XXIV IMEKO World Congress "Think Metrology"
August 26 – 29, 2024, Hamburg, Germany

AUTOMATION IN SENSOR NETWORK METROLOGY: AN OVERVIEW OF METHODS AND THEIR IMPLEMENTATIONS

Anupam Prasad Vedurmudi [a *], *Kruno Miličević* [b], *Gertjan Kok* [c], *Bang Xiang Yong* [d], *Liming Xu* [d], *Ge Zheng* [d], *Alexandra Brintrup* [d], *Maximilian Gruber* [a], *Shahin Tabandeh* [e], *Martha Arbayani Zaidan* [f], *André Xhonneux* [g], *Jonathan Pearce* [h]

[a] Physikalisch-Technische Bundesanstalt, Abbestr. 2-12, 10587 Berlin, Germany
[b] Random Red, Ulica Ivana Zajca 40, Osijek, Croatia
[c] Van Swinden Laboratory (VSL), Delft, Thijsseweg 11, 2629 JA, the Netherlands
[d] University of Cambridge, 17 Charles Babbage Road, Cambridge, CB3 0FS, United Kingdom
* Corresponding author, email address: anupam.vedurmudi@ptb.de

Abstract **–** Sensor networks are an integral component of the ongoing automation of industrial processes in a diverse range of sectors. As sensors and, by extension, sensor networks provide information about physical quantities in the form of measurements, the development and adaptation of metrological practices that ensure the reliability, accuracy, and traceability of the data thus generated is essential. A complementary development of tools for the implementation of metrological methods is necessary. In this contribution we present a review of the tools and methods relevant to the automated application of metrological practices to large-scale transient sensor networks with an emphasis on uncertainty aware soft- and middleware, data fusion and machine learning. In this review, we will discuss the state-of-the-art with respect to general metrological methods and specific soft- and middleware tools and motivate future developments in sensor network metrology.

Keywords**:** sensor networks; data fusion; machine learning; agent-based systems; metrology

1. INTRODUCTION

The deployment of interconnected sensors in large numbers as part of both homogeneous, i.e. measuring the same physical quantity, as well as heterogenous networks is a natural way to extend the utility of sensors and is of increasing importance in several domains. As sensor networks effectively function as distributed measurement systems, the development of metrological practices tailored to sensor networks is essential. Moreover, automation in both the application of the developed methods as well as in the analysis and processing of the typically large amount of data generated is essential to the functioning of large-scale sensor networks. In many real-world scenarios, transience, or changes in the network properties with time adds an additional layer of complexity. In this context, we aim to review tools and methods relevant to the automated application of metrological practices to large-scale transient sensor networks (i.e., changes of topology, availability, or placement of sensors in the network over time) with an emphasis on metrological conformity and uncertainty awareness in data evaluation, soft- and middleware, data fusion and machine learning.

The review is divided into two sections. In the first, we focus on general methods relevant to sensor networks. In particular, we focus on the calibration using hybrid models as well as in-situ and co-calibration, i.e. a calibration based on measurement values of available sensors. Thereafter, we will focus on data aggregation in sensor networks and uncertainty-aware sensor fusion as a means to estimate physical quantities that are not amenable to direct measurement. Subsequently, we will provide a discussion of a key aspect of modern industrial processes – the leveraging of redundant information to obtain more accurate or robust estimates of observed quantities using data aggregation and fusion. Both model-based and deep learning methods will be covered based on their suitability to exploit the large amount of data generated by sensor networks in a metrologically consistent manner. Specific methods relevant to large-scale transient sensor networks such as online and federated learning as well as using redundancy to reduce measurement uncertainty will constitute this portion of the review. The section will be concluded with a discussion of the use of multi-agent systems in sensor networks – particularly their architecture and the advantages they provide.

The second section will cover specific soft- and middleware tools relevant to sensor network metrology. Given the large number of commercial and open-source software packages and toolboxes for uncertainty evaluation, a thorough discussion on its automated implementation in sensor networks will be provided. In particular, the evaluation of dynamic measurement uncertainty and its real-time implementation in sensor networks will be reviewed. Subsequently, we will discuss the use of agentMET4FOF, a Python-based software framework to set up asynchronous agents with common features for distributed data streaming processes. General aspects regarding agent-architectures and the influence of network topology will inform the discussion. A concurrent review of methods relevant to distributed sensor networks will also be conducted. The middleware aspects of sensor networks will be covered based on the example of FIWARE – an open-source framework that provides a standardized approach to managing context information in a cross-domain and interoperable manner. The implementation of relevant protocols in ensuring interoperability will be covered simultaneously.

2. REVIEW: METHODS

In this section we cover recent developments in methods relevant to sensor networks with the aim of characterizing the state-of-the-art and motivating future research in sensor network metrology.

2.1 In-situ calibration and co-calibration

Massive sensor network deployments, though now within economical and technical reach, present challenges with regard to their maintenance and reliability. In particular, reaching and then maintaining the targeted quality of measurements throughout the deployment duration is an important issue. Although it would be highly relevant given their tendency to drift due to premature aging, factory calibration is often prohibitively expensive for systematic application to low-cost sensors. Furthermore, there are concerns about the applicability of factory calibration to field conditions. These challenges have in turn fostered a considerable amount of research on in-situ calibration. In-situ calibration refers to the characterization of the measurement model of a deployed sensor and its uncertainty, i.e., its calibration at the location of its deployment without having to disassemble and transport it to a calibration laboratory or factory [1]. In other words, sensors are calibrated at their deployment location, ideally without physical intervention by leveraging their communication capabilities. As a result, the calibration of entire sensor networks is facilitated in cases where the calibration of individual sensors that are deeply integrated in a controlled environment is not feasible.

In sensor networks, a co-calibration is a potential way to carry out an *in situ* calibration by using measurements from nearby sensors [2] along with appropriate interpolation and sensor fusion techniques. A common approach in this regard is the use of macro-calibration techniques, where the response of the entire network is optimized by estimating the relevant optimal parameters [3]. In the case of a blind calibration [4], i.e., in the absence of controlled reference values, a dense deployment of sensors is advantageous as nearby sensors measuring the same quantity can be assumed to have nearly identical responses. However, by exploiting the correlations between the sensors, a blind calibration can be achieved even in the absence of a dense deployment [5]. In-field calibration transfer [6] compares distributions of measured values over time instead of the measurements themselves, making it partially independent from their actual samples and timestamps. Furthermore, advanced machine learning approaches, such as nonlinear autoregressive exogenous model (NARX) and long short-term memory (LSTM) models were developed using data sets obtained from two reference stations in a city to calibrate low-cost air quality sensors [7]. To apply a machine learning approach to the data collected from sensor network nodes, the calibration operation is reformulated as a supervised learning problem, which allows to incorporate customised loss functions that particularly focus on a segment of the signal. As a result, the calibration approaches achieved generalization where the calibration models worked accurately in both separated testing sites.

2.2 Calibration of hybrid models

Hybrid model calibration is a method that harnesses the advantages of two different approaches. This might involve merging two distinct data-driven models or combining laboratory with field calibration methods. A general hybrid calibration model for air quality using low-cost sensors was provided by [8]. The model was designed by combining the strengths of random forest regression, i.e., its ability to capture complicated nonlinear relationships between various inputs and the target output, with the ability of a simple linear model to extrapolate beyond the set of data on which the model is trained. The results showed that the calibration based on hybrid models tend to generalize best for NO, NO_2, and O3 when applied to data collected at new sites. In addition, the Enhanced Ambient Sensing Environment (EASE) method was proposed as a hybrid method by combining the advantages of a laboratory calibration with the increased accuracy of a field calibration for calibrating low-cost gas sensors, such as NO2 and O3 [9]. A hybrid calibration model was also developed to calibrate low-cost air quality sensor networks in the presence of concept drift. This was achieved by combining batch machine learning algorithms and regularly updated online machine learning calibration function(s) for the whole network when a small number of reference instruments are present [10].

2.3 Aggregation of data in sensor networks

A key aspect of modern industrial processes is the collection of large amounts of data from multiple sensors. There is usually a large amount of redundancy in the data, and to reduce the amount of data that has to be transmitted over the network, data aggregation techniques come into play. Data aggregation can be defined more precisely as a creative process that collects data from various sensors and IoT devices and then integrates them using an aggregation function to minimize the injected traffic into the system [19]. The main motivation for using data aggregation techniques is that, while it is cheap to measure quantities with low-cost sensors, the primary challenge in sensor networks execution lies in moving and storing the data from the source to the sink nodes. Hence, energy-efficient routing protocols have been developed to aggregate the data such as the Low Energy Adaptive Clustering Hierarchy (LEACH) [11], Power Efficient Gathering Sensor Information Systems (PEGASIS) [12], Cluster-Chain Mobile Agent Routing (CCMAR) [13], and Max-Sum algorithm [14]. Although these protocols are geared towards improving energy efficiency in wireless sensor networks, they differ mainly in their structural approach, role of nodes and computation algorithms.

2.3.1 Uncertainty aware sensor fusion for drift detection

In order to leverage aggregate information and obtain a more robust estimate of an observed quantity, sensor fusion operations are typically employed [15]. In other words, measurements from disparate sensors can be combined to generate a new value based on a particular mathematical model or algorithm to generate a new “fused” measurement. The resulting value typically cannot be directly measured by the constituent sensors. In the context of metrology, the key challenges here are to ensure compliance with quality requirements and to preserve the traceability of the fused value. Moreover, the sensor fusion needs to be implemented within suitable digital architectures for Industrial Internet of Things (IIoT) environments. In [16] a method for sensor aggregation and fusion that reduces the uncertainty of the derived value was proposed. The method can also be used to detect drifting sensors and is an extension of the classical method of using a chi-squared test for detecting outliers, first proposed in [17]. In [18] metrics for quantifying redundancy were defined. A changing value of such a redundancy metric can indicate a drifting sensor. Some of these methods were implemented in the Met4FoF framework so that they can be used for automated uncertainty evaluation. In addition, failure and anomaly detection based on an adaptive Weibull distribution was developed to identify sensors that drift and generate anomalous data patterns in comparison to the measurements of reference instruments [19]. The method showed promising results after evaluating it on a dense air quality sensor network consisting of 126 low-cost sensors distributed in a city. Furthermore, a parallel calibration method based on white and black box Bayesian calibration models was developed for performing low-cost sensor calibration to cope with extreme events [20]. In the study, the uncertainty around Bayesian mean estimation also enables hidden sensor drifts to be detected. A method for uncertainty aware sensor aggregation and fusion was proposed in [16]. It was based on weighing the sensor data with their respective uncertainties. In addition, a linear transformation between measured sensor data and the measurand of interest was considered. Some of these methods were implemented in the agentMET4FOF framework[1] so that they can be used for automated uncertainty evaluation. Furthermore, a modular approach towards homogeneous sensor-fusion using digital twins to represent the entities of two separate IIoT testbeds was presented in [21].

2.4 Machine learning

The use of advanced machine learning (ML) methods is another key aspect of the Industrial Internet of Things, or IIoT, paradigm [22]. In contrast to classical ML algorithms like linear regression and support vector machines, deep learning methods are particularly well suited to exploit the large datasets generated by such systems [23] as well as their inherent nonlinearity [24]. For instance, a common application is the use of soft sensors based on machine learning to generate sophisticated measurements using data from a few sensors [25]. With sufficiently large datasets, deep learning based soft sensors can model nonlinear measurement models better than traditional methods [26]. Uncertainty propagation in this context refers to the determination of the output of the Machine Learning (ML) algorithm given the uncertainty of the inputs. For instance, the unscented transform is a means to efficiently propagate the probabilistic first and second moments of the input distributions through nonlinear models like neural networks [27], [28]. Furthermore, advanced machine learning approaches, such as nonlinear autoregressive exogenous model (NARX) and long short-term memory (LSTM) models were developed using data sets obtained from two reference stations in a city to calibrate low-cost air quality sensors [7]. The calibration approaches achieved generalization where the calibration models worked accurately in both separated testing sites.

2.4.1 Online machine learning

Online machine learning (also known as incremental or out-of-core learning) refers to a family of machine learning methods that use data as it becomes available in a sequential order as opposed to batch learning techniques which rely on learning on the entire training data set at once [29]. As large scale sensor networks will invariably generate a large amount of data, centralized processing of this data can become computationally expensive. In the case of anomaly detection, a key advantage of online learning is that an ensemble of classifiers trained and executed "closer" to the sensor on a distributed embedded system can reduce energy consumption by limiting communication to a centralized server [30]. The development of online machine learning methods is especially relevant for networks that are transient in nature.

2.4.2 Distributed networks and federated learning

A distributed sensor network or DSN is defined as network consisting of a set of sensor nodes, a set of processing elements (PEs), and a communication network interconnecting the various PEs [31]. Large-scale sensor networks deployed in most applications would correspond to this definition. Each PE is associated with one or more sensors and a given sensor can report to more than one PE. A PE and its associated sensors are referred to as a *cluster* and each cluster can function autonomously and can serve as a processing node for incoming sensor data. Pre-processing sensor data in this way can reduce bandwidth costs and potential computational overheads arising from the centralized processing of raw data.

Federated learning (FL) refers to a machine learning paradigm in which a shared global model is trained from a *federation* of participating devices acting as local learners under the coordination of a central server for model-aggregation [32]. Federated learning was proposed [33] as a means to allow a model to be trained across multiple decentralized devices that hold local data samples without exchanging them - a feature that is particularly relevant in scenarios where data privacy, security, and access rights are a concern. For instance, a mobile device can download a given FL model from the cloud and improve it by learning from its own data. The resulting model is then sent back to the cloud to be averaged with updated models from other devices to improve the shared model. Thus, FL enables collective learning without sharing raw data by decentralizing the training of common predictive models. FL has been widely used for many use cases such as cross-institutional medical diagnosis [34], forecasting hospitalizations in the USA [35], [36], fraud detection [37], energy-demand forecasting [38], and pharmaceuticals discovery [39]. Sensor networks are a key application area of FL. For example, [40] developed a dynamic average consensus-based FL in a decentralized sensor network to overcome the challenges of imbalanced communication congestions and a possible single point of failure caused by relying on a centralized topology. FL was used by [41] to build a malicious node detection model for the Internet of Sensor Things (IoST) while [42] used FL to propose an aerial sensing framework for fine-grained 3-D ground air-quality monitoring and forecasting. Another example [43] used FL to solve the anomaly detection problem using sensor data in smart buildings with the aim of improving the prediction of energy usage in smart buildings. Recent efforts to quantify uncertainty quantification in FL have concentrated on methods to determine the model uncertainty [44], [45], [46]. However, approaches to

[1] https://github.com/Met4FoF/agentMET4FOF

uncertainty propagation in FL are still lacking and require further research.

2.5 Sensor networks and multi-agent systems

Sensor networks collect data over long periods of time in challenging environments under limited computational capabilities [47]. Broadly speaking, the development of sensor networks benefits from both hardware and software advancements. From a hardware perspective, the electronics are advancing towards the use of long-lasting batteries, energy-efficient sensors, reliable sensors, and wireless communication networks. On the other hand, much research on the software development has been to deploy energy-efficient algorithms and autonomous software agents [48], [49], [50]. Software agents are independent software processes imbued with autonomous duties (able to operate without human intervention) and common interfaces to communicate with other agents [48]. In computer science terms, an agent framework as a software package would provide abstract or base agent classes for developers to inherit from. Such software agents could be installed on edge computing devices and distributed servers, while enabling a seamless communication. Here, edge computing refers to a central concept of the IIoT, providing limited but capable computational power close to the actual sensors – aiming to enable energy efficient and decentral decision making, contributing to the systems sustainability.

2.5.1 Agent Architectures

Common agent classes designed for sensor networks include the Sensor Agent, Cluster Agent, Data Processing Agent, and Monitor Agent [48], [50], [51]. Their roles could be described as follows: the Sensor Agent represents the data stream of a sensor and has information and control over the sensor sampling rate; the Cluster Agent groups similar or related agents within its vicinity and aggregates sensors data; the Data Processing Agent performs mathematical operations on the sensor data to extract insights from the data. In more advanced applications, machine learning models could be used to detect anomalies and quantify uncertainties [52]; the Monitor Agent simply acts as a sink for the Data Processing Agent by gathering all the processed data and presenting them to the end-users. Another common role is the storage of historical data in a database.

In addition to defining the roles of agents, designing the topology of an agent network is also vital. Common sensor network topologies are star, tree and mesh topologies [53], [54]. In a star topology, the Sensor Agents are connected to a centralized server which hosts the Data Processing Agent and the Monitor Agent. This setup is simple and intuitive to design; however, as a centralized setup, the network is prone to a single point of failure.

The tree topology arranges the agents in a hierarchical fashion, and Cluster Agents are often used to orchestrate the Sensor Agents within their vicinity before sending the data to the Data Processing and Monitor Agents. The processing power of devices increases as we move higher in the tree hierarchy. This improves the self-healing capability and makes it straightforward to diagnose and rectify faults; however, it becomes increasingly complex to manage as the number of clusters and agents increases.

Lastly, in a flat, non-hierarchical mesh topology, all agents are fully connected to other agents. In a fully decentralized fashion, agents must cooperate efficiently to route the data transmission and processing. The major advantage is that a single point of failure will not form a bottleneck in the overall network, while enabling the use of distributed message-passing algorithms to improve energy and communication efficiency. These, however, comes with the cost of greater complexity in managing and implementing compared to a conventional centralized approach.

2.5.2 Real-world applications

The use of MAS for sensor networks has been demonstrated for numerous real-world applications such as structural health monitoring of large aircraft structures [51], fire-fighting disaster intervention [47], intrusion detection in wireless networks [55], crops and cattle monitoring [56], [59], energy efficient logistics [60], smart road systems [57], [58], [59], multi-video monitoring [60], autonomous underwater vehicles (AUVs) and unmanned aerial vehicles (UAVs) [61], [66], oil and gas refinery monitoring [62], and smart home monitoring [68], among others.

3. REVIEW: TOOLS

In this section we discuss some of the available tools that are relevant to automation in sensor networks. In particular, topics directly relevant to metrology such as the automated evaluation and propagation of measurement uncertainty, as well as more general topics such as the FIWARE framework and agentMET4FOF package will be discussed.

3.1 Automated Uncertainty Evaluation

Automated uncertainty evaluation (AUE) comprises algorithms that can process measurement data, identify sources of uncertainty (both systematic and random errors), and calculate the combined uncertainty as well as propagate known uncertainties to a derived quantity. Importantly, online capability and automation in such algorithms must be integrated directly into a measurement system – in the present case this refers to a given sensor network. An automated machine learning toolbox (AMLT), published by [63], allows for feature extraction from, and evaluation of cyclic sensor data. The aforementioned tool was extended in [64] to allow for uncertainty propagation according to the "Guide to the expression of Uncertainty in Measurement" (GUM, [65]). Several software packages for propagation of uncertainty based on the GUM are currently available. For example, the GUM Tree Calculator (GTC) is a data processing tool available as a python-based stand-alone Windows executable with full support for uncertainty calculations involving real and complex quantities [66]. The GUM Workbench supports the evaluation of measurements with multiple results and multiple budget tables as well as uncertainty propagation using Monte Carlo simulations [67]. The NIST Uncertainty machine [68] – a web-based uncertainty calculator, and the "metRology" package [69] support uncertainty evaluation and are based on the R programming language. Metas.UncLib [70] was created as part of the metrology software for the vector network analyser (VNA). As a Python-based general-purpose library for measurement uncertainty evaluation, it is capable of handling complex-valued quantities that require a multivariate treatment for the calculation of the measurement uncertainty. PyDynamic [71] is another Python based library for users in metrology and related areas who want to specifically deal with time-dependent, i.e. *dynamic*, measurements. PyDynamic allows for an off-the-shelf application of NMI-level data analysis and measurement uncertainty evaluation methods.

3.3 agentMET4FOF

agentMET4FOF [50], [72] is a Python-based software framework to set up asynchronous agents with common features for distributed data streaming processes. It is an implementation of a multi-agent system for agent-based analysis and processing of both static data sets and data streams with IIoT applications in mind. Main applications have been demonstrated on sensor network for industrial use cases such as quality monitoring and equipment condition monitoring. Some key features of agentMET4FOF are as follows:

- **Modularity**: Several preconfigured classes are provided for simulating or handling real distributed sensor networks.
- **Reconfigurability**: The agent connections can be reconfigured on the fly. It can be used to model different sensor network topologies. When necessary, the properties or parameters of agents could also be modified.
- **Extensibility**: custom classes can easily be integrated in the provided interface structure
- **Buffering:** is a commonly required feature for processing online streaming data. For instance, online machine learning applications need to update the model with incoming data. The agent therefore needs to push and pop the data buffer with every new observation. The AgentBuffer class is provided as a common interface to check the buffer filled status and update it. Optionally, the buffer can be turned off for agents that do not need it.
- **Graphical user interface**: agentMET4FOF is provided with an interactive web-based dashboard that offers real-time visualization and configuration capabilities. This helps in quickly setting up demonstrations to stakeholders.
- **Different backends**: The osbrain backend [73] is primarily used to deploy real distributed software processes; for instance, the deployment can take place on different edge devices like Raspberry Pi's or other computers connected via a TCP network. For pure testing and simulation purposes, the backend can be swapped to the Mesa backend [74]. This could be to verify the agent network works as intended before actual deployment using the osbrain backend.
- **Metrological data classes**: A key feature of agentMET4FOF is the provision of specific metrological data classes that incorporate measurement uncertainties. This includes the support of both data streams with uncertainty and metrological agents that process data with uncertainty.

The agentMET4FOF software is accompanied by comprehensive documentation and tutorials [75].

3.4 FIWARE

FIWARE offers an open-source framework that provides a standardized approach to managing context information in a way that is interoperable across various domains [76]. In order to ensure interoperability, a variety of protocols need to be implemented. Some typical examples of communication protocols are MQTT, CoAP, and AMQP, which are widely used within IoT ecosystems for their efficiency and low bandwidth requirements [77]. These protocols facilitate real-time data communication, which is essential for online machine learning where immediate data processing is critical [78]. FIWARE's role as a middleware can abstract the complexity of these protocols, providing a unified interface through its Orion Context Broker, which adheres to the Next Generation Service Interface (NGSI) standard [79]. This standardization is vital for ensuring that sensor networks are scalable and that machine learning algorithms can be distributed and federated across different systems [80]. Federated learning (see section 2.4.2) is especially relevant for sensor networks where privacy and bandwidth are concerns. FIWARE can support federated learning by enabling distributed devices to interact with a central context broker, managing the global model updates without exposing local data [81]. This is particularly important in scenarios like Smart Cities, where data from various sensors could be leveraged to improve urban services while maintaining data privacy. Generally, FIWARE presents a suite of components, with the Orion Context Broker [82] as a centrepiece, enabling the management and gathering of context information at a large scale. It offers a standardized Application Programming Interface (API) - e.g., NGSI - for context information management that facilitates the development and deployment of smart applications in a variety of relevant applications, such as environmental monitoring, across numerous use-cases. A few key examples in this regard are a FIWARE based smart-city model demonstrating the advantage of the Orion Context Broker [83]; an architecture for wireless sensing and actuation in smart buildings, including real-time information exchange using a BIM (Building Information Model) [84]; the intelligent management of sports infrastructure [85]; an IoT-based acquisition of agricultural data [86]. There are also examples of FIWARE infrastructure used in manufacturing processes [87] and for increasing energy efficiency in buildings beyond the traditional energy-saving measures [88].

4. CONCLUSIONS AND OUTLOOK

In this review, a discussion of methods and tools relevant to the automated application of metrological methods to large scale sensor networks was presented. The first part of the review discussed methods relevant to applications using sensor networks such as *in situ* calibration and the calibration of hybrid models. The use of machine learning methods, particularly online machine learning and federated learning using distributed systems were subsequently covered. Thereafter, the discussion focussed on the use of agent-based systems in sensor networks. The discussion on methods was complemented by a review of specific software tools with potential applications to sensor network metrology. In particular, tools for automated uncertainty evaluation, the agentMET4FOF package for agent-based simulations and the FIWARE middleware framework were discussed. The aim of the review is to develop a set of requirements and scenarios that will be included in future software environments for the automated application of metrological methods developed for sensor network metrology.

In order to be applicable to large-scale transient networks, a software environment would need to fulfil certain requirements. An online-capable uncertainty propagation feature is necessary due to the transient nature of the networks used, particularly when mobile sensors are involved (e. g. in air-quality networks). Ideally, the methods would be implementable in both the time- and frequency/Laplace-domains with the propagation of uncertainties for dynamic measurements integrated into the framework. Simultaneously, the capacity for the online implementation of drift detection methods is necessary. The ability to implement uncertainty-aware sensor fusion methods for both data-driven and physics-based models is also necessary. Data-aggregation protocols will need to be appropriately adapted to use cases with both fixed and mobile sensors. Finally, methods, both semantic and numerical, to assess the correctness and trustworthiness of data will need to be integrated into the framework. Ideally, a testbed for models developed in this manner will be able to conduct simulations at both the software and middleware levels.

ACKNOWLEDGMENTS

The project (22DIT02 FUNSNM) has received funding from the European partnership on metrology, co-financed from the European union's Horizon Europe Research and Innovation Programme and by the participating states.

REFERENCES

[1] F. Delaine, B. Lebental, and H. Rivano, "In Situ Calibration Algorithms for Environmental Sensor Networks: A Review," *IEEE Sens. J.*, vol. 19, no. 15, pp. 5968–5978, Aug. 2019, doi: 10.1109/JSEN.2019.2910317.

[2] M. Gruber, A. P. Vedurmudi, and S. Eichstädt, "Co-Calibration in Distributed Homogeneous Sensor Networks," Sensor and Measurement Science International, SMSI 2023, 2023, pp. 47–48. doi: 10.5162/SMSI2023/A3.2.

[3] K. Whitehouse and D. Culler, "Macro-Calibration in Sensor/Actuator Networks," *Mob. Netw. Appl.*, vol. 8, pp. 463–472, Aug. 2003, doi: 10.1023/A:1024548100497.

[4] M. S. Stanković, S. S. Stanković, K. H. Johansson, M. Beko, and L. M. Camarinha-Matos, "On Consensus-Based Distributed Blind Calibration of Sensor Networks," *Sensors*, vol. 18, no. 11, p. 4027, Nov. 2018, doi: 10.3390/s18114027.

[5] L. Balzano and R. Nowak, "Blind Calibration of Sensor Networks," IEEE, 2007, pp. 79–88. doi: 10.1109/IPSN.2007.4379667.

[6] Y. Cheng, X. He, Z. Zhou, and L. Thiele, "ICT: In-field Calibration Transfer for Air Quality Sensor," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 3, no. 1, pp. 6–19, Mar. 2019, doi: 10.1145/3314393.

[7] M. A. Zaidan *et al.*, "Intelligent Calibration and Virtual Sensing for Integrated Low-Cost Air Quality Sensors," *IEEE Sens. J.*, vol. 20, no. 22, pp. 13638–13652, Nov. 2020, doi: 10.1109/JSEN.2020.3010316.

[8] C. Malings, R. Tanzer, A. Hauryliuk, S. P. N. Kumar, N. Zimmerman, L. B. Kara, A. A. Presto, and R. Subramanian, "Development of a general calibration model and long-term performance evaluation of low-cost sensors for air pollutant gas monitoring," *Atmospheric Meas. Tech.*, vol. 12, no. 2, pp. 903–920, Feb. 2019, doi: 10.5194/amt-12-903-2019.

[9] H. S. Russell, L. B. Frederickson, S. Kwiatkowski, A. P. M. Emygdio, P. Kumar, J. A. Schmidt, O. Hertel, and M. S. Johnson, "Enhanced Ambient Sensing Environment—A New Method for Calibrating Low-Cost Gas Sensors," *Sensors*, vol. 22, no. 19, p. 7238, Sep. 2022, doi: 10.3390/s22197238.

[10] E. Bagkis, T. Kassandros, and K. Karatzas, "Learning Calibration Functions on the Fly: Hybrid Batch Online Stacking Ensembles for the Calibration of Low-Cost Air Quality Sensor Networks in the Presence of Concept Drift," *Atmosphere*, vol. 13, no. 3, p. 416, Mar. 2022, doi: 10.3390/atmos13030416.

[11] G. Yin, G. Yang, W. Yang, B. Zhang, and W. Jin, "An energy-efficient routing algorithm for wireless sensor networks," in *2008 International Conference on Internet Computing in Science and Engineering*, IEEE, 2008, pp. 181–186.

[12] K. Dasgupta, K. Kalpakis, and P. Namjoshi, "An efficient clustering-based heuristic for data gathering and aggregation," IEEE, 2003, pp. 1948–1953. doi: 10.1109/WCNC.2003.1200685.

[13] S. Sasirekha and S. Swamynathan, "Cluster-chain mobile agent routing algorithm for efficient data aggregation in wireless sensor network," *J. Commun. Netw.*, vol. 19, no. 4, pp. 392–301, Aug. 2017, doi: 10.1109/JCN.2017.000063.

[14] A. Farinelli, A. Rogers, and N. R. Jennings, "Agent-based decentralised coordination for sensor networks using the max-sum algorithm," *Auton. Agents Multi-Agent Syst.*, vol. 28, pp. 337–380, 2014, doi: 10.1007/s10458-013-9225-1.

[15] S. G. Tzafestas, "12 - Mobile Robot Localization and Mapping," in *Introduction to Mobile Robot Control*, S. G. Tzafestas, Ed., Oxford: Elsevier, 2014, pp. 479–531. doi: https://doi.org/10.1016/B978-0-12-417049-0.00012-2.

[16] G. Kok and P. Harris, "Uncertainty Evaluation for Metrologically Redundant Industrial Sensor Networks," presented at the 2020 IEEE International Workshop on Metrology for Industry 4.0 & IoT, Rome, Italy: IEEE, Jun. 2020, pp. 84--88. doi: 10.1109/MetroInd4.0IoT48571.2020.9138297.

[17] M. G. Cox, "The evaluation of key comparison data," *Metrologia*, vol. 39, no. 6, p. 589, 2002, doi: 10.1088/0026-1394/39/6/10.

[18] G. Kok and P. Harris, "Quantifying Metrological Redundancy in an Industry 4.0 Environment," presented at the 2020 IEEE International Workshop on Metrology for Industry 4.0 & IoT, Rome, Italy: IEEE, Jun. 2020, pp. 464--468. doi: 10.1109/MetroInd4.0IoT48571.2020.9138297.

[19] M. A. Zaidan *et al.*, "Dense Air Quality Sensor Networks: Validation, Analysis, and Benefits," *IEEE Sens. J.*, vol. 22, no. 23, pp. 23507–23520, Dec. 2022, doi: 10.1109/JSEN.2022.3216071.

[20] M. A. Zaidan *et al.*, "Intelligent Air Pollution Sensors Calibration for Extreme Events and Drifts Monitoring," *IEEE Trans. Ind. Inform.*, vol. 19, no. 2, pp. 1366–1379, Feb. 2023, doi: 10.1109/TII.2022.3151782.

[21] M. Gruber *et al.*, "Application of Uncertainty-Aware Sensor Fusion in Physical Sensor Networks," in *2022 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, 2022, pp. 1–6. doi: 10.1109/I2MTC48687.2022.9806580.

[22] H. Boyes, B. Hallaq, J. Cunningham, and T. Watson, "The industrial internet of things (IIoT): An analysis framework," *Comput. Ind.*, vol. 101, pp. 1–12, Oct. 2018, doi: 10.1016/j.compind.2018.04.015.

[23] R. A. Khalil, N. Saeed, M. Masood, Y. M. Fard, M.-S. Alouini, and T. Y. Al-Naffouri, "Deep Learning in the Industrial Internet of Things: Potentials, Challenges, and Emerging Applications," *IEEE Internet Things J.*, vol. 8, no. 14, pp. 11016–11040, Jul. 2021, doi: 10.1109/JIOT.2021.3051414.

[24] S. Latif, M. Driss, W. Boulila, Z. E. Huma, S. S. Jamal, Z. Idrees, and J. Ahmad, "Deep Learning for the Industrial Internet of Things (IIoT): A Comprehensive Survey of Techniques, Implementation Frameworks, Potential Applications, and Future Directions.," *Sensors*, vol. 21, no. 22, Nov. 2021, doi: 10.3390/s21227518.

[25] B. Maschler, S. Ganssloser, A. Hablizel, and M. Weyrich, "Deep learning based soft sensors for industrial machinery," *Procedia CIRP*, vol. 99, pp. 662–667, May 2021, doi: 10.1016/j.procir.2021.03.115.

[26] C. Shang, F. Yang, D. Huang, and W. Lyu, "Data-driven soft sensor development based on deep learning technique," *J. Process Control*, vol. 24, no. 3, pp. 223–233, Mar. 2014, doi: 10.1016/j.jprocont.2014.01.012.

[27] A. H. Abdelaziz, S. Watanabe, J. R. Hershey, E. Vincent, and D. Kolossa, “Uncertainty propagation through deep neural networks,” Proceedings of Interspeech 2015, 2015, pp. 3561–3565. doi: 10.21437/Interspeech.2015-706.

[28] R. F. Astudillo and J. P. da Silva Neto, “Propagation of uncertainty through multilayer perceptrons for robust automatic speech recognition,” Proceedings of Interspeech 2011, 2011, pp. 461–464. doi: 10.21437/Interspeech.2011-196.

[29] S. C. H. Hoi, D. Sahoo, J. Lu, and P. Zhao, “Online learning: A comprehensive survey,” *Neurocomputing*, vol. 459, pp. 249–289, Oct. 2021.

[30] H. H. W. J. Bosman, G. Iacca, A. Liotta, and H. Wörtche, “Online Fusion of Incremental Learning for Wireless Sensor Networks,” IEEE, 2014, pp. 525–532.

[31] H. Qia, S. S. Iyengar, and K. Chakrabarty, “Distributed sensor networks – a review of recent research,” *J. Frankl. Inst.*, vol. 338, no. 6, pp. 665–668, May 2001, doi: 10.1016/S0016-0032(01)00026-6.

[32] S. Savazzi, M. Nicoli, and V. Rampa, “Federated Learning With Cooperating Devices: A Consensus Approach for Massive IoT Networks,” *IEEE Internet Things J.*, vol. 7, no. 5, pp. 4641–4654, Jan. 2020, doi: 10.1109/JIOT.2020.2964162.

[33] B. McMahan and D. Ramage, “Federated Learning: Collaborative Machine Learning without Centralized Training Data.” [Online]. Available: https://blog.research.google/2017/04/federated-learning-collaborative.html

[34] M. J. Sheller *et al.*, “Federated learning in medicine: facilitating multi-institutional collaborations without sharing patient data,” *Sci. Rep.*, vol. 10, no. 1, pp. 1–12, Jul. 2020, doi: 10.1038/s41598-020-69250-1.

[35] T. S. Brisimi, R. Chen, T. Mela, A. Olshevsky, I. Ch. Paschalidis, and W. Shi, “Federated learning of predictive models from federated Electronic Health Records,” *Int. J. Med. Inf.*, vol. 112, pp. 59–67, Apr. 2018, doi: 10.1016/j.ijmedinf.2018.01.007.

[36] I. Dayan *et al.*, “Federated learning for predicting clinical outcomes in patients with COVID-19,” *Nat. Med.*, vol. 27, pp. 1735–1743, Sep. 2021, doi: 10.1038/s41591-021-01506-3.

[37] W. Yang, Y. Zhang, K. Ye, L. Li, and C.-Z. Xu, “FFD: A Federated Learning Based Method for Credit Card Fraud Detection,” Springer, Cham, 2019. doi: 10.1007/978-3-030-23551-2_2.

[38] Y. L. Tun, K. Thar, and C. S. Hong, “Federated Learning based Energy Demand Prediction with Clustered Aggregation,” IEEE, 2021, pp. 164–167. doi: 10.1109/BigComp51126.2021.00039.

[39] O. Choudhury, Y. Park, T. Salonidis, A. Gkoulalas-Divanis, I. Syalla, and A. Das, “Predicting Adverse Drug Reactions on Distributed Health Data using Federated Learning,” American Medical Informatics Association, 2020, pp. 313–322.

[40] Z. Chen, D. Li, J. Zhu, and S. Zhang, “DACFL: Dynamic Average Consensus-Based Federated Learning in Decentralized Sensors Network,” *Sensors*, vol. 22, no. 9, p. 3317, Apr. 2022, doi: 10.3390/s22093317.

[41] Z. Abubaker, N. Javaid, A. Almogren, M. Akbar, M. Zuair, and J. Ben-Othman, “Blockchained service provisioning and malicious node detection via federated learning in scalable Internet of Sensor Things networks,” *Comput. Netw.*, vol. 204, p. 108691, Feb. 2022, doi: 10.1016/j.comnet.2021.108691.

[42] Y. Liu, J. Nie, X. Li, S. H. Ahmed, W. Y. B. Lim, and C. Miao, “Federated Learning in the Sky: Aerial-Ground Air Quality Sensing Framework With UAV Swarms,” *IEEE Internet Things J.*, vol. 8, no. 12, pp. 9827–9837, Jun. 2021, doi: 10.1109/JIOT.2020.3021006.

[43] R. A. Sater and A. B. Hamza, “A Federated Learning Approach to Anomaly Detection in Smart Buildings,” *ACM Trans. Internet Things*, vol. 2, no. 4, pp. 1–23, Nov. 2021, doi: 10.1145/3467981.

[44] Y. Gal and Z. Ghahramani, “Dropout as a Bayesian approximation: representing model uncertainty in deep learning,” in *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, in ICML’16. JMLR.org, 2016, pp. 1050–1059.

[45] L. Kong, J. Sun, and C. Zhang, “SDE-Net: equipping deep neural networks with uncertainty estimates,” in *Proceedings of the 37th International Conference on Machine Learning*, in ICML’20. JMLR.org, 2020.

[46] F. Linsner, L. Adilova, S. Däubener, M. Kamp, and A. Fischer, “Approaches to Uncertainty Quantification in Federated Deep Learning,” in *Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, vol. 1524, M. Kamp, I. Koprinska, A. Bibal, T. Bouadi, B. Frénay, L. Galárraga, J. Oramas, L. Adilova, Y. Krishnamurthy, B. Kang, G. Graça, et al., Eds., in Communications in Computer and Information Science, vol. 1524. , Cham: Springer International Publishing, 2021, pp. 128–145. doi: 10.1007/978-3-030-93736-2_12.

[47] R. Fuentes-Fernández, M. Guijarro, and G. Pajares, “A Multi-Agent System Architecture for Sensor Networks,” *Sensors*, vol. 9, no. 12, pp. 10244–10269, Dec. 2009, doi: 10.3390/s91210244.

[48] O. Dagdeviren, I. Korkmaz, F. Tekbacak, and K. Erciyes, “A survey of agent technologies for wireless sensor networks,” *IETE Tech. Rev.*, vol. 28, no. 2, pp. 168–184, Mar. 2011, doi: 10.4103/0256-4602.72509.

[49] A. Rogers, D. D. Corkill, and N. R. Jennings, “Agent technologies for sensor networks,” *IEEE Intell. Syst.*, vol. 24, no. 2, pp. 13–17, Mar. 2009, doi: 10.1109/MIS.2009.22.

[50] B. X. Yong and A. Brintrup, “Multi Agent System for Machine Learning Under Uncertainty in Cyber Physical Manufacturing System,” Springer, Cham, 2019, pp. 244–257. doi: 10.1007/978-3-030-27477-1_19.

[51] J. Wu, S. Yuan, S. Ji, G. Zhou, Y. Wang, and Z. Wang, “Multi-agent system design and,” *Expert Syst. Appl.*, vol. 37, no. 3, pp. 2028–2036, Mar. 2010, doi: 10.1016/j.eswa.2009.06.098.

[52] B. X. Yong and A. Brintrup, “Coalitional Bayesian Autoencoders - Towards explainable unsupervised deep learning with applications to condition monitoring under covariate shift,” *Appl. Soft Comput.*, vol. 123, p. 108912, Jul. 2022, doi: 10.1016/j.asoc.2022.108912.

[53] A. Mouapi and N. Hakem, “A new approach to design autonomous wireless sensor node based on rf energy,” *Sensors*, vol. 18, no. 3, p. 133, Jan. 2018, doi: 10.3390/s18010133.

[54] M. J. McGrath and C. N. Scanaill, *Sensor Network Topologies and Design Considerations*. Berkeley, CA: Apress, 2013. doi: 10.1007/978-1-4302-6014-1_4.

[55] H. Wang, Z. Yuan, and C. Wang, “Intrusion detection for wireless sensor networks based on multi-agent and refined clustering,” in *2009 WRI International Conference on Communications and Mobile Computing*, IEEE, 2009, pp. 450–454.

[56] A. L. Barriuso, G. Villarrubia González, J. F. De Paz, Á. Lozano, and J. Bajo, “Combination of multi-agent systems and wireless sensor networks for the monitoring of cattle,” *Sensors*, vol. 18, no. 1, p. 108, 2018.

[57] M. Tubaishat, Y. Shang, and H. Shi, “Adaptive traffic light control with wireless sensor networks,” in *2007 4th IEEE Consumer Communications and Networking Conference*, IEEE, 2007, pp. 187–191.

[58] J. Lu, L. Feng, J. Yang, M. M. Hassan, A. Alelaiwi, and I. Humar, “Artificial agent: The fusion of artificial intelligence and a mobile agent for energy-efficient traffic control in wireless sensor networks,” *Future Gener. Comput. Syst.*, vol. 95, pp. 45–51, 2019.

[59] M. V. Muntean, “Multi-agent system for intelligent urban traffic management using wireless sensor networks data,” *Sensors*, vol. 22, no. 1, p. 208, 2021.

[60] M. A. Patricio, J. Carbó, O. Pérez, J. García, and J. Molina, “Multi-agent framework in visual sensor networks,” *EURASIP J. Adv. Signal Process.*, vol. 2007, pp. 1–21, 2006.

[61] M. R. Bharamagoudra, S. S. Manvi, and B. Gonen, “Event driven energy depth and channel aware routing for underwater acoustic sensor networks: Agent oriented clustering based approach,” *Comput. Electr. Eng.*, vol. 58, pp. 1–19, 2017.

[62] M. S. Taboun and R. W. Brennan, “An embedded multi-agent systems based industrial wireless sensor network,” *Sensors*, vol. 17, no. 9, p. 2112, 2017.

[63] M. Bastuck, T. Baur, and A. Schütze, “DAV3E – a MATLAB toolbox for multivariate sensor data evaluation,” *J. Sens. Sens. Syst.*, vol. 7, no. 2, pp. 489–506, Sep. 2018.

[64] T. Dorst, T. Schneider, S. Eichstädt, and A. Schütze, “Uncertainty-aware automated machine learning toolbox,” *Tm -*

Tech. Mess., vol. 90, no. 3, pp. 141–153, Sep. 2022, doi: 10.1515/teme-2022-0042.
[65] BIPM, IEC, IFCC, ILAC, ISO, IUPAC, IUPAP, and OIML, “Guide to the expression of uncertainty in measurement.” 2008.
[66] B. Hall, “The GUM Tree Calculator: A Python Package for Measurement Modelling and Data Processing with Automatic Evaluation of Uncertainty,” *Metrology*, vol. 2, no. 1, pp. 128–149, 2022, doi: 10.3390/metrology2010009.
[67] Metrodata GmbH, “GUM Workbench Professional Version 2.4.” [Online]. Available: http://www.metrodata.de/ver24_en.html
[68] T. Lafarge and A. Possolo, “NIST Uncertainty Machine — User’s Manual.” Nov. 2020. [Online]. Available: https://uncertainty.nist.gov/NISTUncertaintyMachine-UserManual.pdf
[69] S. L. R. Ellison, “metRology: Support for Metrological Applications.” [Online]. Available: https://CRAN.R-project.org/package=metRology
[70] M. Zeier, J. Hoffmann, and M. Wollensack, “Metas.UncLib – a measurement uncertainty calculator for advanced problems,” *Metrologia*, vol. 49, pp. 809–815, Nov. 2012, doi: 10.1088/0026-1394/49/6/809.
[71] S. Eichstädt, C. Elster, I. M. Smith, and T. J. Esward, “Evaluation of dynamic measurement uncertainty – an open-source software package to bridge theory and practice,” *J. Sens. Sens. Syst.*, vol. 6, no. 1, pp. 97–105, Feb. 2017, doi: 10.5194/jsss-6-97-2017.
[72] B. Ludwig, B. X. Yong, A. P. Vedurmudi, H. Lulic, M. Gruber, and G. Kok, “Met4fof/agentmet4fof: v0.13.2.” [Online]. Available: https://doi.org/10.5281/zenodo.5965562.
[73] OpenSistemas, “osbrain—0.6.5 documentation.” 2016. [Online]. Available: https://osbrain.readthedocs.io/en/v0.6.5/.
[74] J. Kazil, D. Masad, and A. Crooks, “Utilizing python for agent-based modeling: The mesa framework,” in *Social, Cultural, and Behavioral Modeling*, R. Thomson, H. Bisgin, C. Dancy, A. Hyder, and M. Hussain, Eds., Springer International Publishing, 2020, pp. 308–317. doi: 10.1007/978-3-030-61255-9_30.
[75] B. X. Yong, B. Ludwig, A. P. Vedurmudi, M. Gruber, and H. Lulic, “agentMET4FOF - Metrological agent-based system.” [Online]. Available: https://agentmet4fof.readthedocs.io/en/latest/
[76] SmartCitiesWorld and FIWARE Foundation, “City data spaces: a guide to building-and operationalising data services.” Jun. 2023. [Online]. Available: https://www.smartcitiesworld.net/whitepapers/city-data-spaces-a-guide-to-building-and-operationalising-data-services
[77] FIWARE Foundation, “FIWARE: The Open Source Platform of Choice for Building Smart Energy Solutions.” Aug. 2018. [Online]. Available: https://www.fiware.org/about-us/smart-energy/
[78] “Paving the Way for a Data-Driven Industry Digitalization.” Apr. 2018. [Online]. Available: https://www.fiware.org/wp-content/directories/marketing-toolbox/material/FIWAREBrochure_SmartIndustry.pdf
[79] IGS, “ETSI GS CIM 009 V1.8.1.” Accessed: Apr. 08, 2024. [Online]. Available: https://www.etsi.org/deliver/etsi_gs/CIM/001_099/009/01.08.01_60/gs_CIM009v010801p.pdf
[80] FIWARE Foundation, “FIWARE for Smart Cities and Territories: A Digital Transformation Journey.” Oct. 2023. [Online]. Available: https://www.fiware.org/wp-content/directories/marketing-toolbox/material/FIWAREBrochure_SmartCities.pdf
[81] FIWARE Foundation, “FIWARE: The Open Source Platform of Choice for Building Smart Farming Solutions.” Jun. 2018. [Online]. Available: https://www.fiware.org/wp-content/directories/marketing-toolbox/material/FIWAREBrochure_SmartAgriFood.pdf
[82] F. Cirillo, G. Solmaz, E. L. Berz, M. Bauer, B. Cheng, and E. Kovacs, “A Standard-Based Open Source IoT Platform: FIWARE,” *IEEE Internet Things Mag.*, vol. 2, no. 3, Sep. 2019, doi: 10.1109/IOTM.0001.1800022.
[83] A. Sinanovic, E. Meskovic, A. Mujcic, and N. Suljanovic, “Smart city use case development based on FIWARE technology,” IEEE, 2022, pp. 1–4. doi: 10.1109/TELFOR56187.2022.9983683.
[84] A. Z. Román-García, R. Nieto, P. V. Hernandez, M. C. Rodriguez-Sanchez, and M. G. Carrillo, “Smart Building Digital Twin: Wireless Sensing and Actuation Architecture at Rey Juan Carlos University,” IEEE, 2023, pp. 65–69. doi: 10.1109/MetroLivEnv56897.2023.10164028.
[85] C. Camargo, G. Silva, H. V. Rosse, J. Barbosa, and D. Marques, “Internet of Things for Intelligent Management of Professional Football Turf Pitches,” IEEE, 2023, pp. 427–432. doi: 10.1109/MetroInd4.0IoT57462.2023.10180142.
[86] A. Heideker, D. Ottolini, I. Zyrianoff, A. T. Neto, T. S. Cinotti, and C. Kamienski, “IoT-based Measurement for Smart Agriculture,” IEEE, 2020, pp. 68–72. doi: 10.1109/MetroAgriFor50201.2020.9277546.
[87] N. Guedes, I. Capitanio, H. V. Rosse, J. P. Coelho, J. Barbosa, N. Pires, and J. Magalhães, “Improving the Traceability of Wood-based Sheet Leftovers using Computer Vision,",” IEEE, 2023, pp. 1–6. doi: 10.1109/ICPS58381.2023.10127865.
[88] T. Schranz, Q. Alfalouji, T. Hirsch, and G. Schweiger, “An Open IoT Platform: Lessons Learned from a District Energy System,” IEEE, 2022, pp. 1–9. doi: 10.1109/SMART55236.2022.9990228.

AUTHOR INFORMATION

[e] VTT MIKES, Tekniikantie 1, 02150 Espoo, Finland
[f] University of Helsinki, Gustaf Hällströmin katu 2a, FI-00560 Helsinki, Finland
[g] Forschungszentrum Jülich, Jülich, Germany