ABIDI: A Reference Architecture for Reliable Industrial Internet of Things

Gianluca Rizzo[1,2](✉), Alberto Franzin[3], Miia Lillstrang[1,2,3,4,5,6,7], Guillermo del Campo[5], Moisés Silva-Muñoz[3], Lluc Bono[3], Mina Aghaei Dinani[1,2], Xiaoli Liu[6], Joonas Tuutijärvi[4], Satu Tamminen[4], Edgar Saavedra[5], Asuncion Santamaria[5], Xiang Su[4,7], and Juha Röning[4]

[1] HES SO, Valais, Sierre, Switzerland
{gianluca.rizzo,mina.dinani}@hevs.ch
[2] Universitá di Foggia, Foggia, Italy
[3] Université Libre de Bruxelles (ULB), Brussels, Belgium
{alberto.franzin,moises.silva-munoz,lluc.bono}@ulb.be
[4] University of Oulu, Oulu, Finland
{joonas.tuutijarvi,satu.tamminen,xiang.su,juha.roning}@oulu.fi
[5] Universidad Politecnica de Madrid (UPM), Madrid, Spain
{guillermo.campo,edgar.saavedra,asuncion.santamaria}@upm.es
[6] University of Helsinki, Helsinki, Finland
xiaoli.liu@helsinki.fi
[7] Norwegian University of Science and Technology, Gjøvik, Norway
xiang.su.su@ntnu.no

Abstract. The rationale behind the ever increasing combined adoption of Artificial Intelligence and Internet of Things (IoT) technologies in the industry lies in its potential for improving resource efficiency of the manufacturing process, reducing capital and operational expenditures while minimizing its carbon footprint. Nonetheless, the synergetic application of these technologies is hampered by several challenges related to the complexity, heterogeneity and dynamicity of industrial scenarios. Among these, a key issue is how to reliably deliver target levels of data quality and veracity, while effectively supporting a heterogeneous set of applications and services, ensuring scalability and adaptability in dynamic settings. In this paper we perform a first step towards addressing this issue. We outline ABIDI, an innovative and comprehensive Industrial IoT reference architecture, enabling context-aware and veracious data analytics, as well as automated knowledge discovery and reasoning. ABIDI is based on the dynamic selection of the most efficient IoT, networking and cloud/edge technologies for different scenarios, and on an edge layer that efficiently supports distributed learning, inference and decision making, enabling the development of real-time analysis, monitoring and prediction applications. We exemplify our approach on a smart building use case, outlining the key design and implementation steps which our architecture implies.

L. Barolli (Ed.): AINA 2023, LNNS 654, pp. 26–39, 2023.
https://doi.org/10.1007/978-3-031-28451-9_3

1 Introduction

In recent years, the automation of industrial processes has taken a step forward towards a more fine-grained control and actuation with the widespread adoption of technologies, such as Industrial Internet of Things (IIoT) and Artificial Intelligence, that propel what is being called the fourth industrial revolution, or Industry 4.0 [1,2]. The main idea underlying Industry 4.0 is to collect large amounts of data at every stage of the production process, and to exploit them to make automated decisions as informed as possible, in order to reach the production goals in the most efficient way, while reducing or eliminating the need for human intervention. Such approach opens up countless new challenges in IIoT. Among these, how to optimally deploy sensors in a complex industrial machinery, in order to detect variations in the state of the system and enable targeted, proactive interventions and maintenance; how to transmit IIoT data in a reliable and energy-efficient way; how to effectively address potential security and privacy issues of cloud computing; how to implement reliable and real-time distributed decision making, moving the computing load to the edge of the network and within IIoT systems; how to efficiently process IIoT data streams with high variety, volume, and velocity; and how to flexibly support a heterogeneous set of applications, services, prediction models and visualization tools that provide information to stakeholders. The sheer amount and heterogeneity of data available in large IIoT systems amplify these challenges, in terms of scalability and information integration.

Some of these issues can be addressed by introducing computing nodes physically close to where data is produced [3,4]. These devices, which form what is called the *edge* layer, allow shifting the computing load away from the cloud, reducing latency of computing tasks, relieving IIoT systems from much of the computing load due to data pre-processing, but also of more complex tasks such as anomaly detection, or training and execution of machine learning models.

The modularity of this approach and the distribution of the computing load has several advantages. Among these, it allows alleviating the burden on the centralized part of the infrastructure, in particular for time-sensitive applications. Moreover, it enables processing information closer to the source makes it possible to perform computations without transmitting sensitive information throughout the entire network. In general, moving the computation to the edge improves the computational performance and the communication latency and robustness [5]. Edge nodes can also add context to the data collected, thus enabling informed decisions pertaining to the part of the network to which they are connected.

However, this novel paradigm also introduces several new challenges. First, there is no clear consensus on how an heterogeneous edge-based architecture should be structured, in order to efficiently support the above mentioned services in an IIoT environment [6]. Edge nodes may be heterogeneous, and have limited resources, making scalability and efficient real time orchestration a key issue in real scenarios. The sensing, communication and computing infrastructure needs to be resilient to different types of faults and service disruptions. Thus, it must be designed and managed by taking into account reliability and service availability requirements, in order to deliver the target levels of service in case of hardware/software failures. New learning paradigms, and in general new decentralized algorithmic patterns, need to be developed and efficiently supported by the edge/cloud infrastructure to fully exploit the possibilities offered by the availability of large data streams. To the best of our knowledge, most IIoT architectures are not edge-based [7,8], and very few edge-based IoT architectures have been proposed so far. Debauche et al. propose an edge infrastructure to deploy microservices and AI applications at the edge layer, which is used for IoT applications in agriculture [9]. Guimarães et al. propose an edge-based IoT architecture to monitor industrial nodes [10]. These architectures are however tailored for a narrow, specific application domain, and though they demonstrate the potential of edge computing in IIoT, they do not specify how to generalize their approaches to other domains and applications.

To achieve the goal of designing a general edge-based IoT architecture, in this work we outline ABIDI, a framework for context-aware and veracious data analytics with automated knowledge discovery and reasoning for IIoT. The ABIDI framework encompasses the entire IIoT stack, from the devices to the edge, and to the cloud or central infrastructure, where the application performs the desired computation. The goal of this framework is to enable the efficient and reliable collection of data and the development of AI applications that can be seamlessly deployed on a variety of IIoT scenarios. This is achieved by designing an IIoT architecture whose efficiency depends on both the integration between its modules and the optimization within each module.

In particular, we enable improvements of network performance and reliability by designing a methodology to select the best communication technologies in different contexts, and by proposing an IIoT network architecture which allows reducing latency and energy consumption while easing integration with upper layers. We propose an edge architecture that enables the AI-based IIoT systems, distributing the computation between the cloud/central infrastructure and edge nodes transparently to the application developers. We introduce new privacy-preserving, fully distributed and scalable learning schemes which do not need any parameter server and benefit from node mobility. We further develop visualization tools for data quality assessment that provide insight on the structure, contextual properties and dependencies present in the data streams and thus assist in the development of case dependent pre-processing methods, and we implement energy load prediction models for real world use cases.

The paper is organized as follows. In Sect. 2 we outline the architecture of our framework, we describe our approaches to the implementation of its main functional components. In Sect. 3 we present an application of our framework to a real world case. Finally, Sect. 4 discusses some of the key open research issues that our approach implies, and Sect. 5 presents our main conclusions.

2 The ABIDI Framework

A schematic representation of the overall architecture of the ABIDI framework is presented in Fig. 1. It is divided into three main layers: 1) the *IoT layer*, encompassing the IoT devices and the communication network; 2) the *edge layer*, providing low-latency decision making for IIoT devices and end user devices; and 3) the *cloud layer*, composed by a big data processing level, a data analysis level focusing on prediction of future events and patterns, and an application level.

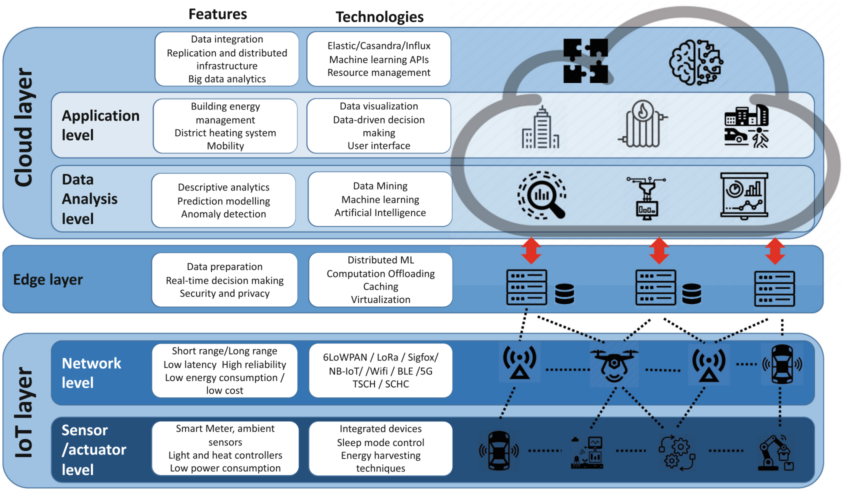

Fig. 1. A high-level representation of the architecture of the ABIDI framework.

IoT Layer. A first key component of the ABIDI framework at the IoT layer is a methodology for the selection of the most appropriate wireless communication technology (in terms of resource efficiency, but also of reliability and QoS support) for each use case or final application. Another important element of the ABIDI IoT layer is the use of energy harvesting techniques to power IIoT devices, taking advantage of the many energy sources typically available at industrial facilities.

Edge Layer. It is typically composed by an heterogeneous set of autonomous computing and communication devices, such as gateways, industry robots, wireless access points and cellular base stations. This layer is responsible for several functionalities related to *data quality*: 1) collection, aggregation and contextualization of the data coming from IIoT environments; 2) aggregation/real time monitoring and collection of metrics about data quality, such as data integrity, consistency, accuracy, completeness, validity, uniqueness and timeliness; 3) data creation (e.g. auto-filling values in forms, automatic extraction of data) and data enrichment; 4) data maintenance (reactive: data correction; proactive: business rules) and data unification (matching and deduplication); and 5) (in synergy with cloud) data protection (e.g. identification of sensitive data, detection of fraudulent behavior) and data retirement (end of life).

Given the tight resource constraints of IoT devices, another key role of the edge layer is the implementation of computation offloading services. Offloading computation (and power consumption) intensive tasks to the edge enables faster decision making of applications running at the edge (thus improving capability to handle latency-sensitive applications), and it saves energy in IoT devices, extending their lifetime.

With respect to the IIoT layer, edge devices implement IIoT and network coordination functions in a self-organized and autonomous manner. This includes enabling IoT integration by acting as gateways for local IoT systems, and joint management of IoT, network and edge resources. Such coordinated control has as its main goals to enable the delivery of the QoS required by the different verticals and applications (such as the support for latency-sensitive applications), and to implement reactive (and possibly proactive) schemes for ensuring service continuity in case of disruptions.

For what concerns the use of ML and data intensive strategies (for the implementation of ABIDI platform applications as well as for the management of the platform itself) the edge layer plays a double role. On one side, it implements mechanisms for model training which are close to data and thus resource efficient and context aware. In addition, it executes local machine learning prediction models. With this respect, one of the key roles of the edge layer is to enable the implementation of learning architectures which are able to provide high levels of data security and privacy preservation, of scalability (with respect to both participating systems and of applications) and of resiliency to infrastructure failures. Indeed these features are critical in present day IIoT scenarios in which data (as well as computing resources) are spread across an ever growing number of heterogeneous devices, and in which harnessing locally available devices, even in an opportunistic manner, is key to achieve high levels of QoS (e.g. in terms of latency of computing tasks) in a resource efficient manner.

To perform efficient inference and learning at the edge, the ABIDI architecture is designed to enable the communication not only of data, but also of models and computational tasks. This increases the overall efficiency of the infrastructure, by distributing the computation in an organic manner in the edge layer, and between the centralized infrastructure and the edge. For example, in a classical IoT network, sensors collect data and transmit it to the central server, which is in charge of all the computation. In an edge infrastructure, the intermediate layers can instead manage part of the operations, such as aggregating data or spotting malfunctioning devices, transmitting to the central server only the correct, aggregated information. Edge nodes can therefore relieve the central server of unnecessary operations, making local decisions. This paradigm brings clear advantages in terms of computational and transmission speed.

Cloud Layer. The data collected by the IIoT devices and potentially the results of the elaboration at the edge level are transferred to the cloud layer. The ABIDI infrastructure relies on suitable database technologies to collect the data. Different applications may require different databases, or a pre-existing infrastructure could be integrated in the ABIDI architecture. Regardless of how this infrastructure is defined and which hardware and software are used, the storage of the data collected remains a potential bottleneck in any data-centric pipeline. Therefore, the ABIDI architecture adopts a flexible data infrastructure that can be optimized for different tasks.

The data analysis level from the cloud layer includes automated and semi-automated data cleaning, data visualization tools, and machine learning for predictive and descriptive modeling. The successful operations of final applications, such as evidence-based decision support tools, depends on the quality of the data, such as their timeliness and reliability. In ABIDI architecture, automated data cleaning methods are applied to solve any data quality deficiencies that are relatively simple to treat, and to perform basic fault detection procedures. The adoption of automated methods, when they are reliable, allows minimizing human effort, which is crucial when operating with big data. As a solution for optimizing between reliability and human effort, semi-supervised methods are applied in cases that cannot be reliably solved using automated methods.

In the data analysis level, interactive exploratory data visualization tools are utilized to enable effortless monitoring and inspection of the big data and of the data quality. The visualization tool prepares the developers of automated data processing system to improve the quality of their data to meet the contextual requirements, to reflect the needs of decision-making process and to allow providing domain specific answers to the user. Through an effective visualization, the massive amount of data becomes accessible and understandable, which makes it possible to both ensure the appropriateness of the automated pre-processing steps and to add use case dependent methods above the automated ones. The combined application of these two approaches allows achieving high quality standards for IIoT data, particularly in those application contexts where it is often plagued by noise, or where it is often incomplete and inaccurate.

Machine learning regression is applied in the analysis level for descriptive modeling and for prediction of IIoT data streams. The descriptive models estimate the value of a data variable at a certain moment, and the estimations are useful for missing value imputation and anomaly detection. Predictive regression models differ from the descriptive ones in that they estimate values of the variables at a future moment. The predictions can offer substantial profits when utilized in decision support tools. The ABIDI architecture includes a full automated pipeline for creating baseline regression models for time series prediction.

Technology Selection in IIoT Network. Although in industrial environments, traditionally, assets have been connected using wired communication technologies (based on Field-bus or Industry Ethernet), recent advances on wireless communications have enabled the access to new elements and data, providing advantages in terms of flexibility, mobility, installation, and cost, among others [11]. While wireless sensor networks (WSNs) have been largely used in building automation, smart city or agriculture domains, the industrial environments differ from these due to their particular constraints, especially in terms of latency, environment, heterogeneity and mobility [12]. There are many wireless communication technologies and protocols that may be named as Industrial IoT networks [13]. Regarding existing literature that presents technical features, existing deployments, and future trends, the ABIDI framework considers following IoT network technologies as the most relevant: BLE, ZigBee, WiFi, WirelessHart, LoRaWAN, Sigfox, 6LoWPAN, NB-IoT, LTE Cat-M1, and 5G.

Although there are many surveys and reviews on IIoT networks, such as [14], few studies have considered factors beyond technical parameters, including the constraints of factories environments and its integration with the other layers of the IIoT architecture [15]. Through reviewing literature and technical specifications, Table 1, which summarizes the main parameters of each technology, has been created to assist technology selection. As it can be observed, the different IIoT network technologies have their strengths and weaknesses, and therefore cannot comply with all the requirements of every use case or application.

The ABIDI framework is based on a two-step procedure for selecting the appropriate communication technology for a specific use case or application, as follows.

1. Determine the essential use case specific requirements set by the final application. These requirements may be divided in the following categories:
   - **Technical factors:** They include technical characteristics such as the transmission capacity (data rate), the time taken from the instant the node transmits the message until it arrives to the final application (latency), the communication coverage (range), the bi-directionality (duplex) and the loss of messages (reliability).
   - **Implementation factors:** They integrate those factors especially relevant during the IoT network implementation phase. The most important one is cost, which is the sum of the cost of IoT devices and nodes plus the cost of network infrastructures (for those technologies that demand the deployment of private network elements, such as 6LoWPAN, Zigbee, WiFi or LoRa), or the cost of data plans (for those technologies that provide the network infrastructure, such as Sigfox, 5G, or NB-IoT).
   - **Functional factors:** They cover factors that affect everyday working of IoT applications, including the autonomy of the devices (energy consumption), which is determined by the time the IoT device is turned on and especially by the energy consumption during the communication process.
2. Compare these requirements with Table 1, and select the most suitable technology. This step is implemented via Machine Learning based algorithms, which recommend the best communication technology based on use case requirements, and on all system constraints.

Table 1. Summary of the main parameters considered in ABIDI methodology for IIoT technology selection.

Technology	Data rate[a]	Latency[b]	Range[c]	Duplex	Reliability	Consumption	Cost
BLE	Mbps	30 ms	100 m	half	low	low	low
ZigBee	kbps	40 ms	100 m	half	high	low	low
WiFi	Mbps	30 ms	100 m	half	med	med	low
WirelessHart	kbps	10 ms	200 m	half	high	med	high
LoRaWAN	kbps	300 ms	10 km	half	med	med	med
Sigfox	bps	4 s	50 km	limited[d]	high	high	med
6LoWPAN	kbps	20 ms	100 m	half	med	low	low
NB-IoT	kbps	2 s	10 km	half	high	high	high
LTE Cat-M1	kbps	2 s	10 km	half	high	high	high
5G	Gbps	10 ms	10 km	half	high	high	high

[a, b, c] Approximate values—in the order of magnitude.
[d] Sigfox provides limited bidirectional capacity: the IoT device can upload up to 140 12-byte messages a day, but it can only receive four 8-byte messages.

3 A Building Management Use Case

In order to assess the ABIDI framework, we implemented it in a smart building testbed at CEDINT-UPM in Madrid, Spain, a three-story construction that hosts offices, research labs, and other facilities. It is equipped with 30 IoT power meter devices are installed at panel boards, allowing specific energy consumption monitoring of 560 electrical lines; 40 IoT ambient sensor devices measuring temperature, luminosity, humidity and presence detection—apart from battery level; and 30 HVAC controllers, which provide set-point temperature, fan speed, working mode (cold/heat), state (on/off) and indoor temperature data. By means of an Elastic Stack-based IoT Platform, data collected were distributed and replicated to provide inputs for machine learning (ML) and visualization tools. The two main goals have been: i) optimizing energy consumption by context-aware data analytics of energy consumption patterns, taking into account energy measurements, ambient parameters and user behaviour; and ii) ensuring data reliability and veracity, by improving communications, and detecting and correcting missing or wrong measurements.

We applied the ABIDI framework methodology to select the optimal communication technology. The main technical requirements were low data rate, medium reliability, non-critical latency, variable sending frequency (30 s–15 min), and bi-directionality. To this end, we performed an experimental characterization for communication reliability and energy consumption.

Experimental results for reliability (latency and error rate) of the different technologies were obtained using an ad-hoc testbed (Table 2). Latency was measured considering an end-to-end trip, from the Industrial IoT node to the application server. For error rate, the same latency packets were used. Based on these results, 6LoWPAN outperformed alternative protocols with regards to communication latency.

Table 2. Experimental latency, error rate and consumption results

Technology	6LoWPAN	LoRaWAN	Sigfox	BLE	WiFi
Latency (ms)	20	290	3700	26	32
Error Rate (%)	0.01	0.6	0	0.03	0
Tx. Consumption (mAs)	0.8	6.3	804.8	1.0	3.4

As a second step, we experimentally measured the energy consumption of the transmission process at 5 V (Table 2). These measurements were taken using a Nordic Semiconductor Power Profiler Kit II. Then, the transmission current demand per se was integrated during the time of packet transmission. Regarding power consumption, 6LoWPAN outperformed again the other technologies, especially Sigfox, which was expected to have a greater consumption as its on-air time is much longer.

Bottom line, considering the number on IoT nodes (100) and area of deployment (50 m × 40 m), the variable sending frequency and the non-restrictive requirements in terms of latency an reliability, BLE and 6LoWPAN seemed to be the best choices. However, the features of 6LoWPAN mesh topology, which enables the utilization of a single network coordinator or access point for the entire use case (together with the fact that it implements IPv6 connectivity, allowing direct access from the Internet), made 6LoWPAN the final choice. To infer energy consumption patterns, we have combined temperature measurements of indoor ambient sensors and the HVAC energy consumption measured with BatMeter smart meters, as the latter data sources alone proved insufficient.

A baseline XGBoost regression model was built for short-term (one hour ahead) HVAC energy consumption prediction. Error metrics CV-RMSE, Rel-RMSE and MASE for the model were 0.292, 0.811, and 0.870, respectively. Rel-RMSE and MASE measures include a built-in comparison to a naïve time series prediction model, and the value being less than one indicates the model is performing better than the naïve model. This showed that the suggested baseline model is capable of providing useful outputs in short-term predictions.

In data pre-processing phase, it was possible to automatically detect a malfunctioning sensor in the monitored area by inspecting the rate of data packages sent by each sensor node. Semi-supervised methods, where IoT data streams were combined with relevant metadata, allowed imputing missing temperature data by utilizing peer sensors in the room. While a reasonably light approach was enough to meet the data quality requirements of the desired application, a more thorough visual interface was also developed for this setup (Fig. 2) for ease of inspection of data streams.

The layout of the final application is in Fig. 3. The quality of data was ensured for each of the five parameters utilized in the final application. Gradients of temperature data sources were inspected together with the gradient of the electricity consumption data, so as to label times of HVAC usage in every room. The application layout shows a figure of each data stream, highlights the times of HVAC usage in each temperature figure, visualizes the number of active HVAC

Table 3. Basic information of the UPM data set

Indoor ambient sensors	Number of sensors	34
	Sensors per space	1–4
	Parameters	Temperature, humidity, light, motion
Smart meters	Number	32
	Parameters	Power consumption (528 lines)
Time range of data collection	Start	In steps through 2018–2020
	End	Ongoing (12/2021)
Frequency of data collection		From seconds to an hour

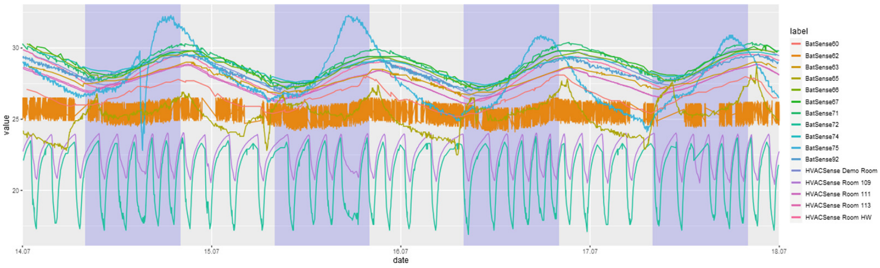

Fig. 2. Application screenshot, detecting HVAC usage in the UPM building, displaying all of the relevant data streams.

units as a function of time, and provides a summary of the estimated electricity consumption per room for a given period of time. These data allowed determine new opportunities for optimization of power consumption. In particular, the fact that the highest peaks in power consumption were caused by HVAC units being turned on simultaneously in multiple rooms suggest that smart scheduling of HVAC duty cycles could substantially reduce these peaks, and thus contribute to preventing outages (Table 3).

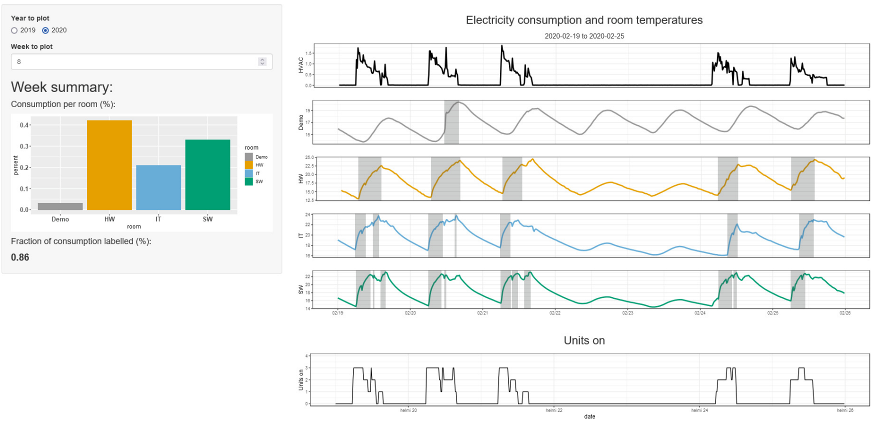

Fig. 3. Screenshot of the visualization tool, detecting HVAC usage in UPM building.

4 Key Research Challenges

We discuss some technical challenges for full implementation of the ABIDI approach, including IIoT architecture and protocols, energy harvesting, network optimization, Cloud infrastructure optimization, and Application layer. It is crucial to identify and analyze those challenges for seeking novel solutions.

Reliable IIoT Architecture and Protocols. In order to increase performance and reliability for IIoT networks, and meet with the most-demanding communication requirements (e.g. robot control), some leading-edge techniques must be implemented at various levels of the architecture. Specifically, MAC layer enhancements such as Time Slotted Channel Hopping (TSCH) could be used in 6LoWPAN. TSCH avoids packet losses and reduces latency by dynamically changing the carrier frequency in a globally synced mesh network among all the nodes in the network [16]. On the other hand, for other communication technologies, such as LoRaWAN or NB-IoT, the scheme of Static Context Header Compression (SCHC) could be implemented. SCHC allows compression of IPv6/UDP/CoAP packets, with the aim of making them suitable for transmission over their restricted links of these technologies and providing higher interoperability by using IPv6 connectivity [17].

Resource Optimization of ML Training at the Edge. One of the key open issues in gossip learning lies in the lack of understanding of the relationship between patterns of exchange of models and of movement of agents, and some of the primary performance parameters of the scheme. A key challenge concerns how to optimally tune model merging as a function of the context and of the specific problem. Different merging strategies have shown to perform very differently according to the specific model, but also as a function of the degree of dynamicity of the environment. New approaches need to be designed in order to improve their efficiency in heterogeneous settings, i.e. when applied to set of nodes with very diverse sensing and computing capabilities. Finally, strategies for improving the communication efficiency of these schemes have to be designed, and the trade-off between performance and resource efficiency has to be characterized.

Energy Harvesting. As already mentioned, the location of IoT devices within manufacturing equipment and processes means that they have to be battery-powered. Energy harvesting (EH) rises as a green, sustainable, and virtually infinite power supply to wireless devices, obtaining the available energy from the environment to reduce the need for storage components. Power generation density depends mainly on the real characteristics of the ambient energy availability for the IoT device location. Even if RF appears to be a common energy source provided by manufacturing equipment and existing wireless communications, its power density is small compared to other energy sources such as light or magnetic induction. A more in-depth analysis of the power density of the diverse energy harvesting techniques in factories is needed. Another deciding factor is the availability of the energy source, which may be steady (RF) or more unpredictable (light), affecting the power supply profile. Time variation of the energy sources should be characterized. Finally, a recent trend of study is the use of hybrid energy harvesting schemes, combining high-power-density techniques (PV) with more steady sources (RF).

Edge/Cloud Balance-Network Optimization. The flexibility of the ABIDI architecture at the edge layer allows it to adaptively distribute the computation. For this, the particular application deployed at the edge level will be containerized providing the architecture with more flexibility at the edge layer. This containerization provides the edge-layer with the option to dynamically adapt the computational resources by using an (intra-)edge layer load balancing mechanism such as Kubernetes. The flexibility of the architecture will then be complemented with an inter-layer load-balancing mechanism which allows the edge layer to offload tasks to the server infrastructure. To this end, appropriate load balancing mechanisms need to be designed, capable of efficiently cooperate with the data-offloading and task balancing processes.

Cloud Infrastructure Optimization. On the cloud side, for each use case the database used must be tailored to the specific needs to optimize its performance in terms of throughput. This is better done when the scenario characteristics in terms of data and operations are fully determined, to obtain the database configuration that best serves the application. This approach can be also applied to entire software pipelines, such as when Kafka is employed to transmit data from the edge nodes to the database on the cloud, to optimize every step of the data collection process. The configuration methodology remains the same, requiring only to define the interface between Irace and the desired database/pipeline [18].

Application Layer. Turning the current approach taken for improving data quality in cloud environment into a full, low-effort pipeline applicable to a wide range of use cases is a key challenge. A full pipeline from data to decision support tool visualization has currently only been implemented for time series regression models, and expanding to other kind of tasks, such as classification, is important to widen the spectrum of covered use cases. Utilizing Bayesian Estimation or some other suitable algorithm for hyperparameter tuning instead of using a grid search could also improve computational efficiency of the process.

5 Conclusions

Technologies such as artificial intelligence and Internet of Things are reshaping industrial processes to the point that relevant actors are calling this transformation the fourth industrial revolution. The combination of big data and automated decision making is helping companies in transitioning from general mass production to a smart production that uses information to increase efficiency and reduce waste and operational costs. This transformation does not come without challenges, since current approaches are limited in scope and application.

In this work we have presented the ABIDI architecture for Industrial Internet of Things. ABIDI is a general framework that can be instantiated to address different real world cases, making use of the most suitable technologies for each scenario. It encompasses the whole IIoT stack, from the sensors and network layer to the final application, combining the use of cloud architectures with an edge layer of computational nodes that can improve the performance and robustness of the final application, and can perform distributed AI tasks. We have discussed how the components of our architecture address the shortcomings of the current state of the art. Finally, we have reported a real world scenario where we instantiated our architecture, and we have outlined the steps necessary to reach the full vision of the ABIDI infrastructure.

Acknowledgment. This work has been supported by the CHIST-ERA project CHIST-ERA-17-BDSI-001 ABIDI "Context-aware and Veracious Big Data Analytics for Industrial IoT". This work has been partially supported by COST INTERACT, the FARI Institute, and by SNF Dymonet project. AF is supported by Service Public de Wallonie Recherche under grant n° 2010235 - ARIAC by DIGITALWALLONIA4.AI. Published with a contribution from 5 × 1000 IRPEF funds in favour of the University of Foggia, in memory of Gianluca Montel.

References

1. Tao, F., Qi, Q., Liu, A., Kusiak, A.: Data-driven smart manufacturing. J. Manuf. Syst. **48**, 157–169 (2018)
2. Sisinni, E., Saifullah, A., Han, S., Jennehag, U., Gidlund, M.: Industrial internet of things: challenges, opportunities, and directions. IEEE Trans. Ind. Inf. **14**(11), 4724–4734 (2018)
3. Shi, W., Cao, J., Zhang, Q., Li, Y., Xu, L.: Edge computing: vision and challenges. IEEE Internet Things J. **3**(5), 637–646 (2016)
4. Qiu, T., Chi, J., Zhou, X., Ning, Z., Atiquzzaman, M., Wu, D.O.: Edge computing in industrial internet of things: architecture, advances and challenges. IEEE Commun. Surv. Tutor. **22**(4), 2462–2488 (2020)
5. Ejaz, M., Kumar, T., Ylianttila, M., Harjula, E.: Performance and efficiency optimization of multi-layer IoT edge architecture. In: 2nd 6G Wireless Summit (6G SUMMIT), pp. 1–5. IEEE (2020)
6. Sittón-Candanedo, I., Alonso, R.S., Rodríguez-González, S., García Coria, J.A., De La Prieta, F.: Edge computing architectures in industry 4.0: a general survey and comparison. In: Martínez Álvarez, F., Troncoso Lora, A., Sáez Muñoz, J.A., Quintián, H., Corchado, E. (eds.) SOCO 2019. AISC, vol. 950, pp. 121–131. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-20055-8_12
7. Boyes, H., Hallaq, B., Cunningham, J., Watson, T.: The industrial internet of things (IIoT): an analysis framework. Comput. Ind. **101**, 1–12 (2018)
8. Sobin, C.: A survey on architecture, protocols and challenges in IoT. Wirel. Pers. Commun. **112**(3), 1383–1429 (2020)
9. Debauche, O., Mahmoudi, S., Mahmoudi, S.A., Manneback, P., Lebeau, F.: A new edge architecture for AI-IoT services deployment. Procedia Comput. Sci. **175**, 10–19 (2020)
10. Guimarães, C.S.S., Jr., de Andrade, M., De Avila, F.R., Gomes, V.E.D.O., Nardelli, V.C.: IoT architecture for interoperability and monitoring of industrial nodes. Procedia Manuf. **52**, 313–318 (2020)

11. Gungor, V.C., Hancke, G.P.: Industrial wireless sensor networks: challenges, design principles, and technical approaches. IEEE Trans. Ind. Electron. **56**(10), 4258–4265 (2009)
12. Catenazzo, D., O'Flynn, B., Walsh, M.J.: On the use of wireless sensor networks in preventative maintenance for industry 4.0. In: 2018 12th International Conference on Sensing Technology (ICST), pp. 256–262 (2018)
13. Li, X., Li, D., Wan, J., Vasilakos, A.V., Lai, C.-F., Wang, S.: A review of industrial wireless networks in the context of industry 4.0. Wirel. Netw. **23**(1), 23–41 (2015)
14. Raza, S., Faheem, M., Guenes, M.: Industrial wireless sensor and actuator networks in industry 4.0: exploring requirements, protocols, and challenges-a MAC survey. Int. J. Commun. Syst. **32**(15), e4074 (2019)
15. Liu, Y., Kashef, M., Lee, K.B., Benmohamed, L., Candell, R.: Wireless network design for emerging IIoT applications: reference framework and use cases. Proc. IEEE **107**(6), 1166–1192 (2019)
16. Urke, A.R., Kure, Ø., Øvsthus, K.: A survey of 802.15.4 TSCH schedulers for a standardized industrial internet of things. Sensors **22**(1) (2022)
17. Sanchez-Gomez, J., Gallego-Madrid, J., Sanchez-Iborra, R., Santa, J., Skarmeta, A.F.: Impact of SCHC compression and fragmentation in LPWAN: a case study with LoRaWAN. Sensors **20**(1) (2020)
18. Silva-Muñoz, M., Franzin, A., Bersini, H.: Automatic configuration of the Cassandra database using Irace. PeerJ Comput. Sci. **7**, e634 (2021)