Platform-Agnostic Software Architecture for Edge Automation of Low-Voltage Distribution Grids

Erdem Gümrükcü, Debopama Sen Sarma,
Eaton Research Labs (ERL), Eaton Industries GmbH, Aachen, Germany
Email: [erdemguemruekcue,debopamasensarma]@eaton.com

***Abstract*—The proliferation of distributed energy resources is transforming power distribution grids, driving the deployment of advanced equipment to monitor and manage grid operations and end-user flexibility. This evolution introduces significant challenges in data volume, communication complexity, and device diversity within grid automation systems. To address these, we propose a modular and interoperable software architecture tailored for deployment at secondary substations. The architecture enhances grid observability, supports real-time responsiveness, and enables the integration of end-user flexibility—advancing digitalization and edge automation in distribution networks. Core design principles include modularity, interoperability across logic, data storage, and communication layers, and platform agnosticism to ensure broad applicability across hardware and software ecosystems. The system supports bidirectional communication: southbound with end-devices such as phasor measurement units and control actuators, and northbound with cloud-based applications that process grid constraints and generate setpoints for flexible edge assets. A reference implementation using EdgeX Foundry demonstrates feasibility, considering current edge hardware capabilities and limitations.**

***Index Terms*—distribution grid management, edge automation, situational awareness, software design, substation digitalization**

I. Background and Motivation

In recent decades, countries worldwide have witnessed a significant rise in distributed energy resources (DERs) owned by end users, particularly those capable of both generation and consumption and connected to distribution grids—most notably within the low-voltage (LV) segment. In Germany, 66% of the 10.7 GW of newly installed photovoltaic (PV) capacity in 2024 was integrated into LV grids [1]. Additionally, 64.6% of newly constructed residential buildings now use electric heat pumps, and electric vehicles (EVs) account for 18% of all registered cars [2]. In China, distributed and residential battery storage reached 74.3 GWh, representing 46% of the nation's total installed energy storage capacity by the end of 2024. This proliferation of DERs presents multifaceted challenges for distribution grid management. Increased electricity demand from EVs and heat pumps, combined with the intermittency of PV generation, intensifies the need for enhanced operational flexibility and real-time grid observability. These dynamics can place considerable stress on LV networks, manifesting as extreme voltage fluctuations in feeders and overloading of power cables and transformers. Without advanced and localized intelligence—featuring on-premises data processing, adaptive control logic, and interoperable communication frameworks—grid automation systems may struggle to maintain stability. Currently, for most distribution system operators (DSOs), grid intelligence typically terminates at secondary substations, leaving LV networks largely unobserved and unmanaged. To improve situational awareness and harness the flexibility available at the LV level, modular edge computing solutions are essential. These capabilities are critical to achieving the broader goals of a sustainable, secure, and resilient electricity system.

Building on the outlined requirements, recent literature presents various edge computing solutions for distribution grid management. These works primarily target the architectural facets of edge automation, exploring different configurations of hardware, communication frameworks, and control hierarchies. In [3], a hierarchical intelligence framework is proposed using advanced edge hardware at substations and end devices. While communication protocols and infrastructure are well detailed, functional modules beyond algorithm containerization are minimally addressed. Similarly, [4] introduces a multilayered fog computing architecture enabling data exchange among end users, aggregators, and operators. This structure improves scalability and responsiveness by processing data locally, though it increases installation complexity and cost. Z. Xu et al. [5] propose a fully distributed system where autonomous edge nodes collaboratively achieve system-wide consensus. Despite its scalability and resilience, widespread deployment faces economic and interoperability challenges due to hardware costs and standardization needs. To mitigate complexity, [6] explores hybrid frameworks combining edge nodes with centralized cloud control. However, reliance on top-down management limits edge autonomy, crucial for real-time operations. Addressing this, M. Sun et al. [7] present a Bandwidth Release Model (BRM) that dynamically allocates communication and computing resources across edge and central systems. While effective in optimizing resource use, its implementation must consider network topology, latency, and device heterogeneity in LV grids.

Complementing the previously discussed cluster of contributions, several studies focus on the functional aspects of edge automation, offering detailed insights into the design of edge computing systems within distribution grid environments. These works extend beyond hardware specifications to examine the algorithms deployed, their operational features, data

All work presented in this paper is funded for the BEAVER project by Federal Ministry for Economic Affairs and Energy (BMWE) under the Grant number: 03EI6125A

979-8-3315-8138-1

TABLE I
CONTRIBUTIONS TARGETING FUNCTIONAL ASPECTS OF EDGE AUTOMATION

Item	Summary
[8]	Voltage regulation framework leveraging cloud-edge collaboration
[9]	Joint energy management scheme to coordinate EV operations with local distribution substations
[10]	Conservation voltage regulation (CVR) strategy employing edge devices at DER-equipped nodes in collaboration with legacy devices

requirements, and interactions with other grid applications. They underscore the importance of functional transparency, particularly in algorithmic design, data handling, and system integration. However, in most cases, the functional characteristics of the proposed edge systems are narrowly tailored to the specific needs of individual algorithms—often addressing a singular objective such as voltage regulation, as summarized in Table I. As a result, existing implementations focus narrowly on algorithmic logic, lacking a comprehensive framework that integrates monitoring and control. This limits scalability and adaptability in complex LV grids, where multifunctional intelligence and seamless system integration are critical.

While edge computing is increasingly adopted in grid automation, many existing approaches treat architectural and functional aspects in isolation. This fragmentation hampers modularity, interoperability, and scalability—key requirements for deployment in real-world distribution grids. To address these limitations, this paper presents the following contributions:

- A unified functional template for intelligent edge nodes, tailored for deployment at secondary substations. This template enables real-time monitoring and decentralized flexibility management, supporting dynamic grid operations.
- Platform-agnostic deployment capabilities across edge and cloud environments. This allows computationally intensive tasks to be offloaded to the cloud while maintaining edge responsiveness. The template also supports integration of third-party software developed in any programming language, fostering flexibility in algorithm design and implementation.
- A reference implementation based on EdgeX Foundry[1], an open-source edge automation platform selected for its built-in orchestration, scheduling, and seamless integration of control logic, data handling, and communication workflows. By offering a standardized architecture, the proposed template facilitates integration of advanced control and optimization frameworks from prior research, as demonstrated in Section III.

II. PROPOSED SOFTWARE ARCHITECTURE

In this section, we present a detailed overview of the proposed functional template for an intelligent edge computing node. Notably, the edge computing device is deployed exclusively at secondary substations or MV/LV transformer boxes. Terminal devices connected to the system include measurement sensors, control actuators, and privately installed energy management (EM) units in residential or commercial buildings.

A. Functional Template for the Edge Computing Node

Fig. 1 illustrates the proposed functional template, which has two distinguishing properties. First, it adopts a layered architecture comprising northbound and southbound communication interfaces, edge applications, and edge data management modules. Second, regardless of their position within the architecture, all edge software components are containerized to ensure versatility across diverse edge computing environments and CPU architectures, such as ARM and x86. Containerization enables self-contained software modules, each capable of functioning as a standalone computing unit in shared environments. When their input/output relationships are logically sequenced, these containers can seamlessly execute complex tasks. This approach supports platform agnosticism, scalability, and simplified maintenance. The following subsections elaborate on the layers of this functional template.

1) Northbound Connectors: The functional template includes a northbound communication interface, which can be utilized to connect edge nodes to multiple backend systems. In the presented case, the connected endpoint is the centralized Energy Management System (EMS) of the DSO. Within the broader grid automation ecosystem, this EMS monitors several edge nodes to retrieve relevant operational variables, such as power injection at the interface between the LV and MV grids. In future implementations, the same northbound interface can be extended to transmit control signals from the central

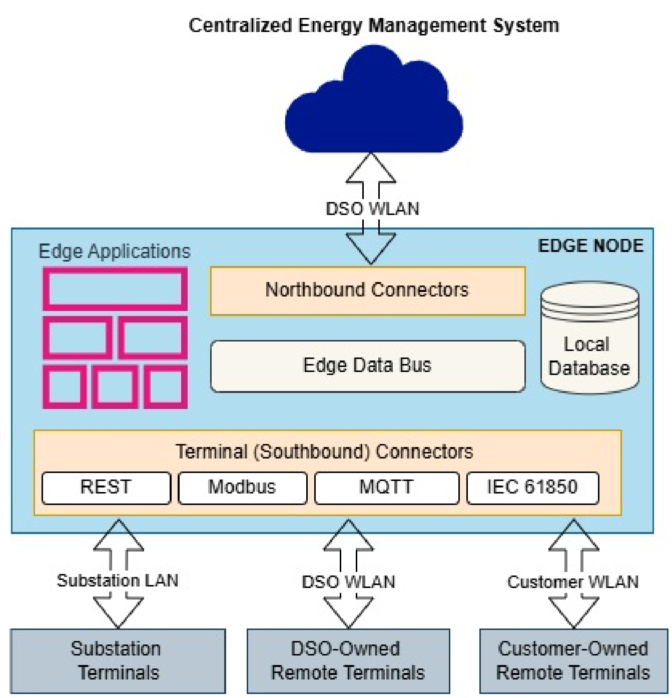

Fig. 1. Funtional Template for the Edge Computing Node

[1]EdgeX Foundry, "An Open Source Platform for Edge Computing," The Linux Foundation. [Online]. Available: https://www.edgexfoundry.org

979-8-3315-8138-1

EMS to the secondary substation, such as day-ahead capacity constraints for the LV grid. The communication between the central EMS and edge nodes can be implemented using various technologies, including MQTT, REST, or legacy systems.

2) Edge Applications: In this context, edge applications refer to the algorithmic functions executed by edge nodes—specifically, monitoring and flexibility management. Together, these applications form a comprehensive framework for holistic distribution grid management. This integrated approach enables real-time sensing of grid measurements, detection and resolution of critical grid conditions, and optimal utilization of available DERs. As a result, the system supports secure and efficient operation of renewable-powered distribution grids, enhancing resilience and enabling smarter and more adaptive energy systems.

- **Monitoring applications** enhance grid observability performing tasks such as state estimation, power flow analysis, and fault detection. These algorithms also contribute to resolving critical grid conditions once identified, such as transformer or line congestion and fault restoration. They require detailed knowledge of the physical grid infrastructure and must be supplied with real-time measurements from field devices. Additionally, they issue control signals to actuators connected to end devices to mitigate congestion and restore normal operation. Given their reliance on proprietary grid data and their role in maintaining grid stability, these applications should be managed by the associated DSO, who possesses the necessary measurements and operational authority to execute control actions such as power flow adjustments.
- **Flexibility management applications** perform single- or multi-objective constrained optimization routines by leveraging controllable assets owned by residential and commercial end users, such as rooftop PV systems, battery storage units, and EV charging stations. Because of the limited processing and storage capabilities of edge devices, traditional solver-based optimization algorithms are impractical because of their high computational demands. Moreover, flexibility services are typically provided by third-party aggregators or service providers, who lack access to the complete dataset required for centralized optimization. To address these constraints, we adopt and recommend distributed optimization methods, which can be periodically reset based on a centralized algorithm executed at a computationally capable control center. The integration of real-time feedback mechanisms significantly reduces the need for repeatedly solving complex nonlinear and nonconvex power flow equations. Instead, these mechanisms can leverage outputs from the state estimation module embedded within the monitoring applications, streamlining the optimization process and enhancing responsiveness.

3) Data Handling and Storage: Two primary data categories support the computational tasks required for distribution grid management: static data and dynamic data. The first category, static data, includes information such as grid topology, nominal parameters, and operational constraints of both the network and connected devices. This data remains unchanged during algorithm execution and can be loaded once prior to computation. In this paper, this static information is referred to as metadata and is stored locally in a database within the edge environment. Given its proprietary nature, metadata typically belongs to the DSO for the physical network and to end users for their respective flexible assets. To ensure data security and privacy, the database provides authorized access to specific northbound applications, primarily the monitoring modules. The second category, dynamic data, encompasses real-time field measurements, current flexibility limits from end users, and the operational states of controllable assets—such as the state-of-charge of battery storage systems. This data is time-sensitive and must be refreshed at regular intervals to maintain relevance and optimize storage efficiency. In the proposed architecture, a DataBus facilitates the exchange of dynamic data. Southbound devices write their measurements to the DataBus, from which northbound applications can retrieve the necessary information. Each data packet is tagged with a unique identifier, enabling precise source attribution, such as distinguishing between data written by a monitoring container and that from a phasor measurement unit.

4) Terminal Layer: The final layer of the proposed architecture defines the interface between the edge computing node—typically located at a secondary substation—and a diverse set of field devices that provide real-time measurements. As shown in Fig. 1, these field devices may be located either within the substation or at remote sites. The remote locations can be owned by the DSO or by customers connected to the LV grid. In all cases, a corresponding communication network is required. For generality, three types of communication networks are identified: the substation local area network (LAN), the DSO wide area network (WAN), and the customer WAN. These networks support data exchange using various communication protocols (e.g., MQTT, REST) and data models (e.g., IEEE 2030.5, OpenADR).

B. Features of the Functional Template

The defining features of the functional template can be grouped into three categories. The first includes design-time features such as modularity and scalability, which define the architecture of the template itself. The second category comprises environmental features—namely cybersecurity and interoperability—enabled by the chosen implementation platform, i.e., EdgeX Foundry. The third category addresses with hardware compatibility.

1) Modularity and Scalability: The proposed tiered architecture distinctly separates data management processes from operational algorithms. Within the edge applications, monitoring and flexibility management are further modularized into distinct components, each addressing specific concerns. Communication between these modules uses standardized interfaces, promoting interoperability with external computing resources and field devices. Well-defined application program-

979-8-3315-8138-1

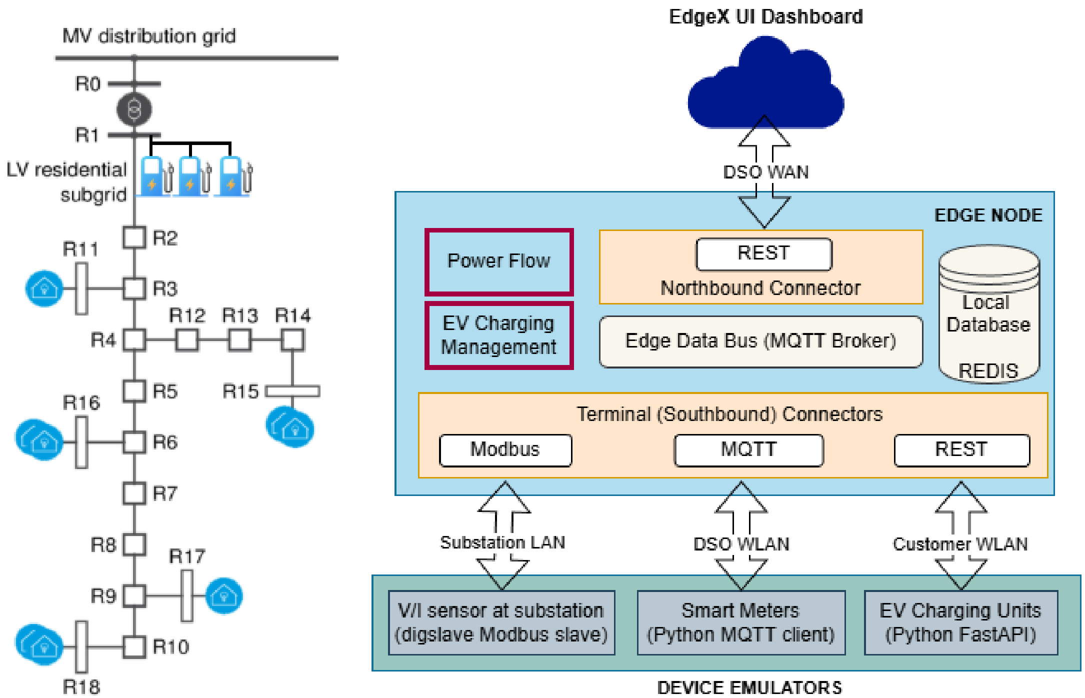

Fig. 2. Implementation Configuration – CIGRE LV Residential Grid (left) and Functional Template with Device Emulators (right)

ming interfaces (APIs) encapsulate sensitive details, such as grid operational data or proprietary software logic, to ensure secure and controlled data exchange between applications and storage systems. Each module can be developed, deployed, and maintained independently, enabling scalable integration of third-party solutions. Containerization further enhances modularity by supporting a plug-and-play architecture. Containers, Compared to virtual machines, are more resource-efficient and support horizontal scaling and platform independence. Therefore, the architecture adheres to all essential principles of a modular and scalable design.

2) Cyber-Security and Interoperability: EdgeX Foundry ensures cybersecurity and interoperability through a combination of architectural principles, standardized interfaces, and collaborative ecosystem support. Advanced security features, including support for protocols such as MQTT over TLS, are integrated into both northbound and southbound interfaces as well as the internal message bus, to ensure secure data exchange between applications and field devices while minimizing risks of unauthorized access or data tampering. The platform's support for a wide range of standardized protocols enhances interoperability, enabling EdgeX-based systems to communicate seamlessly with diverse field devices. The underlying Linux-based operating system contributes additional security through features like read-only partitions and secure boot, maintained through a community-driven development model. Furthermore, containerization helps isolate individual components, limiting the impact of potential security breaches and enabling independent deployment and updates. EdgeX is designed to run on any hardware, operating system, or edge/cloud environment, offering broad deployment flexibility. In our implementation, we leverage EdgeX's native capabilities while integrating custom modules into the appropriate southbound and northbound layers of the architecture.

3) Hardware Compatibility: The EdgeX services can be deployed on diverse systems, ranging from single-board computers such as the Raspberry Pi to enterprise-grade cloud infrastructure like Microsoft Azure. Commercial edge automation devices typically operate on Linux or Windows operating systems and offer hardware configurations with up to 16 GB of RAM and 64 GB of storage.

III. Results and Discussion

For demonstration purposes, we considered a scenario in which an edge node deployed at a secondary substation performs the following tasks: (1) real-time monitoring of transformer load, (2) determination of available capacity for EV charging, and (3) optimization of power dispatch to EV chargers based on consumer preferences and prevailing load conditions. This scenario was implemented on the 400 V residential subnetwork of the CIGRE LV Benchmark Grid [11]. The setup includes inflexible residential loads distributed across five nodes and ten flexible EV charging stations connected to the low-voltage side of the MV/LV transformer. A reference edge computing node was developed to execute the required monitoring and control functions within a fully digitized testing environment. For containerized deployment, we utilized Docker[2].

Fig. 2 illustrates the configuration of our reference implementation. The system interfaces with three external compo-

[2]Docker Inc., "Docker: Empowering App Development for Developers," [Online]. Available: https://www.docker.com

979-8-3315-8138-1

TABLE II
RESOURCE USE OF EDGE NODE COMPONENTS

Container	CPU (max)	Memory	Network
EV Control	2.50%	50 MB	54 KB/min
Power Flow	3.50%	114 MB	12 KB/min
Edge Database	1.47%	6 MB	33 KB/min
Terminal MQTT	0.36%	9 MB	52 KB/min
Terminal REST	3.11%	33 MB	123 KB/min

nents: (1) a Modbus server hosting substation sensor data, (2) MQTT clients streaming real-time smart meter data from residential consumers, and (3) REST APIs for updating EV charger setpoints. These endpoints are emulated in our test setup (highlighted in green). To enable seamless integration, the edge node (shown in light blue) incorporates corresponding EdgeX device services as interface connectors. Northbound communication is limited to periodic data streaming for LV grid monitoring via the EdgeX user interface (UI). For data handling, the system includes an MQTT broker as the edge DataBus and a local REDIS database.

For testing, the edge node was configured to update EV charger setpoints every 15 seconds. It runs two edge applications: a power flow algorithm from the pandapower library [12] to compute the available EV charging margin, and a rule-based charging management algorithm from [13] to allocate this margin. For instance, when residential loads report a combined consumption of 80 kW (at 5% load factor), the power flow container estimates transformer loading at 89% (500 kVA), leaving 55 kW available for EV charging. This margin is then optimally distributed across connected EVs.

To assess the computational and communication load of the edge node, we analyzed key metrics: peak CPU usage, memory footprint, and average network traffic per minute. Table II summarizes resource consumption across individual microservices. CPU usage is normalized to an AMD Ryzen 7 PRO 8845HS for relative comparison purposes. Resource demands scale with problem size—e.g., more EV chargers or grid nodes increase memory and network usage linearly, while CPU load grows exponentially. Depending on computational needs, algorithmic adaptations—such as code optimization or lightweight alternatives—can help reduce resource demands.

IV. CONCLUSION

This paper presents a software architecture for edge automation in distribution grids, aimed at improving observability and flexibility management in LV networks. The proposed functional template features a layered structure: (1) a northbound connection layer for communication with higher-level systems; (2) an edge application layer for executing monitoring and optimization algorithms; (3) localized data handling for efficient processing and storage; and (4) a southbound terminal layer interfacing with field devices via standardized protocols. The architecture is modular, allowing components to operate independently or in combination, depending on the use case. Containerization ensures platform agnosticism and enables seamless deployment across edge and cloud environments. Hardware constraints typical of edge devices—such as limited memory and processing power—are addressed through optimized storage design and algorithm selection, ensuring reliable and efficient operation.

ACKNOWLEDGMENT

The authors acknowledge the use of Microsoft Copilot for reviewing this document, including support with grammar, spelling, and language refinement. We also thank the Eaton Center of Excellence for Distributed Energy Resource Management, Canada, for their support in the EdgeX-based reference implementation.

REFERENCES

[1] German Network Agency (Bundesnetzagentur), "Growth in renewable energy in 2024", January 2025, url: https://www.bundesnetzagentur.de/SharedDocs/Pressemitteilungen/EN/2025/20250108_EE.html (last accessed: 20.10.2025)

[2] German Federal Statistical Office, url: https://www.destatis.de/EN/Press/2022/06/PE22_226_31121.html,https://www.destatis.de/Europa/EN/Topic/Transport/Car.html (last accessed: 20.10.2025)

[3] J. Stoupis, R. Rodrigues, M. Razeghi-Jahromi, A. Melese and J. I. Xavier, "Hierarchical Distribution Grid Intelligence: Using Edge Compute, Communications, and IoT Technologies," in IEEE Power and Energy Magazine, vol. 21, no. 5, pp. 38-47, Sept.-Oct. 2023, doi: 10.1109/MPE.2023.3288596.

[4] H. R. Massrur, M. Fotuhi-Firuzabad, P. Dehghanian and F. Blaabjerg, "Fog-Based Hierarchical Coordination of Residential Aggregators and Household Demand Response With Power Distribution Grids—Part II: Data Transmission Architecture and Case Studies," in IEEE Transactions on Power Systems, vol. 40, no. 1, pp. 99-112, Jan. 2025, doi: 10.1109/TPWRS.2024.3392018.

[5] Z. Xu, W. Jiang, J. Xu, Y. Deng, S. Cheng and J. Zhao, "A Power-Grid-Mapping Edge Computing Structure for Digital Distributed Distribution Networks," in IEEE Transactions on Smart Grid, vol. 15, no. 4, pp. 3432-3445, July 2024, doi: 10.1109/TSG.2023.3343373.

[6] Q. Jia, W. Jiao, S. Chen, Z. Yan and H. Sun, "A Trustworthy Cloud-Edge Collaboration Framework for Scheduling Distributed Energy Resources in Distribution Networks," in IEEE Transactions on Smart Grid, vol. 16, no. 3, pp. 2691-2694, May 2025, doi: 10.1109/TSG.2025.3545311.

[7] M. Sun et al., "BRM-Based Joint Resource Management for Demand-Side Communication in Smart Grid Considering Resource Coupling Effects," in IEEE Transactions on Smart Grid, vol. 16, no. 5, pp. 3817-3830, Sept. 2025, doi: 10.1109/TSG.2025.3576229.

[8] Z. Zhang, Y. Zhang, D. Yue, C. Dou, X. Ding and H. Zhang, "Economic-Driven Hierarchical Voltage Regulation of Incremental Distribution Networks: A Cloud-Edge Collaboration Based Perspective," in IEEE Transactions on Industrial Informatics, vol. 18, no. 3, pp. 1746-1757, March 2022, doi: 10.1109/TII.2021.3085670.

[9] W. Zhong et al., "Joint Energy-Computation Management for Electric Vehicles Under Coordination of Power Distribution Networks and Computing Power Networks," in IEEE Transactions on Smart Grid, vol. 16, no. 2, pp. 1549-1561, March 2025, doi: 10.1109/TSG.2024.3498945.

[10] H. Padullaparti et al., "Conservation Voltage Reduction with Distributed Energy Resource Management System, Grid-Edge, and Legacy Devices," 2023 IEEE Power & Energy Society General Meeting (PESGM), Orlando, FL, USA, 2023, pp. 1-5, doi: 10.1109/PESGM52003.2023.10252669.

[11] CIGRE Task Force C6.04.02, Benchmark Systems for Network Integration of Renewable and Distributed Energy Resources, Technical Brochure No. 575, CIGRE, Paris, 2014.

[12] L. Thurner, A. Scheidler, F. Schäfer, J. H. Menke, J. Dollichon, F. Meier, S. Meinecke, and M. Braun, "pandapower — An Open-Source Python Tool for Convenient Modeling, Analysis, and Optimization of Electric Power Systems," IEEE Transactions on Power Systems, vol. 33, no. 6, pp. 6510–6521, Nov. 2018, doi: 10.1109/TPWRS.2018.2829021.

[13] G. K. Roy, E. Gümrükcü, C. Joglekar, F. Ponci and A. Monti, "The impact of data latency on coordinated real-time charging management for electric vehicles," 2020 IEEE Power & Energy Society General Meeting (PESGM), Montreal, QC, Canada, 2020, pp. 1-5, doi: 10.1109/PESGM41954.2020.9282101.

979-8-3315-8138-1