AI-Enabled Life Cycle Automation of Smart Infrastructures

Carolina Fortuna, Halil Yetgin, Miha Mohorcic
*Jožef Stefan Institute, Ljubljana, Slovenia
{carolina.fortuna, halil.yetgin, miha.mohorcic}@ijs.si

***Abstract*—Deployment and maintenance of *large* smart infrastructures used for powering data-driven decision making, regardless of retrofitted or newly deployed infrastructures, still lack automation and mostly rely on extensive manual effort. In this paper, we focus on the two main challenges in the life cycle of smart infrastructures: *deployment* and *operation*, each of which is rather generic and apply to all infrastructures. We discuss the existing technologies designed to help improve and automate deployment and operation for for smart infrastructures in general and use smart grid as a guiding example to ground some examples across the paper. Next, we identify and discuss opportunities where the broad field of artificial intelligence (AI) can help further improve and automate the life cycle of smart infrastructures to eventually improve their reliability and drive down their deployment and operation costs. Finally, based on the usage of AI for web and social networks as well as our previous experience in AI for networks and cyber-physical systems, we provide decision guidelines for the adoption of AI.**

***Index Terms*—smart infrastructure, artificial intelligence, deployment automation, operation automation.**

I. Introduction

Cheaper and more reliable computing devices and improved connectivity solutions are enabling near real-time remote monitoring and control of points of interest, such as the climate in a smart home or the progress of production on individual factory floors. The artificial intelligence (AI)-driven industrial revolution [1], fed by machine-generated data flowing through the global communications infrastructure, is advancing process automation and data-driven decision making to increase production efficiency and overall quality of life. In addition, AI-powered automation is also being considered for core communication networks, sometimes referred to as intent-based networking [2].

A report commissioned by the Committee on Artificial Intelligence in a Digital Age [3] on ”how AI applications can be used in urban mobility and smart cities and how their deployment can be facilitated” found that optimised deployment, operation, and maintenance of infrastructure, including waste and water management, transportation, energy grids, and urban lighting, are key enablers for smart city implementation. For example, smart city lighting is considered a solution for developing a smart city by equipping lampposts with IoT devices to collect and analyse data gleaned from traffic, pedestrians and environmental factors to improve quality of life. However, the report [3] also suggests that there are many technological challenges to overcome, especially in terms of deployment and operation. Furthermore, a recent study related to adopting AI in operations research [4] highlights that ”a multidisciplinary approach to AI design and evaluation is recommended” as also noted in [5]. To explicitly illustrate the challenges of the life cycle of smart infrastructures, and present how they can be automated, we consider smart grid as a guiding example of such infrastructure throughout the paper since they; (i) represent a very mature sector requiring retrofitting with heterogeneous smart devices; (ii) are by nature very large and extremely distributed, and (iii) require continuous monitoring for the provision of very high reliability of operation in spite of vulnerability to different external effects.

Figure 1a depicts in the Energy Infrastructure with the traditional and green centralized generation, transmission, distribution and the traditional consumers that are now becoming prosumers, thus also able to push energy obtained from their small distributed generation capabilities to the grid. To support this process towards improved sustainability, the traditional infrastructure is already being extended and enhanced also with new information and communication technology (ICT) devices for improved observability and management represented in Figure 1b. The deployment of the ICT inevitably brings significant additional costs in terms of:

Challenge-1: Deployment

The deployment of smart devices tends to be human labour intensive. Each device needs to be manually picked, configured, physically positioned and then provisioned as symbolically depicted in Figure 1b. When large numbers of devices are deployed, the required time and costs may become significant. For example, according to a recent whitepaper from a device manufacturer, provisioning ten thousand smart light bulbs in a factory can take nearly 2 years before they can actually commence data streaming [6]. According to another manufacturer, the deployment represents 30% of the costs of a smart metering project [7]. Both examples gleaned from the state-of-the-art expose the excessive costs and delays incurred by massive deployments of smart devices. In Section III we discuss traditional approaches, the current state of the art and identify further AI-aided automation that could enable even more simple and rapid deployment, *initial configuration and out-of-the-box operational provisioning*.

Challenge-2: Operation

One of the most fundamental operational challenges of smart devices is to ensure scheduled maintenance as recommended by the manufacturer. For example, as the lifespan of

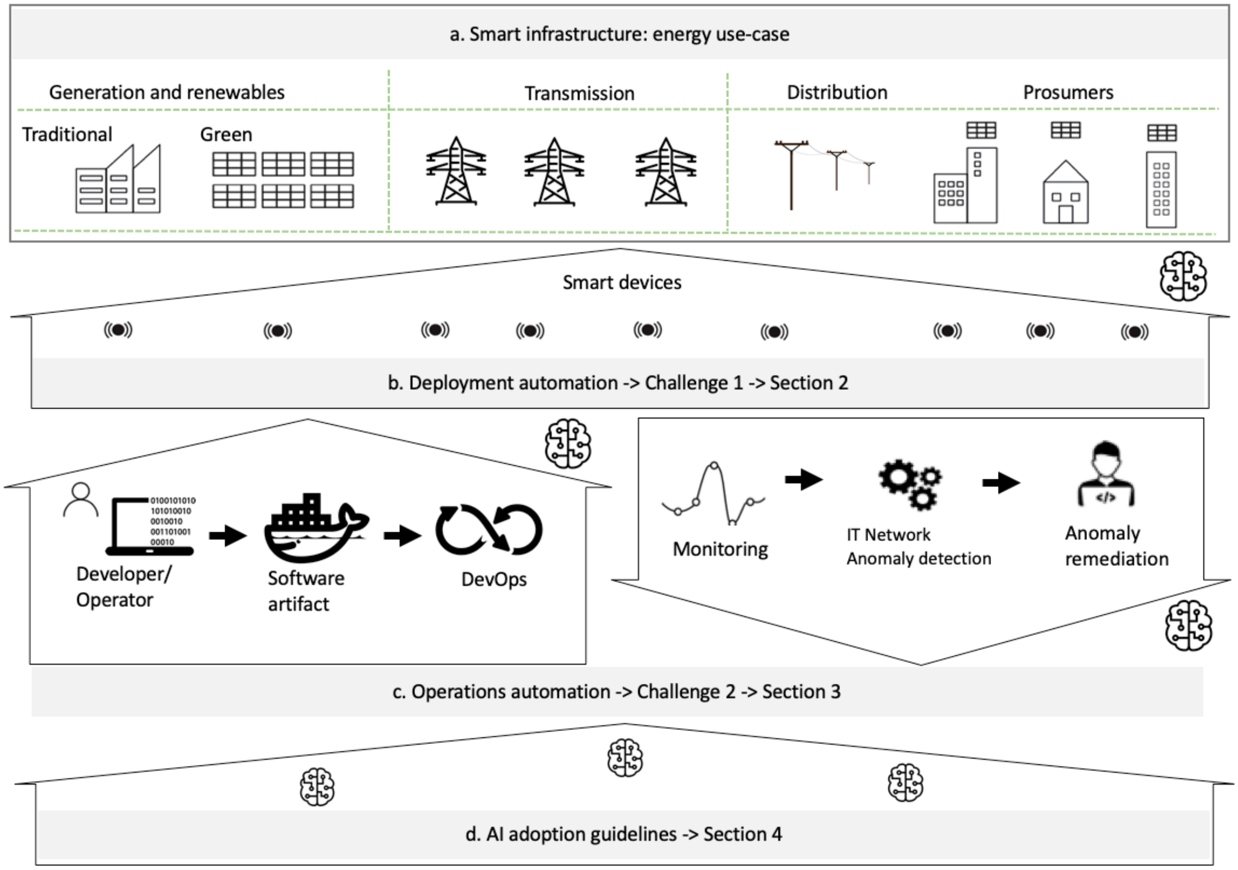

a. Smart infrastructure: energy use-case
Generation and renewables
Traditional
Green
Transmission
Distribution
Prosumers
Smart devices
b. Deployment automation -> Challenge 1 -> Section 2
0100101010
101010010
0010010
001101001
00010
Developer/
Operator
Software
artifact
DevOps
Monitoring
IT Network
Anomaly detection
</>
Anomaly
remediation
c. Operations automation -> Challenge 2 -> Section 3
d. AI adoption guidelines -> Section 4

Fig. 1: Overview of emerging smart infrastructures on an energy use-case.

a smart meter is estimated to be 10+ years, most vendors and businesses expect no associated regular physical maintenance [7]. However, over such a long lifespan, various contextual aspects in which the the smart device operates might change. For instance, the mobile operator might bankrupt, a cheaper operator may become available for the served area or simply, the network coverage may vary depending on the changes in network planning. Furthermore, previously undetected malfunctioning in the embedded smart meter software may arise, old versions of software may require upgrades to keep up with the emerging sophisticated technologies or new regulations may require software updates.

From the perspective of the long term operation of the network, we focus on two sub-challenges that i) have been subject to major progress in the last few years, ii) are specific to control centres [8] and iii) are related to monitoring and control operations as follows a) *ensuring remote network reconfiguration and software updates* as depicted on the left of Figure 1c and b) *ensuring automated detection of anomalous system behaviours* occurring during the lifespan of devices as depicted on the right of Figure 1c. In Sections IV, we discuss traditional approaches, the current state of the art and identify further AI-aided automation that could enable even more efficient remote updates and monitoring. As identified in [9] there exist other important infrastructure challenges such as security [10] and sustainability [11], however these are beyond the scope of this already broad paper.

In this paper, commencing with the two aforementioned challenges that are faced by all stakeholders involved in greenfield deployments or brownfield retrofitting, we discuss existing technologies designed to help improve and automate some aspects related to the two challenges. Increased automation for smart devices involves recent advances in the areas of network virtualization, cloud computing and AI, in addition to expertise in industrial electronics, embedded devices, wireless communications and targeted application domain [5], thus the findings of this paper are relevant to a broad community. The main contributions of this paper are as follows:

1) We systematically analyse the two main phases in the life cycle of smart infrastructures that concern all involved stakeholders and discuss future directions enabled by AI.
   - With respect to *deployment*, we analyse the current practices and show the benefit of new emerging zero-touch provisioning methods and identify further promising automation through human friendly AI-powered voice assistants in Section III.
   - With respect to *operation*, we first analyze solutions for development and operation (DevOps) automation

and identify further promising automation using data-driven AI models and knowledge-driven AI techniques in Section IV-A.
- As part of *operation*, we also analyse the current practices in fault and anomaly detection and show the benefit of more expressive approaches in SectionIV-B2.

2) We provide practical guidelines that can be used to employ AI technologies in solving automation problems as symbolically depicted in Figure1d and sicussed in Section V.

The paper is structured as follows. Section II provides the related work, Sections III and IV provide an analysis of existing and emerging automation solutions while also identifying the possible role of AI for increased automation for the identified challenges. Section V provides guidelines on when to employ artificial intelligence technologies for realizing automation. Finally, Section VI concludes the paper.

II. Related work

Considering the two challenges identified in Section I, there is only a paucity of contributions in the literature, mostly focusing on smart city IoT infrastructure challenges and/or data collection, processing and decision making aspects. Corchado *et al.* [12] proposes a cyber-physical platform for efficient management of smart territories, which solely focuses on data acquisition, classification, clustering, optimization and visualization that are mainly adopted for better decision-making. This paper indeed brings about better decision making for operational aspects, while it assumes that a system has already been deployed, where we fill this gap with deployment automation strategies, development and operations automation (DevOps) and AI enhancements as per Figure 1b, left of c and d, respectively.

In [13] they introduce a digital twin in Industry 4.0, where any industrial process produced by a physical object is assisted with digital replicas in the cyber space. More explicitly, the behaviour of a physical device is replicated, where an industrial machine becomes a software-enhanced object incorporating self-management capabilities. This concept of digital twin is also suitable for data analyses and better decision making, focused mainly on operational aspects in addition to monitoring, remote control, predicting behaviours of machines, which can ultimately aid in optimizing an already deployed factory floor. However, deployment automation, DevOps automation and specific AI enhancements are kept outside the focus of this concept. Next, Sotres *et al.* [14] mainly focused on the large-scale deployment of IoT infrastructure. Their practical field implementation proved that the manual deployment of such infrastructure is costly and time-consuming, which in return supports our hypothesis of the need for deployment automation. The authors of [14] summarized practical lessons from their on-site smart city implementation, where the significance of operational aspects, such as real-time monitoring of the devices and efficient management of data, was also stressed on, which again indicates the vast need for more attention on the automated deployment and operations that are the main contributions of our paper.

III. Towards Deployment Automation

Traditional devices in various domains, even if comprised of embedded systems for their control and operation, were not supporting connectivity to the internet and remote monitoring and management, but were in best case restricted to SCADA systems [16]. To benefit the life cycle automation, they need to be complemented by additional sensing, actuation and connectivity capabilities.

As identified in *Challenge-1* the initial deployment of smart devices to upgrade the capabilities of an existing infrastructure comes with considerable initial provisioning and configuration efforts [6]. As depicted in Figure 2a, when deploying embedded devices without standard I/O capabilities, such as touchscreen or keyboard, the traditional way of connecting them to an access point was via a universal serial bus (USB) or joint test action group (JTAG) cable to a computer [15]. This physical connection was followed by manual configuration using wireless credentials. Then, once being connected to the local network, not necessarily to the Internet, the devices are flashed and/or a configuration file with network credentials is transferred to them over a secure shell (SSH) connection. Finally, the access to the local network and/or to the Internet is tested.

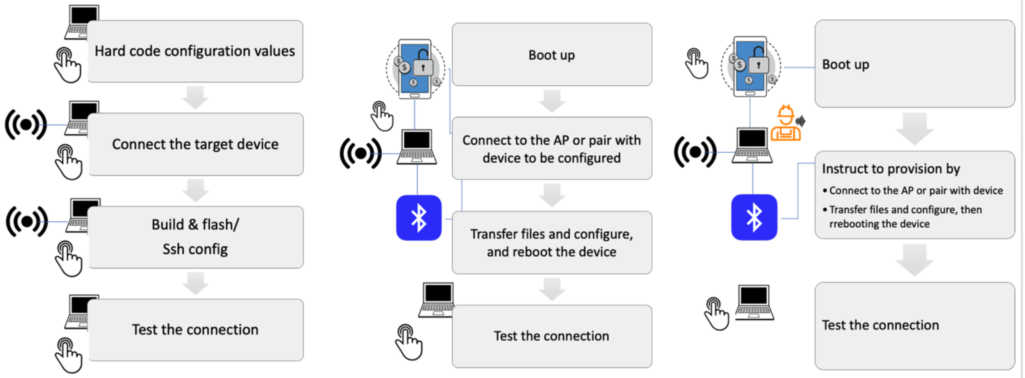
The above-mentioned traditional provisioning method could prove tedious, particularly if the user is not familiar with such highly technical process. Therefore, each new device to be on-boarded into the network tends to bring additional deployment delays, possibly leading to such substantial deployment times and costs as identified under Challenge-1. To this end, new deployment automation solutions are needed to speed up the deployment during *infrastructure enhancement*, as depicted in Figure 1b.

A. Zero-touch automation for deployment

The current state of the art with respect to deployment automation is represented by zero-touch provisioning (ZTP) methods [15] that do not only help reduce human-induced errors and deployment related delays, but also contribute to the efficiency of the work schedule of users or technicians. The main idea behind ZTP methods is to remove the necessity of physical connection for the configuration process and to minimize human interactions. More explicitly, the physical connection is replaced by a wireless connection that the devices allow upon boot-up and the configuration files are automatically fetched, as portrayed in Figure 2b.

ZTP methods can be realized either by leveraging existing industry standards, such as WiFi and Bluetooth, or by hinging on vendor-specific proprietary software and hardware solutions [17][1]. The main industry standards for a ZTP method include WiFi protected setup (WPS) [18], Push-Button-Connect (PBC) [18], software-enabled access point (Soft-AP) [19] and QR code [20]. WPS and PBC may require an external interface and a close proximity to the devices to be configured. With Soft-AP, the credentials need to be entered manually, while a QR code can be scanned by the mediator device to

[1] Commercial ZTP solutions are already available on the market from Cisco, Juniper, Apple and Texas Instruments as discussed in [15] Section II-B.


Hard code configuration values
Connect the target device
Build & flash/
Ssh config
Test the connection
Boot up
Connect to the AP or pair with device to be configured
Transfer files and configure, and reboot the device
Test the connection
Boot up
Instruct to provision by
• Connect to the AP or pair with device
• Transfer files and configure, then rrebooting the device
Test the connection

a) Traditional (manual). b) Automated zero-touch. c) Voice-assisted provisioning.

Fig. 2: Traditional vs. zero-touch provisioning vs voice assisted mass provisioning deployment scenarios.

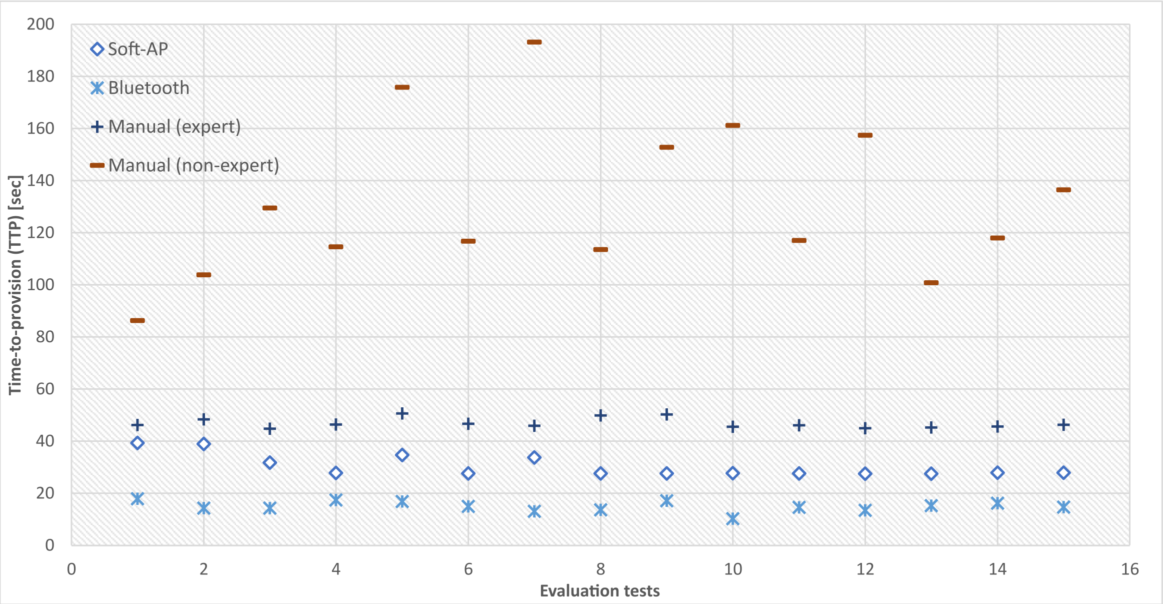


Fig. 3: Distribution of time-to-provision (TTP) evaluation tests is portrayed in [15] for two ZTP automated solutions and one manual expert provisioning of 15 times, while 15 non-experts manually provision relying on a device provisioning guideline without any previous knowledge.

create a reliable connection so as to transfer configuration files. As indicated, these standards require additional hardware components, such as a camera for scanning the QR code, close proximity to the devices and minor manual interactions.

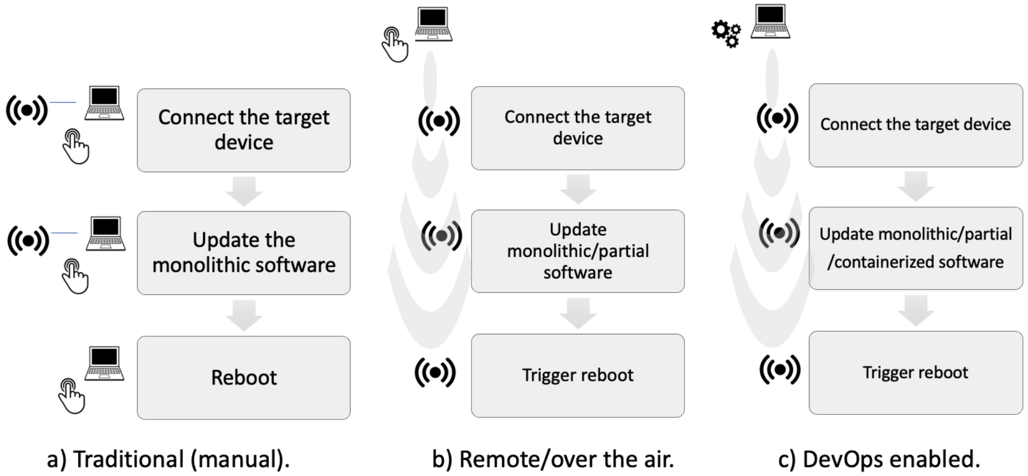
Figure 3 reproduced from [15] provides outcomes of preliminary survey on how automated ZTP solutions can shorten the deployment time of smart devices compared to manual provisioning solutions. For example, it can be seen that a person using Bluetooth-based ZTP solution can be more efficient than networking expert, who manually provisions a device, by up to 4.3 times, while this performance increases up to nearly 12.2 times when compared to a non-expert. Using Soft-AP based ZTP solution provides smaller improvement compared to the Bluetooth counterpart, but still it can realize the provisioning of a device nearly up to 1.6 times faster compared to an expert and up to 4.6 times faster than a non-expert.

According to the process detailed in Figure 2b and results in Figure 3, the ZTP approach has the potential to shorten the deployment time of the light bulbs exemplified in Challenge-1 and symbolically illustrated in Figure 1b, from 2 years to about half a year. However, both processes presented in Figure 2 a and b assume that the human needs to provide configuration details directly or through a mediator.

B. AI automation for voice-assisted provisioning

To further decrease the deployment effort, rather than manually clicking to trigger the connection/pairing as in Figure 2b, the human could trigger such actions automatically by giving instructions to a voice assistant as depicted in Figure 2c. While verbally controlling the provisioning process, the human could in parallel work on other physical node deployment tasks as per Figure 1b.

Voice-assistants are mostly general purpose [21] , however efforts to adapt an existing general purpose voice assistant to



Connect the target device
Update the monolithic software
Reboot
a) Traditional (manual).
Connect the target device
Update monolithic/partial software
Trigger reboot
b) Remote/over the air.
Connect the target device
Update monolithic/partial /containerized software
Trigger reboot
c) DevOps enabled.

Fig. 4: Traditional vs. remote/over the air vs DevOps enabled software updates and reconfigurations.

enable voice driven configuration of software defined networks has been proposed in [22]. Furthermore, very recent research [23] investigating the role of voice assistants with respect to productivity confirms that ”there is a positive relationship between satisfaction with voice based digital assistants and individuals productivity”.

Given the potential of this proposed automation step, to serve the purpose as envisioned in this section, the existing speech-to-text engine of the voice assistants would need to be optimized to understand technical configuration commands or infer them from higher level, non-technical, requests. They would also need to be very robust to various industrial noises. The speech-to-text engine is the core of a voice assistant and for adapting an existing one or developing a new one, a relevant corpus of speech recordings needs to be generated and a machine learning (ML) model needs to be adapted to perform the translation as per the guidelines in Section V-A. The text to executable command mapping could be done using predefined rules, however, especially when the need to infer more technical configuration commands from more generic talk, these could be enhanced by reasoning engines based on symbolic AI [24] to guide the configuration process as discussed Section V-B1.

IV. Towards Operation Automation

In this section, we discuss the two operations related sub-challenges identified in Section I starting from the current state-of-the-art, followed by beyond the state-of-the-art AI supported automation solutions that can push automation to the next level.

A. Automating the remote reprogramming infrastructure

As discussed in the introduction of Section III, early devices used in smart infrastructure were not supporting connectivity to the internet. In order to update their configurations such as sensor sampling rates, upgrade their firmware or software, the maintainers traditional process was manually intensive and involved physically connecting to each of them and typically updating the entire embedded software using monolithic software updating approaches [25] as depicted in Figure 4a. Conceptually and process-wise, this is analogous to Figure 2a and the related discussion in Section III.

As modern smart devices are connected to the internet or an intranet, remote reconfiguration/reprogramming [26] techniques were developed. This way, the process could be remotely triggered from the control room as depicted on the left side of Figure 1c. As the smart devices increasingly became connected via wireless technology, rather than wired, such updates were also referred to as over the air reprogramming as illustrated in Figure 4b. Due to decreased bandwidth and reliability or wireless communications, techniques to reduce the amount of data sent over the air were developed and referred to as partial updates [27] rather than monolithic.

1) DevOps automation: The modern pipelines supporting remote updates are automated by DevOps tools [28] initially developed for the management and automation of large cloud systems. Their development was triggered by the need to scale such capacity in a cost-efficient way with low human involvement and enabled largely by moving from monolithic software architecture to service oriented architectures of which microservices are an example [29]. Similar automation approaches have been proposed for networking, including cellular networks, and is referred to as Networking DevOps [30]. DevOps-inspired automation for smart devices, especially the ones falling into the category of embedded systems have also been recently proposed [31], [32] and are commercially available as surveyed and compared in [33].

Let us consider the smart grid scenario portrayed on the left side of Figure 1c, where developers prepare new code or models and system administrators put together new configurations for enabling reliable operation of the network of smart devices. These software artifacts are stored in an appropriate repository of code, configurations or binaries. Using DevOps automation, the manually triggered and custom scripts previously used for control as illustrated at the top of Figure 4b, are replaced by the automation system as illustrated in Figure 4c. Additionally, due to containerization technologies [34], that represent a popular way to pack and deploy microservices, also individual applications on smart nodes can be replaced independently of any other changes.

2) AI-aided automation: Along the various segments of the energy infrastructure depicted in Figure 1a tens of different types of energy services [35] will be deployed and maintained. For example consumption and production estimation services may be deployed at each transformer station in the distribution network to enable more agile demand-response. The demand-response estimators will then provide short and long term estimates. The deployment and update of the thousands of instances (i.e. software artifacts) of the energy services requires careful consideration. By looking at similar transformations taking place in the networking community, that is currently migrating to a software-defined paradigm, we notice that also in that case, the number of software modules (i.e. micro-services) that need to be managed and deployed is increasing. To manage such high number of services and their various versions and regulatory compliance, catalogues and app stores that host them are developed [36]. A similar approach will also be pursued for smart infrastructures.

To manage the plethora of software artifacts, efficient automated search [37] and recommendation based on compatibility checks will come useful. Rather than manually writing service chaining and configuration files that control the deployment of the software artifacts, the meta-data accompanying them can be provided in a standardized manner that enables automatic compatibility checks and recommendation. To enable that, the standard for representing the meta-data should provide a representation that is suitable for reasoning using symbolic AI, such as using lightweight semantic schemes as discussed in Section V-B2. This way, all the meta-data would be connected inside the machine as a large scale structured representation of information, also referred to as knowledge graph [38], on which a symbolic AI reasoning engine, as described in Section V-B3, could operate and infer suitable chaining and configurations.

While finding relevant, compatible and regulatory-compliant software artifacts is an important operational aspect, often also the needed to configure some of these artifacts according to operator or client needs. For instance, an energy trading platform may require data with certain frequency and reliability be provided by their subscribers while a demand-response service may want certain control reliability guarantees. In many such cases, threshold based rules, such as ”if the connectivity capacity of the network is at 80% usage, upgrade to a new plan with higher capacity” might suffice. However, in other cases, non-symbolic AI for automated short and long term capacity dimensioning of the IT infrastructure as proposed in [39] are likely to be used as support tools for an informed decision making rather than ”built-in” autonomous control loop elements. Such approach can automate repetitive manual planning as per the decision process and guidelines provided in Section V-A.

Finally, to make the overall process of controlling and configuring aspects of the network during the operation *as symbolically depicted in Figure 1c*, a specialized voice assistant, similar to the one discussed in Section IIIc, could be used to provide a more human-friendly interaction modality. Rather than requiring a human operator to check and manually edit the content of the automatically generated configuration files, the voice assistant could initiate a conversation related to the proposed set-up that would lead to and improved and approved version. This kind of AI automation is proposed for telecommunication networks in Vivonet [22] and intent-based networking [2]. Furthermore, in the last decade, various flavours of AI technologies have been considered as the solution for completely automated and autonomous network management that would replace manual decision makers in the loop [40]. AI-aided DevOps for increased automation has also been considered and is recently being referred to as AIOps [41].

TABLE I: Performance of supervised and unsupervised methods to detect the four link layer anomalies defined in [42].

Anomaly type	Approach	Method	F1 score [%]
SuddenD	Supervised	Logistic Regression (LR)	100
		LR + Autoencoder	100
		Random Forest (RF)	100
		RF + Autoencoder	100
		Support Vector Machine (SVM)	100
		SVM + Autoencoder	100
	Unsupervised	Local Outlier Factor (LOF)	76
		LOF + Autoencoder	38
		Isolation Forest (IF)	83
		IF + Autoencoder	97
		One-Class SVM (OCSVM)	98
		OCSVM + Autoencoder	99
SuddenR	Supervised	Logistic Regression (LR)	100
		LR + Autoencoder	100
		Random Forest (RF)	99
		RF + Autoencoder	100
		Support Vector Machine (SVM)	100
		SVM + Autoencoder	100
	Unsupervised	Local Outlier Factor (LOF)	98
		LOF + Autoencoder	74
		Isolation Forest (IF)	75
		IF + Autoencoder	98
		One-Class SVM (OCSVM)	95
		OCSVM + Autoencoder	84
InstaD	Supervised	Logistic Regression (LR)	97
		LR + Autoencoder	98
		Random Forest (RF)	97
		RF + Autoencoder	97
		Support Vector Machine (SVM)	98
		SVM + Autoencoder	98
	Unsupervised	Local Outlier Factor (LOF)	89
		LOF + Autoencoder	38
		Isolation Forest (IF)	70
		IF + Autoencoder	92
		One-Class SVM (OCSVM)	90
		OCSVM + Autoencoder	93
SlowD	Supervised	Logistic Regression (LR)	97
		LR + Autoencoder	100
		Random Forest (RF)	99
		RF + Autoencoder	100
		Support Vector Machine (SVM)	100
		SVM + Autoencoder	100
	Unsupervised	Local Outlier Factor (LOF)	36
		LOF + Autoencoder	24
		Isolation Forest (IF)	63
		IF + Autoencoder	91
		One-Class SVM (OCSVM)	71
		OCSVM + Autoencoder	95

B. Automating fault/anomaly detection

Operational networks need to be constantly monitored to ensure their reliable and SLA compliant functioning also for the edge devices. As they can be large in size and complex in terms of the forming elements and their interconnection, providing automated alerts for faults or anomalies [43] is a desired feature. As it can be seen on the right side of Figure 1c, monitoring data from smart devices should be collected, passed through fault/anomaly detection system, which then selects and presents detected anomaly events to a manual decision marker. The respective decision maker then decides whether the presented anomaly or fault is indeed a relevant anomaly rather than a misdetection, and decides on the appropriate course of action for handling.

Traditional fault/anomaly detection systems are reactive by nature assuming that the fault is noticed and reported by humans and the system is consulted to understand the cause of the fault and plan mitigation actions. Additionally, traditional systems ingested data in batches, sometimes daily, therefore immediate insight into a fault was not available to the operators.

1) Big data and AI aided automation: Current fault and anomaly detection tools mostly rely on big data, streaming databases and standard visualization dashboards that have emerged in the last few years [44]. Unlike with legacy systems, the steaming databases enable constant data ingestion and the operator is able to check and have a near-real time report of the situation. However, these tools tend to also be mostly reactive in nature and mostly enable visualizations of the raw time-series, simple aggregates and threshold based detection. Manually searching through dashboard-based visualizations to find a problem in a large software network with tens of thousands of software artifacts can take hours to weeks depending on the complexity of the system and gravity of the fault. Users still need to manually find the correct device for faults, check the dashboard for various metrics, understand the issue, check the status of other related devices and subsequently decide on the next step.

While automatic fault/anomaly detection is far from being a solved problem [45], various machine learning-based techniques for outlier [46] or motif [47] detection have been proposed. When an anomaly or fault is identified by such techniques, a near real-time warning can be sent to the operations control room, as depicted on the right side of Figure 1c, to inform and trigger remediation actions even before used or customer complaints are filed. Such a mechanism can change fault/anomaly detection from reactive to proactive, increase the efficiency of planning the remediation and decrees the overall remediation time. To develop such as system, sufficient training data for the target problem needs to be collected in order to train the ML algorithm as per the guidelines discussed in Section V-A. Attempts for automating fault and anomaly detection using algorithms from the realm of AI, such as ML mostly rely on offline models and synthetic data are also discussed more in depth in [40].

2) AI for more expressive and online fault detection: Faults or anomalies can also be defined in a more specific way that is suitable for a specific application. Once detected, such faults can already indicate the malfunctioning that causes them and ease the anomaly remediation from Figure 1c. The motif detection approaches identifying such faults can use both supervised and unsupervised, depending on the amount of labelled data that is available. For instance, [42] defines four wireless link level anomalies observed on edge devices: sudden link degradation (SuddenD), sudden link degradation with recovery (SuddenR), instantaneous link degradation (InstaD) and slow link degradation (SlowD). As it can be seen in Table I, the supervised methods generally yield better performance than the unsupervised counterparts for the respective anomalies.

V. AI Adoption Guidelines for Life Cycle Automation

The field of AI is comprised of efforts to develop synthetic mechanisms that mimic the intelligence found in nature. For instance, some AI approaches attempt to mimic the intelligence of simple organisms while others mimic human-like intelligence leveraging the ability of learning and problem solving. On high level, we distinguish between *non-symbolic* and *symbolic* AI [24]. We refer to *non-symbolic* or data-driven AI as the set of techniques that only use data to automatically develop a model. We refer to *symbolic* or knowledge-driven AI as a body of structured knowledge (i.e. Wikipedia) that can complement and enhance the models resulted from non-symbolic AI.

In this section we provide practical guidelines for adopting non-symbolic and symbolic AI, as depicted in on the right side of Figure 1d, that we visually compress on a decision diagram aimed at quickly guiding the decision makers in the adoption process. These guidelines were compiled based on existing experiences and best practices gleaned from other areas such as AI for cloud computing and data centres, telecommunication networks, human computer interaction and the experience of the authors of developing AI-driven automation solutions for cyber-physical systems. An alternative, more complex study that focuses on AI adoption is presented in [48] while a per firm-level AI readiness study is presented in [49]. Other drivers, barriers and social considerations of AI adoption have been recently studied in [4].

A. Non-symbolic AI

In Figure 5, we provide a decision diagram that serves as a guideline for when to adopt a data-driven AI technique to solve an automation problem. First, one should identify *knowledge tasks* that have to be repeatedly done by a human, as illustrated in the top decision element numbered with *1*. Then, either the task can be automatized reliably by writing a set of rules or it can be supported by the collected sufficient data (labelled or unlabelled). For instance, consider the wireless link layer anomaly example of [42] discussed in Section IV-B2. If a person is able to look at a time series coming from a smart meter and recognize a specific anomaly within, then such task is worth automating if it needs to be performed regularly and at scale. In such case, one needs to see if there is a *set of rules* as per decision *2* in Figure 5 that can be used to accurately and automatically realize the device recognition performed by

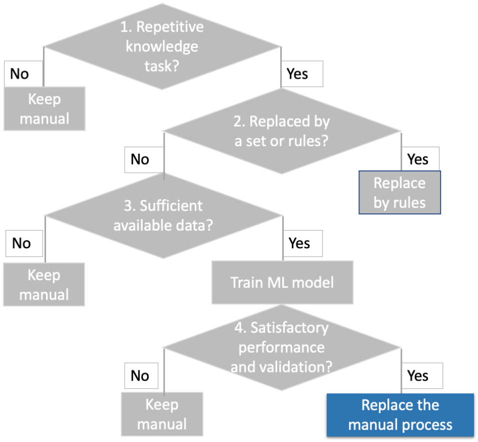

1. Repetitive knowledge task?
No
Yes
Keep manual
2. Replaced by a set or rules?
No
Yes
Replace by rules
3. Sufficient available data?
No
Yes
Keep manual
Train ML model
4. Satisfactory performance and validation?
No
Yes
Keep manual
Replace the manual process

Fig. 5: Decision diagram guiding the decision process of AI adoption.

a human. If the answer is positive, then the set of rules will suffice and AI is likely not needed.

For the case when rules for detection do not suffice and enough data can be collected as per decision *3* in Figure 5, a ML model can be developed to automatize the recognition task. If the performance of the model is satisfactory and complies with the business requirements as per decision *4*, then the manual process can be replaced by the model developed using ML. In such case, the learnt model can be deployed to the production system.

B. Symbolic AI

The decision process for the adoption of the less popular knowledge-driven or symbolic AI is more straightforward. There are essentially three cases when it should be employed.

1) Case1: If a business aspect uses a set of rules and meta-rules that have evolved to become too complex and hard to maintain, they should be swapped for a knowledge-driven AI system that includes knowledge representation and reasoning components.

2) Case2: When the internal data or meta-data is becoming large and difficult to maintain, lacks schema inter-operability and is perhaps not even machine-readable, scalable inter-interoperability rules as discussed in the previous case are not even possible before a standard descriptive representation language is employed.

3) Case3: One might want to develop an expressive and comprehensive digital representation of the network and all its elements and their interconnections, sometimes referred to also as the network's digital twin. The state-of-the-art realization, management and exploitation of such digital twin is represented by the knowledge graph that uses structured knowledge-driven AI for describing and representing its various entities.

VI. Summary

In this paper, we analyzed the two main challenges that hinder the large scale enhancement of legacy infrastructures such as smart grids, i.e., deployment and operations. We then discussed the state of the art in automating aspects of deployment and operation to increase efficiency and reduce costs. Furthermore, we analysed the role of AI in automating deployment, software update and anomaly detection in the emerging smart infrastructures, provided future directions and guidelines on how to approach the adoption of AI.

References

[1] S. Makridakis, "The forthcoming Artificial Intelligence (AI) revolution: Its impact on society and firms," *Futures*, vol. 90, pp. 46–60, 2017.

[2] T. Szyrkowiec, M. Santuari, M. Chamania, D. Siracusa, A. Autenrieth, V. Lopez, J. Cho, and W. Kellerer, "Automatic intent-based secure service creation through a multilayer sdn network orchestration," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 10, no. 4, pp. 289–297, 2018.

[3] D. Diran, A. F. Van Veenstra, T. Timan, P. Testa, and M. Kirova, "Briefing (requested by the AIDA committee): Artificial intelligence in smart cities and urban mobility," https://www.europarl.europa.eu/RegData/etudes/BRIE/2021/662937/IPOL_BRI(2021)662937_EN.pdf, 2021, accessed on 2.1.2022.

[4] M. Cubric, "Drivers, barriers and social considerations for ai adoption in business and management: A tertiary study," *Technology in Society*, vol. 62, p. 101257, 2020.

[5] R. I. Ogie, P. Perez, and V. Dignum, "Smart infrastructure: an emerging frontier for multidisciplinary research," *Proceedings of the Institution of Civil Engineers-Smart Infrastructure and Construction*, vol. 170, no. 1, pp. 8–16, 2017.

[6] J. Wilhelm, J. Williams, and S. Macy. (2017) Whitepaper on IoT onboarding - a device manufacturers perspective. [Online]. Available: https://www.intel.com/content/dam/www/public/us/en/documents/white-papers/kaiser-associates-iot-onboarding-for-device-manufacturers-whitepaper.pdf

[7] O. Pauzet, "Cellular communications and the future of smart metering," *Sierra Wireless, Inc*, 2010.

[8] Y. Yan, Y. Qian, H. Sharif, and D. Tipper, "A survey on smart grid communication infrastructures: Motivations, requirements and challenges," *IEEE communications surveys & tutorials*, vol. 15, no. 1, pp. 5–20, 2012.

[9] B. N. Silva, M. Khan, and K. Han, "Towards sustainable smart cities: A review of trends, architectures, components, and open challenges in smart cities," *Sustainable Cities and Society*, vol. 38, pp. 697–713, 2018.

[10] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Communications surveys & tutorials*, vol. 18, no. 2, pp. 1153–1176, 2015.

[11] M. Erol-Kantarci and H. T. Mouftah, "Energy-efficient information and communication infrastructures in the smart grid: A survey on interactions and open issues," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 1, pp. 179–197, 2014.

[12] J. M. Corchado, P. Chamoso, G. Hernndez, and et al., "Deepint.net: A rapid deployment platform for smart territories," *Sensors*, vol. 21, no. 1, 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/1/236

[13] M. Groshev, C. Guimares, J. Martn-Prez, and A. de la Oliva, "Toward intelligent cyber-physical systems: Digital twin meets artificial intelligence," *IEEE Communications Magazine*, vol. 59, no. 8, pp. 14–20, 2021.

[14] P. Sotres, J. R. Santana, L. Snchez, J. Lanza, and L. Muoz, "Practical lessons from the deployment and management of a smart city internet-of-things infrastructure: The smartsantander testbed case," *IEEE Access*, vol. 5, pp. 14 309–14 322, 2017.

[15] I. Boshkov, H. Yetgin, M. Vucnik, C. Fortuna, and M. Mohorcic, "Time-to-Provision Evaluation of IoT Devices Using Automated Zero-Touch Provisioning," in *IEEE Globecom 2020*, 2020, pp. 363–368.

[16] J. Figueiredo and J. S. da Costa, "A scada system for energy management in intelligent buildings," *Energy and Buildings*, vol. 49, pp. 85–98, 2012.

[17] Cisco zero-touch provisioning configuration guide, Cisco IOS XE 16.x.x. (accessed: 09.12.2019). [Online]. Available: https://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst3850/software/release/16-5/configuration_guide/prog/b_165_prog_3850_cg/zero_touch_provisioning.pdf

[18] S. Viehböck, "Brute forcing WiFi protected setup," *Wi-Fi Protected Setup*, vol. 9, 2011.

[19] Iotivity easy setup, wifi provisioning. (accessed: 28.10.2019). [Online]. Available: https://wiki.iotivity.org/easy_setup

[20] Y. Liu, J. Yang, and M. Liu, "Recognition of QR code with mobile phones," in *IEEE Chinese control and decision conference*, Yantai, Shandong, China, July 2008.

[21] K. Georgila, A. Leuski, V. Yanov, and D. Traum, "Evaluation of off-the-shelf speech recognizers across diverse dialogue domains," in *Proceedings of the 12th Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, May 2020, pp. 6469–6476.

[22] A. Chaudhari, A. Asthana, A. Kaluskar, D. Gedia, L. Karani, L. Perigo, R. Gandotra, and S. Gangwar, "VIVoNet: Visually-represented, intent-based, voice-assisted networking," *International Journal of Computer Networks & Communications (IJCNC)*, vol. 11, no. 2, 2019.

[23] D. Marikyan, S. Papagiannidis, O. F. Rana, R. Ranjan, and G. Morgan, "alexa, lets talk about my productivity: The impact of digital assistants on work productivity," *Journal of Business Research*, vol. 142, pp. 572–584, 2022.

[24] R. Sun and L. A. Bookman, *Computational architectures integrating neural and symbolic processes: A perspective on the state of the art*. Springer Science & Business Media, 1994, vol. 292.

[25] C.-C. Han, R. Kumar, R. Shea, and M. Srivastava, "Sensor network software update management: a survey," *International Journal of Network Management*, vol. 15, no. 4, pp. 283–294, 2005.

[26] R. Oliver, A. Wilde, and E. Zaluska, "Reprogramming embedded systems at run-time," in *Proceedings of the International Conference on Sensing Technology, ICST*, 2014.

[27] P. Ruckebusch, E. De Poorter, C. Fortuna, and I. Moerman, "Gitar: Generic extension for internet-of-things architectures enabling dynamic updates of network and application modules," *Ad Hoc Networks*, vol. 36, pp. 127–151, 2016.

[28] C. Ebert, G. Gallardo, J. Hernantes, and N. Serrano, "Devops," *Ieee Software*, vol. 33, no. 3, pp. 94–100, 2016.

[29] M. Villamizar, O. Garcés, L. Ochoa, H. Castro, L. Salamanca, M. Verano, R. Casallas, S. Gil, C. Valencia, A. Zambrano *et al.*, "Cost comparison of running web applications in the cloud using monolithic, microservice, and aws lambda architectures," *Service Oriented Computing and Applications*, vol. 11, no. 2, pp. 233–247, 2017.

[30] J. Kim, C. Meirosu, I. Papafili, R. Steinert, S. Sharma, F.-J. Westphal, M. Kind, A. Shukla, F. Németh, and A. Manzalini, "Service provider DevOps for large scale modern network services," in *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*. IEEE, 2015, pp. 1391–1397.

[31] E. Yigitoglu, M. Mohamed, L. Liu, and H. Ludwig, "Foggy: A framework for continuous automated iot application deployment in fog computing," in *2017 IEEE International Conference on AI & Mobile Services (AIMS)*. IEEE, 2017, pp. 38–45.

[32] M. Vucnik, T. Solc, U. Gregorc, A. Hrovat, K. Bregar, M. Smolnikar, M. Mohorcic, and C. Fortuna, "Continuous integration in wireless technology development," *IEEE Communications Magazine*, vol. 56, no. 12, pp. 74–81, 2018.

[33] K. Arakadakis, P. Charalampidis, A. Makrogiannakis, and A. Fragkiadakis, "Firmware over-the-air programming techniques for iot networks-a survey," *ACM Computing Surveys (CSUR)*, vol. 54, no. 9, pp. 1–36, 2021.

[34] M. Al-Rakhami, A. Gumaei, M. Alsahli, M. M. Hassan, A. Alamri, A. Guerrieri, and G. Fortino, "A lightweight and cost effective edge intelligence architecture based on containerization technology," *World Wide Web*, vol. 23, no. 2, pp. 1341–1360, 2020.

[35] M. J. Fell, "Energy services: A conceptual review," *Energy research & social science*, vol. 27, pp. 129–140, 2017.

[36] L. Bondan, M. F. Franco, L. Marcuzzo, G. Venancio, R. L. Santos, R. J. Pfitscher, E. J. Scheid, B. Stiller, F. De Turck, E. P. Duarte *et al.*, "Fende: marketplace-based distribution, execution, and life cycle management of vnfs," *IEEE Communications Magazine*, vol. 57, no. 1, pp. 13–19, 2019.

[37] A. Brogi, D. Neri, and J. Soldani, "Dockerfinder: multi-attribute search of docker images," in *2017 IEEE International Conference on Cloud Engineering (IC2E)*. IEEE, 2017, pp. 273–278.

[38] J. Pujara, H. Miao, L. Getoor, and W. Cohen, "Knowledge graph identification," in *International Semantic Web Conference*. Springer, 2013, pp. 542–557.

[39] D. Bega, M. Gramaglia, M. Fiore, A. Banchs, and X. Costa-Perez, "Aztec: Anticipatory capacity allocation for zero-touch network slicing," in *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications*, 2020, pp. 794–803.

[40] C. Benzaid and T. Taleb, "AI-Driven Zero Touch Network and Service Management in 5G and Beyond: Challenges and Research Directions," *IEEE Network*, vol. 34, no. 2, pp. 186–194, 2020.

[41] Y. Dang, Q. Lin, and P. Huang, "AIOps: real-world challenges and research innovations," in *2019 IEEE/ACM 41st International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)*. IEEE, 2019, pp. 4–5.

[42] G. Cerar, H. Yetgin, B. Bertalanic, and C. Fortuna, "Learning to Detect Anomalous Wireless Links in IoT Networks," vol. 8, November 2020, pp. 212 130–212 155.

[43] G. Fernandes, J. J. Rodrigues, L. F. Carvalho, J. F. Al-Muhtadi, and M. L. Proença, "A comprehensive survey on network anomaly detection," *Telecommunication Systems*, vol. 70, no. 3, pp. 447–489, 2019.

[44] L. Rettig, M. Khayati, P. Cudré-Mauroux, and M. Piórkowski, "Online anomaly detection over big data streams," in *Applied Data Science*. Springer, 2019, pp. 289–312.

[45] R. Wu and E. Keogh, "Current time series anomaly detection benchmarks are flawed and are creating the illusion of progress," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–1, 2021.

[46] M. Gupta, J. Gao, C. C. Aggarwal, and J. Han, "Outlier detection for temporal data: A survey," *IEEE Transactions on Knowledge and data Engineering*, vol. 26, no. 9, pp. 2250–2267, 2013.

[47] S. Torkamani and V. Lohweg, "Survey on time series motif discovery," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 7, no. 2, p. e1199, 2017.

[48] J. Wanner, K. Heinrich, C. Janiesch, and P. Zschech, "How much ai do you require? decision factors for adopting ai technology." in *ICIS*, 2020.

[49] S. Alsheibani, Y. Cheung, and C. Messom, "Artificial intelligence adoption: Ai-readiness at firm-level." in *PACIS*, 2018, p. 37.

[50] A. Bosselut, R. Le Bras, and Y. Choi, "Dynamic neuro-symbolic knowledge graph construction for zero-shot commonsense question answering," in *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI)*, 2021.