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," IEEE Communications Surveys & Tutorials, vol. 17, no. 4, pp. 2347–2376, 2015

[6] A. H. Alavi, et al. "Internet of Things-enabled smart cities: State-of-the-art and future trends," Measurement, vol. 129, pp. 589–606, Dec. 2018

[7] P. Hirmer et al., "Situation recognition and handling based on executing situation templates and situation-aware workflows," Computing, vol. 99, no. 2, pp. 163–181, Feb. 2017

[8] J. Gubbi et al., "Internet of Things (IoT): A vision, architectural elements, and future directions," Future Generation Computer Systems, vol. 29, no. 7, pp. 1645–1660, Sep. 2013

[9] A. Giordano et a.i, "Smart Agents and Fog Computing for Smart City Applications," Jun. 2016, pp. 137–146.

[10] R. Angarita, et al. "An Agent Architecture to Enable Self-healing and Context-aware Web of Things Applications:," in Proc. of the Int. Conf. on Internet of Things and Big Data, Rome, 2016, pp. 82–87.

[11] D. Weyns et al., "Self-managing Internet of Things," in SOFSEM 2018: Theory and Practice of Computer Science, A. M. Tjoa, et al., Eds., in LNCS vol. 10706. Cham: Springer, 2018, pp. 67–84.

[12] C. Cabrera et al., "A Self-Adaptive Service Discovery Model for Smart Cities," IEEE Trans. on Services Computing, vol. 15, no. 1, pp. 386–399, 2022

[13] R. C. Motta et al., "Making Cities Smarter: A Literature Review of Characteristics, Technologies, and Challenges," in Anais do Simpósio Brasileiro de Computação Ubíqua e Pervasiva, Aug. 2023.

[14] D. Budgen et al. "Performing systematic literature reviews in software engineering," in Proc. of the 28th Int. Conf. on Software engineering, Shanghai China: ACM, May 2006, pp. 1051–1052.

[15] A. Souza, N. et a., "SAPPARCHI: an Osmotic Platform to Execute Scalable Applications on Smart City Environments," in IEEE 15th Int. Conf. on Cloud Computing (CLOUD), Jul. 2022, pp. 289–298.

[16] M. Provoost et al. "DingNet: A Self-Adaptive Internet-of-Things Exemplar," in 2019 IEEE/ACM 14th Int. Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS), Montreal, CA, May 2019, pp. 195–201

[17] F. Quin, et al. "Efficient Analysis of Large Adaptation Spaces in Self-Adaptive Systems using Machine Learning," in 2019 IEEE/ACM SEAMS, May 2019, pp. 1–12.

[18] M. Sultan and K. N. Ahmed, "SLASH: Self-learning and adaptive smart home framework by integrating IoT with big data analytics," in 2017 Computing Conference, Jul. 2017, pp. 530–538.

[19] E. Giallonardo et al., "Semantics-Driven Programming of Self-Adaptive Reactive Systems," Int. J. Soft. Eng. Knowl. Eng., vol. 30, no. 06, pp. 805–834, Jun. 2020,

[20] A. Medvedev et al., "Storing and Indexing IoT Context for Smart City Applications," Sep. 2016.

[21] F. Z. Merabet and D. Benmerzoug, "Service Modeling and Architecting for Self-Adaptive IoT-Based Systems," Mar. 2022.

[22] P. Rust, et al. "Resilient Distributed Constraint Reasoning to Autonomously Configure and Adapt IoT Environments," ACM Trans. Internet Technol., vol. 22, no. 4, p. 100:1-100:31, Nov. 2022

[23] S. P. Singh, et al. , "Smart and Adaptive Architecture for a Dedicated Internet of Things Network Comprised of Diverse Entities: A Proposal and Evaluation," Sensors, vol. 22, no. 8, Jan. 2022.

[24] D. Todoli-Ferrandis, et al. "Robust Downlink Mechanism for Industrial Internet of Things Using LoRaWAN Networks," Electronics, vol. 10, no. 17, Art. no. 17, Jan. 2021,

[25] E. Gazis, et al., "On the Role of Semantic Descriptions for Adaptable Protocol Stacks in the Internet of Things," May 2014, pp. 437–443.

[26] M. WeiBbach, et al., "Decentrally Coordinated Execution of Adaptations in Distributed Self-Adaptive Software Systems," in 2017 IEEE 11th Int. Conf. on Self-Adaptive and Self-Organizing Systems (SASO), Tucson, AZ, USA: IEEE, Sep. 2017, pp. 111–120.

[27] J. An et al., "Toward Global IoT-Enabled Smart Cities Interworking Using Adaptive Semantic Adapter," IEEE Internet of Things Journal, vol. 6, no. 3, pp. 5753–5765, Jun. 2019

[28] A. Serhani, et al., "AQ-Routing: mobility-, stability-aware adaptive routing protocol for data routing in MANET–IoT systems," Cluster Comput, vol. 23, no. 1, pp. 13–27, Mar. 2020