8

Lesson Learnt and Future of AI Applied to Manufacturing

Valerio Frascolla[1], Matthias Hummert[2], Tobias Monsees[2], Dirk Wübben[2], Armin Dekorsy[2], Nicola Michailow[3], Volkmar Döricht[3], Christoph Niedermeier[3], Joachim Kaiser[3], Arne Bröring[3], Michael Villnow[3], Daniel Wessel[4], Florian Geiser[4], Matthias Wissel[4], Alberto Viseras[4], Bin Han[5], Björn Richerzhagen[6], Hans Schotten[5], Davide Calandra[7], and Fabrizio Lamberti[7]

[1]Intel Deutschland GmbH, Germany
[2]University of Bremen, Germany
[3]Siemens AG, Germany
[4]Motius GmbH, Germany
[5]TU Kaiserslautern, Germany
[6]Siemens, Germany
[7]Politecnico di Torino, Italy

Abstract

This chapter touches on several aspects related to the role of Artificial Intelligence (AI) and Machine Learning (ML) in the manufacturing sector, and is split in different sub-chapters, focusing on specific new technology enablers that have the potential of solving or minimizing known issues in the manufacturing and, more in general, in the Industrial Internet of Things (IIoT) domain.

After introducing AI/ML as a technology enabler for the IoT in general and for manufacturing in particular, the next four sections detail two key technology enablers (EdgeML and federated learning scenarios, challenges and tools), one most important area of the IoT system that needs to decrease energy consumption and increase reliability (reduce receiver

Processing complexity and enhancing reliability through multi-connectivity uplink connections), and finally a glimpse at the future describing a promising new technology (Embodied AI), its link with millimetre waves connectivity and potential business impact.

Keywords: Artificial intelligence, machine learning, internet of things, EdgeML, federated learning, mobile communication, 5G, embodied artificial intelligence, platform economy, millimetre waves, manufacturing, IIoT.

8.1 Introduction

This chapter touches on several aspects related to the role of Artificial Intelligence (AI) and Machine Learning (ML) in the manufacturing sector, and is split in different sub-chapters, focusing on specific new technology enablers that have the potential of solving or minimizing known issues in the manufacturing and, more in general, in the Industrial Internet of Things (IIoT) domain.

The two main challenges that IIoT currently faces are the security of the system and the capability to scale the number of devices, which continuously increase year by year. Among the most suited new technology enablers to cope with both challenges, AM/ML techniques are a highly discussed topic, especially the application of *EdgeML* and Federated Learning (FL) seem two very promising approaches. Other important issues of IIoT systems are the complexity at receivers' side and the reliability of the connections, the first impacting the terminals' energy consumption, the latter the minimum guaranteed quality of service of the overall system.

The structure of the chapter is as follows: Section 2 provides an introduction of ML applied to the IoT domain and Section 3 a description of both advantages and challenges of applying edge ML. Section 4 elaborates on FL techniques, their advantages, and the most popular open frameworks and commercial products implementing FL. Section 5 focuses on the main computational issues on the receiver side of IIoT systems, providing an overview of the research carried out in FunKI, a German funded research project, and discussing how to improve reliability in a multi-connectivity set-up for the uplink. Finally, Section 6 provides a more forward-looking view on Embodied AI, a promising approach in IIoT and manufacturing, and evaluates its potential business impact on future systems.

8.2 IoT Enabled by Machine Learning

The term Internet of Things (IoT) describes the intersection between the physical world and digital services. IoT devices are connected to the web and either stream collected data to cloud servers or receive control commands from external devices e.g., other IoT devices or mobile phones. IoT devices are a fundamental part of our daily life and are key for a wide range of industries, including agriculture, energy, security, smart homes, med-tech, and automotive. IoT devices typically include various types of sensors to measure relevant features of an object, e.g., acceleration, orientation, and position, or to sense environmental conditions. Sensors continuously sample the environment, which results in the generation of massive amounts of data. In 2018, there were already 22 B IoT devices in use, and forecasts show that by 2030, the number will reach 50 B devices worldwide [1]. To tame such complexity and extract meaningful values from the huge data generated by this rapidly growing field, ML has emerged as the most promising candidate technology.

The combination of ML algorithms and real-time data provided by IoT devices will positively impact most industrial applications. For example, data collected by IoT devices can be used for creating or enhancing Digital Twins (DTs), as well as for performing big data analytics. When combined with ML approaches, applications such as just-in-time manufacturing or demand forecasting emerge. Nevertheless, the transformation to Industry 4.0, where ML and edge computing are key technologies [2], must deal with several challenges that might slow down its adoption. Examples of those challenges are cyber threats or the issue of the integration of legacy equipment, protocols, and subsystems, which are present in most industrial facilities.

Despite the previously mentioned challenges, multiple approaches have been recently proposed to use ML in combination with IoT devices [3], [4]. ML for IoT has been traditionally accomplished by gathering the collected data from a group of IoT devices into a central location for training a global model, which can be used for prediction across devices. Thanks to the rise in on-demand access to high powered accelerators provided by cloud services, ML models are increasingly often being trained in the cloud. Once trained, it is often easiest to deploy the model on the cloud using similar infrastructure used for training. This approach for training and serving models for inference, known as ***centralized ML***, may result in a high network usage, as all gathered information must be streamed to the cloud. Furthermore, the results from running inference may need to be sent back to the edge. This communication loop is not ideal for some use cases, especially when low latency and data

privacy are in focus. Real-time systems, which require decisions being made in fractions of a second, cannot rely on the communication latency of sending data to and from a central location. Furthermore, by collecting data centrally, it is not guaranteed that sensitive data is treated in private and secure ways.

8.3 Machine Learning at the Edge

One alternative to ***centralized ML*** is to run the model inference on the same devices that collect the data. This approach, known as *EdgeML*, does not require any data to be sent centrally for performing model inference [5]. As a result, it addresses some drawbacks of centralized ML, e.g., high network bandwidth consumption and latency. *EdgeML* also allows for use cases where internet connection is not always reliable or even available. Furthermore, as the data never leaves the device, data privacy poses less problems. *EdgeML* is a trend that has recently found its peak and is expected to reach the plateau of productivity in about two to five years, according to the July 2021 Gartner Hype Cycle for Artificial Intelligence report [6].

In a standard *EdgeML* for IoT use cases, the edge devices may not be powerful enough to run a standard ML model for inference, for example in the cases of microcontrollers such as an ESP32 [7] or some low-powered, Linux-capable devices such as a Raspberry Pi [8]. These devices have limited memory, meaning they may not even be able to load and run a standard deep learning (DL) model. As such, model compression techniques need to be utilized to meet memory and runtime requirements. Tools such as TensorFlow Lite [9], PyTorch Mobile [10], or ONNX Runtime [11] can be used to optimize the models' memory footprint and runtime using techniques such as quantization, pruning, and layer fusion. *EdgeML* can also be supported by using specialized HW for ML acceleration on edge, including application-specific integrated circuits (ASIC) and Field-Programmable Gate Arrays (FPGA).

Unlike the general-purpose Central Processing Units (CPU), ASICs are chips designed to address a specific functionality with a reduced set of operations. ASICs allow for reduced power consumption, higher speeds, and small footprints. Since model inference only requires a specific subset of operations, ASICs are the right approach to address use cases related to model inference. In fact, in the past years, ASICs designed for accelerating model inference have become increasingly popular, e.g., the Coral Edge TPU [12] and Intel's Movidius VPU [13].

FPGAs allow for re-programming the logic gates on the chip after the manufacturing process. This flexibility allows for quickly optimising a chip for a specific model using a HW description language such as Verilog or VHDL. This added optimization on top of what is provided by an ASIC is a powerful tool for supporting ML on the edge, especially when the model may require to be updated over time or cost rather than performance is in focus.

8.3.1 Applications of $EdgeML$ in Industrial IoT

EdgeML can be applied in any use case where network bandwidth consumption, latency, offline functionality, or data privacy is a concern. In an industrial IoT context, it is often important to optimize for at least a few of these aspects, making *EdgeML* perfectly suited for such problems.

For example, consider the **predictive maintenance** use case in a remote oil or gas rig [14]. To ensure low downtime and maintenance costs, IoT sensors installed on the equipment can be used to gather information and predict when the system is close to failure using ML models. Operators can then be notified to ensure the issues are addressed in time. Due to the remote nature of such systems, a reliable internet connection is not always an option, and even when it works, the bandwidth and latency of the connection cannot be guaranteed. Due to these reasons, it is not ideal to set up a predictive maintenance use case using a centralized ML solution as its benefit (the early warning of potential system failure) is limited by the quality of the communication connection. If the model is unavailable during the timeframe where an upcoming failure could have been identified, the system may break, and the model would not have accomplished its task.

Another application of *EdgeML* is in the manufacturing domain for the **automated control of cyber-physical systems** such as robots [15]. For example, a robot could rely on a vision component to identify and localize the position of an item on a conveyor belt. Using this info, it would then interact with the part in some way, such as grabbing and moving the part to a different location. Due to the real-time info needed for controlling the robot in such a dynamic environment, the controlling system cannot rely on the long communication latency associated with centralized ML. Running machine vision models on edge will ensure that the info required for making the split-second decision is available with as low latency as possible.

Finally, the application of **automated quality assurance** (QA) in a manufacturing process can also benefit from *EdgeML* [16]. Standard QA processes require manual inspection, which slows down the throughput of

the factory or reduces the number of items that can be inspected. Manual inspection can be replaced by automated QA processes, which utilize ML models for quickly identifying defects. To ensure that the QA process is not a bottleneck in the system, *EdgeML* can be utilized to perform evaluation in real-time. Furthermore, by not sending any data to a centralized location, sensitive data about the manufacturing process does not leave the factory, ensuring the security of trade secrets.

8.3.2 Challenges in *EdgeML*

EdgeML brings its own unique challenges, which are not present in a centralized ML setup [17]. These issues arise from the distributed network of low-powered devices and lack of direct control over the data.

One challenge is related to fine-tuning of the ML model on device. Depending on system setup, it may be beneficial to adapt the global model for each device to make the predictions more relevant. To support this fine-tuning process, the edge devices must be (i) powerful enough to run the model training process in a reasonable amount of time, and (ii) they must have the capability to store and label data locally. The first issue can be addressed by using more powerful HW such as ASICs or FPGAs. Unfortunately, the latter issue is not as straightforward to address. Generating the set of ground truth labels required for training a model can be a challenge, as this cannot always be automated without human intervention. For example, it is difficult to fine-tune computer vision models on edge, as human effort is often required to generate the necessary labels for training (e.g., class, bounding boxes, or segmentations). When training a model centrally, there is the opportunity to generate labels by hand, something that is not always possible on device.

The problem of generating ground truth labels not only affects the ability to fine-tune models locally, but also makes monitoring model performance on edge harder. Most model prediction performance metrics (e.g., accuracy, recall, or mean squared error) rely on ground truth information. As such, other aspects of the system must be monitored as a proxy to prediction performance. Monitoring is a key component in any production ML system, as the real world is not static, meaning model performance may degrade over time. One cause for model performance degradation is concept drift, or the idea that the underlying properties of what is being predicted may change over time. For example, the performance of an automated QA model may change as the quality of the data from the input sensors degrade over time. By monitoring model performance over time, performance degradation can be quickly identified, triggering a model retraining cycle if necessary. Once

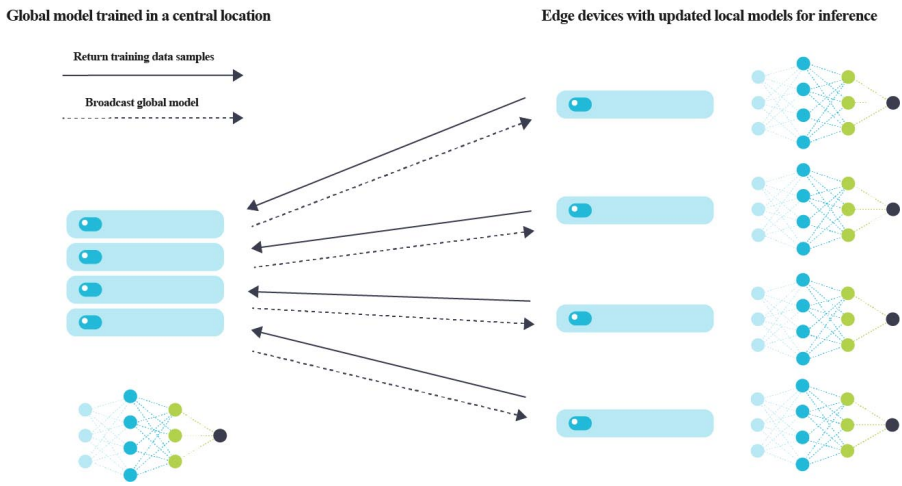

Figure 8.1 The global model is first trained in a central location and then broadcast to edge devices for inference. Edge devices can return data samples to train and update the global model.

the global model has been updated, it needs to be pushed to edge devices for inference. Adding to the system a module that supports over-the-air updates will help facilitate this process (see Figure 8.1). Furthermore, it is beneficial to follow SW deployment best practices, such as A/B Testing, when rolling out model updates to ensure that system stability is not affected. In the case of a model update performing poorly in production, it should be easy to roll back the changes and revert to the prior state.

While *EdgeML* alleviates the need to stream all data centrally for inference, the global model still needs to be trained in a central location before being pushed to devices for inference. To accomplish this, some data still needs to be collected centrally for constructing the dataset used in the training process. Therefore, *EdgeML* does not fully ensure data privacy, as some information still needs to find its way centrally. When data privacy is a major concern, neither centralized ML nor *EdgeML* are sufficient. Therefore, other techniques for training models in a privacy context, such as differentiable privacy [18] or FL [19], have been explored.

8.4 Federated Learning – A Solution to Train ML Models at Scale while Ensuring Privacy

In 2016, Google proposed a concept for training a model across a set of devices in a distributed way, which leverages the availability of data across

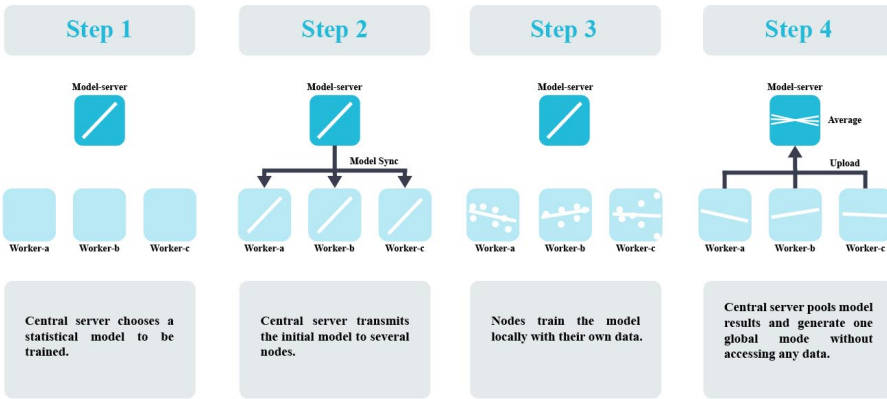

Figure 8.2 Visualization of the FL process. The four steps are executed consecutively and are repeated following the same process until the global model converges.

devices while still preserving privacy [20]. This approach, known as **Federated Learning**, ensures that no data ever leaves the device, and yet in the end of the training process, the output is a global model which can be used across devices.

The FL process is depicted in Figure 8.2, and it works as follows. In step 1, a first model design is chosen for training. This initial global model is distributed in step 2 to a set of devices known as clients or nodes. In step 3, each individual device trains the model on their local dataset for a certain number of iterations. The model updates are then collected centrally and aggregated into a single global model as part of step 4. The steps are then repeated following the same process until the global model converges. Finally, the newly trained global model is distributed to the different devices for performing inference on edge.

FL guarantees that the only info that leaves the device is the one about the model updates. When combined with *EdgeML*, the collected data never leaves the device, ensuring data privacy. This is a crucial aspect in industries like manufacturing, the energy sector, and Medical Technology (MedTech). In fact, *EdgeML* and FL complement each other to reduce bandwidth and improve data security.

8.4.1 Applications for Federated Learning in Industrial IoT

Due to its focus on data privacy, FL has suitable applications across several industries. Some of the most relevant applications for FL can be found in the IIoT sectors, including energy, manufacturing and MedTech.

In the European **energy sector**, FL has the potential to improve the stability of the grid and improve demand and supply forecasting. At the mid-voltage level, the current European electricity grid is split up into a group of distribution system operators (DSOs). Each DSO is independently responsible for their section of the grid and collaboration between DSOs is uncommon. Normally, a DSO will only interact with the transmission system operator (TSO), responsible for the highest voltage levels, to ensure stability and safety of the grid. DSOs are uninclined to share data with other DSOs or organizations as they may lose their competitive advantage. However, due to the safety-critical nature of the grid, all parties would benefit from some sort of cooperation. There is therefore potential for cross-silo (see next section) FL applications to train models across DSOs without sharing any sensitive information.

Another potential application of FL is in the **manufacturing** domain. Consider a company which produces machines used in factories across organizations spread throughout the world. It is in the interest of the machine's producer to provide the best possible product to its clients, and the integration of ML use cases is one potential avenue. It is therefore important for the models to have access to the wide base of machines in the field. However, due to the potential for trade secrets to be leaked, the clients who own the machines and the data are unlikely to want to share the information with the original manufacturer. By employing cross-device FL, the needs of both the system's producer and of the clients can be met.

MedTech is an additional application of FL in Industrial IoT. The wearable health devices domain could benefit from the application of ML, however the collection and analysis of information such as blood pressure or insulin levels in a central location are heavily regulated. This makes the application of centralized ML or *EdgeML* infeasible, as the data must always remain on edge. Cross-device FL is one solution to support the training of models across a large set of wearable IoT devices while staying in line with the regulations.

8.4.2 Federated Learning Scenarios

FL can be split into distinct categories depending on the use case and the topology of the system in focus.

The first differentiation that can be done is cross-device vs. cross-silo.

Cross-device FL considers a large network of low-powered clients with limited compute resources. A client could be a phone, a microcontroller,

an embedded system, or any other low-powered device. Depending on their usage, these devices may not always be available to perform the resource intensive training process. For example, not to bother a user, a phone may only be available for training during night-time while being charged and connected to Wi-Fi. Due to the low availability and reliability of each client, a subset of clients should be selected for each round of training. This subset should be sampled from a representative distribution of the clients to not bias the model towards clients with a higher availability. Furthermore, it is expected that some of the selected clients are unable to complete training within a predefined amount of time. This drop-out rate should be accounted for in each round in the selection of clients.

Cross-silo FL considers a much smaller network of clients compared to cross-device FL, each one representing an organization or data silo. As a result, it is expected that each client is a reliable, high-powered compute instance in the cloud or on-premises. Due to this stability, we can assume that every client will be available for training in every round, and there will be an extremely low drop-out rate. Unlike cross-device, there is no need to subsample clients during each round of training.

FL scenarios can also be differentiated by how the data is split across clients (see Figure 8.3).

Horisontal FL (also known as **Homogenous FL**) concerns the case where each client has the same set of features, but there are different examples/datapoints per client. This scenario applies for example to the manufacturing use case described in the previous section 8.4.1, where the distributed machines all collect the same kind of information, but the datapoints are relative to the specific context of each machine.

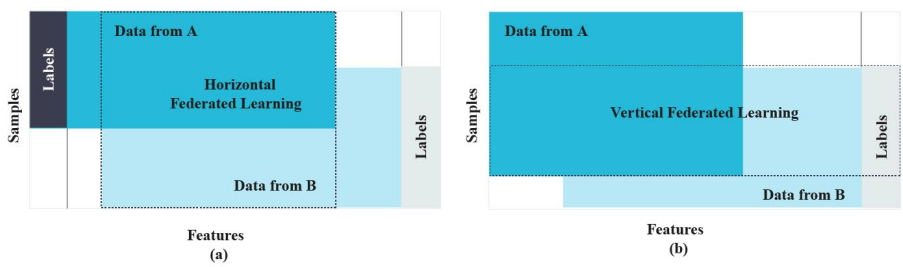

Figure 8.3 FL scenarios according to how the data is split across clients. (a) Horizontal FL. (b) Vertical FL.

Vertical FL (also known as **Heterogenous FL**) concerns the case where different clients have different subsets of features, but they share the same set of examples/datapoints. Due to the examples being shared across clients, special approaches need to be used to ensure that we can still train models while preserving data privacy. One promising approach for supporting vertical FL is secure multi-party computation [21].

8.4.3 Challenges in Federated Learning

A first set of challenges are related to the focus on data privacy. Since data is never sent to a central repository, standard ML tasks related to training and evaluating models become much more difficult to accomplish. Normally, a data scientist would start by performing exploratory data analysis to get a better understanding of underlying distribution of the data they are working with. However, standard data exploration is not possible in an FL context due to the lack of direct access to the data. Luckily, approaches such as federated analytics can be utilized to get an aggregated understanding of statistics about the data across clients [22]. Unfortunately, these approaches cannot fully replace the information and understanding you can get about the data in a centralized ML context.

As mentioned in the previous section 8.3.2, the challenge of generating ground truth labels for model training and evaluation on edge also exists in FL. Ground truth labels need to be generated by each node/client, as they need labels to train a model. However, due to this requirement, evaluation of models in FL becomes easier compared to *EdgeML*, as the standard evaluation metrics can be calculated on the predictions of the trained model, with the caveat that approaches such as federated analytics should still be employed to ensure that data privacy is kept.

Another challenge that ML engineers face when training a model in a FL context is the fact that the independent and identically distributed (i.i.d.) assumption no longer holds. The statistical properties of the data per client are potentially different, leading to possible sources of bias. Algorithms such as SCAFFOLD attempt to address this issue when sampling the clients for the federation and during the aggregation of the model updates [23]. Nevertheless, model convergence in a FL context may not be as good as when the model is trained centrally on the full dataset.

Preventing adversarial actors in the system is another major challenge in FL. While the data never leaves the clients, there is still potential to extract information about the training data from the individual model updates [18].

Therefore, additional steps should be taken to ensure the trust of the model aggregator. One approach to account for this is to apply the concept of differential privacy [18]. Furthermore, it is also possible for untrustworthy clients in the federation to poison the resulting model by injecting bias [24, 25]. Necessary steps should be taken to ensure that the integrity of the model and of the system is maintained.

Standardizing the data interface in the cross-silo FL case is another challenge which needs to be addressed. It is often the case that data infrastructure and schema may be different across organizations and enforcing a single format for training can be a major data engineering challenge. To support training, either the individual silos must agree on a shared data format, or the centralized entity should enforce a schema on all members of the federation. Exceptional care must be taken to ensure that the formats align, because if there are differences, the model may not be able to converge to a performant solution.

8.4.4 Frameworks and products for leveraging Federated Learning

To leverage the benefits of FL and foster the research and development of novel methods, many frameworks and several products have been developed over the past few years [26–31]. The following briefly introduces the most relevant tools from proprietary and open-source domains.

In the open-source world, the current frontrunners are:

- TensorFlow Federated (TFF) is developed by Google as an extension to its TensorFlow framework [28]. TFF is aimed at research and only simulates the distributed setup of the data. Due to the close relationship to TensorFlow, TFF is not DL framework agnostic and therefore provides no support for other frameworks such as PyTorch.
- PySyft and PyGrid are developed by the OpenMined community [29]. The focus lies on approaches for computing on data you do not own (not just in a ML sense), including encrypted computations, differential privacy, and FL. PySyft is responsible for the ML abstractions and has a tight coupling with PyTorch. However, it does also offer support for TensorFlow. PyGrid works as intermediary to deploy PySyft workloads at scale across networks.
- Federated AI Technology Enabler (FATE) was initiated by Webank to enable big data collaboration while ensuring data protection regulation compliance [31]. FATE consists of several components, where Federated

ML implements many standard ML algorithms and supports both the TensorFlow and PyTorch frameworks. Given the original use case it was designed for, deployment is focused on cluster environments, meaning small edge devices are not in scope.

- OpenFL originates from a collaboration between Intel and the University of Pennsylvania to develop the Federated Tumor Segmentation platform [26]. Given its early focus on a real-world application, OpenFL can not only simulate a distributed/FL setup for research, but also handles deployment to physically distributed scenarios. It is also one of the few DL framework agnostic solutions, supporting model implementation in many different frameworks, including TensorFlow, PyTorch, and scikit-learn.
- Flower, currently under development by a German start-up [27], is a DL framework that is agnostic and lightweight in terms of setup and deployment. It provides the possibility to run simulated and real-world application workloads on different HW sizes, opening a wide range of usage scenarios.

In the proprietary world, the most used solutions are:

- NVIDIA Clara targets the healthcare sector and considers itself as an application framework [32]. This includes Graphic Processing Unit (GPU) accelerated libraries, SW development kits (SDK), and reference applications for developers, data scientists, and researchers alike. It is comprised of several components to cover the main steps of the ML lifecycle in a federated way.
- IBM Federated Learning supports multiple DL frameworks for model design [33]. It can handle different learning topologies and is aimed at enterprise and hybrid-cloud settings.

Overall, many frameworks still focus on the theoretical/research side of the problem, only simulating different clients and distributing data from a central location, thus running all the computation on the same system. When considering the non-proprietary solutions, we find that none of the existing solutions provide the necessary set of features for (enterprise) business applications while also being quick and easy to deploy. As such, there is unfortunately no single solution which can bring FL to a wider audience yet.

EdgeML and FL reduce communication complexity by limiting the amount of information passed to a centralized location. Reducing communication bandwidth is only one approach to support scalability with a growing number of IoT devices. Another approach to reduce communication complexity

can stem from focusing on improving the communication protocols on the receiver side. In the following section, we explore AI/ML approaches for reducing complexity in this context.

8.4.5 Reducing Complexity of RX Processing

In current communication systems, the receiver side is the most computationally intensive and therefore power consuming part. AI/ML methods are promising approaches to reduce the receivers' implementation complexity, allowing to improve systems by learning patterns and structures from data, rather than relying on human-made models to approximate the environment. Moreover, hand-crafted algorithms can be replaced by trainable ML algorithms that fully learn to solve the problem at hand using data and trainable parameters. As an example of applied AI/ML techniques, let's consider multiple-input multiple-output (MIMO) systems, in which detection aims to reconstruct parallel superimposed data streams received through multiple antennas at the receiver side. For MIMO detection, AI/ML have shown superior performance compared to model-based state of the art (SotA) approaches [34], [37]. On the receiver side, forward error correction (FEC) decoding is the most computationally intensive part, which also introduces additional latency caused by the needed iterative decoding schemes. In addition, short packets, which are common in machine communication systems very popular in manufacturing environments, reduce even more the performance of these decoders. In the context of FEC, the application of AI/ML has been explored to overcome the aforementioned problems and in the following we present recent achievements in the field of FEC using AI/ML.

Neural Network-based Decoder: A first idea to overcome the mentioned drawbacks is to make use of AI/ML techniques in SotA decoders and learn decoding directly from data only with the help of a neural networks (NN) [38]. A NN usually is a nonlinear function with trainable parameters/weights that can be adapted by processing data with Gradient Descent methods. As data input we have the received signal and as output we get the decoded information words. The weights are iteratively adapted so that the NN decoder is as close as possible to the original transmitted information words. Unfortunately, this approach cannot be practically deployed in real-world scenarios, as the number of required training samples grows exponentially with the length of the information word, and it is even not possible in machine communication systems when the length of the packets becomes too large.

Unrolled Belief Propagation: A way to overcome such limitations is the use of model knowledge about SotA decoders. One approach is based on the iterative Belief Propagation decoder, which however is suboptimal and whose performance decreases for short block lengths. By fixing the number of iterations of this decoder, a fixed structure is obtained, and trainable weights can be introduced into the structure. Therefore, such structure can be trained like an NN so that the performance degradation can be reduced and scaled for longer block lengths [39].

Auto-NN Turbo Decoder: Another way is to incorporate model knowledge is the structure of turbo codes [40]. A Turbo encoder is set up on the transmitter side and a NN is used for decoding. The structure of the decoding NN follows the structure of the turbo decoder, and it was shown that this approach can achieve good performance even for longer block lengths [41].

An extension of this idea is to use also an NN to encode and form an end to end (e2e) system. This is a so-called autoencoder, since the input of the encoding NN is the information word and the output of the receiving NN is in turn the information words, so that this e2e chain effectively forms an identity function. The main difference from a purely data-driven approach is that the structure of the encoding NN and the decoding NN is based on the turbo encoder and the decoder structure. Taking advantage of this knowledge, the resulting Turbo autoencoder [42], [43] can scale to larger block lengths, but not as well for large block lengths.

To reduce the complexity and latency of the FEC decoding, we present two concepts that utilize the benefits of AI and incorporate knowledge of SotA approaches to combine the benefits of both worlds.

NN-based Forecasting: A first approach is to use ML with the aid of a NN to predict the decoder success of SotA decoders, which we named NN-FoC [44]. This is done by inserting an NN into the receiver chain that directly uses the received signals to predict whether the decoder will be able to correctly decode the received packet. Subsequently, the decoder is executed only if the NN predicts a likely decoding success. In addition, this prediction directly enables the marking of packets as acknowledged or unacknowledged. This enables an "Early Automatic Repeat Request (E-ARQ)" and directly triggers retransmission in case of erroneous packets.

In Figure 8.4 the efficiency η for a standard ARQ scheme in comparison to the proposed NN-FoC forecasting with E-ARQ and different decoder delays κ is shown. The proposed NN-FoC can increase the efficiency in comparison to the Standard ARQ schemes for all decoder delays. In comparison to a

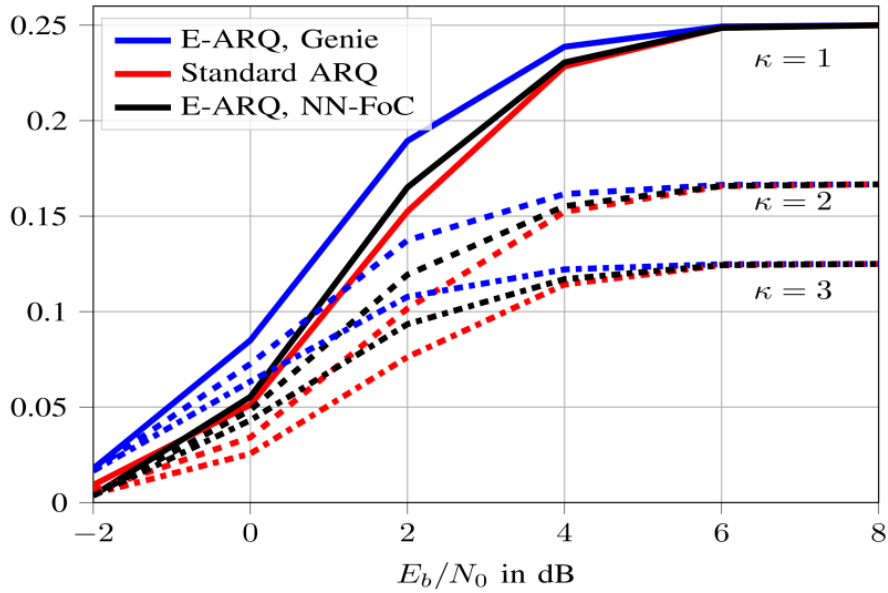

Figure 8.4 Efficiency η over SNR for standard ARQ scheme in comparison to E-ARQ with NN-FoC forecasting and a Genie forecaster for different decoder delays κ

Genie, non-practical, forecaster, a performance gap against the proposed NN-FoC approach is visible. This approach can hence avoid unnecessary decoder executions, reduce latency, and save computational power. Our analysis was limited to codes with very short block lengths; therefore, an extension to longer codes is still an open research question.

Low-Resolution Decoder: From the implementation point of view, the bit-resolution of the decoder is a significant bottleneck, limiting the possibility for efficient HW implementations, especially for codes with a large number of interconnections [45]. Hence, decoders with very-low bit resolution are a necessary element for receiver implementations that aim to fulfil the high requirements of future standards [46].

In SotA soft decision decoder implementations, the complexity is reduced by replacing intensive node operations with simpler approximations and by reducing the bit-resolution of internal variables via quantization. In recent literature, systematic design approaches of finite alphabet decoders gained a significant attention due to its potential to outperform SotA decoding algorithms in terms of error correction performance and implementation complexity.

A novel systematic approach is to design finite alphabet decoders with very low bit resolution and operations that aim to maximize mutual information [47]. This approach is directly related to the Information Bottleneck Method (IBM) [48], [49], which is a novel clustering approach in the context of unsupervised learning that provides a generic approach for the learning of discrete decoders with very-low bit resolution (e.g., 3-4 bit) and replaces all internal node operations by look-up-tables (LUTs). This LUT-MP decoder approach enables the implementation of efficient high throughput decoder implementations [50], [51]. Further improvements on the efficient implementation of information optimized LUTs by using low-range integer calculations are still under investigation [52].

8.4.6 Enhancing Reliability by Multi-Connectivity in the Uplink

Manufacturing and industrial applications place very high demands on the communication system. In particular, a very reliable exchange of information with low latency must be achieved. SotA control applications with periodic communication tolerate several consecutive message errors before stopping. To avoid or reduce costly downtimes, the Radio Access Network (RAN) needs to be designed accordingly, following the always growing number of features that appear at teach new generation of the telecommunication systems [53].

The dense deployment of access points (APs) is a very promising approach in the industrial environment to meet these stringent requirements since it improves significantly the average channel quality between the user equipment and the overall RAN infrastructure. In addition, joint processing of multiple APs allows exploitation of centralization gains, but also places additional burden on the communications infrastructure [54, 55]. To this end, the base station functionality can be divided in 5G networks into three elements [56]:

- Central unit (CU) contains higher layer functions such as RRC and PDPC
- Distributed Unit (DU) containing RLC and MAC as well as some PHY layer functions
- Radio Unit (RU) containing the lower layer PHY functions.

This approach facilitates RAN virtualization with flexible assignment of computing resources across the three different network entities. The physical location of these network entities depends on the specific architecture and available geographical locations. The functional split determines which

protocol stack functionality is executed in which of the three units. In a RAN system with distributed RUs and shared information processing in the DU, information about the received signals must be transmitted from the RUs to the DU via rate-limited fronthauls (FH) for uplink communication. The direct forwarding of I/Q receive signals from the antennas would lead to very high FH data rates [57]. Instead, it is more meaningful to perform pre-processing of the receiver signals in the RUs and limit the FH data rate by forwarding only the necessary amount of data required for successful detection in the DU.

As discussed in the previous section, IBM has successfully been used to learn FEC decoder implementations with reduced complexity. Here, we focus on the ML-based design of quantization schemes and the combination of discrete signals with varying statistics in the DU.

Information Bottleneck Quantization: we consider the RAN system in Figure 8.5 with J APs observing the user equipment of interest. In the APs the noisy observations are pre-processed (e.g., transformation to frequency domain, sub-carrier wise equalization for OFDM and fine pre-quantization [58]) yielding the local observation y_j for the transmitted symbol x with statistical relation given by the conditional probability mass function, $p\left(y_j \mid x\right)$. Prior to forwarding the local observations to the DU, the observations y_j are compressed to reduce the FH data rate. As a joint quantization of all receive signals $\{y_1, y_2, \ldots, y_J\}$ is not feasible in practice, the observations $y_j \in \mathcal{Y}_j$ are individually compressed to the messages $z_j \in \mathcal{Z}_j$ from the discrete alphabets $\mathcal{Z}_j$ with $|\mathcal{Z}_j| \ll |\mathcal{Y}_j|$ by the local quantizer function $z_j = Q_j\left(y_j\right)$. A joint design of the local quantizers $\{Q_1, Q_2, \ldots, Q_J\}$ would be desirable and details can be found in [59], [60]. Here we just

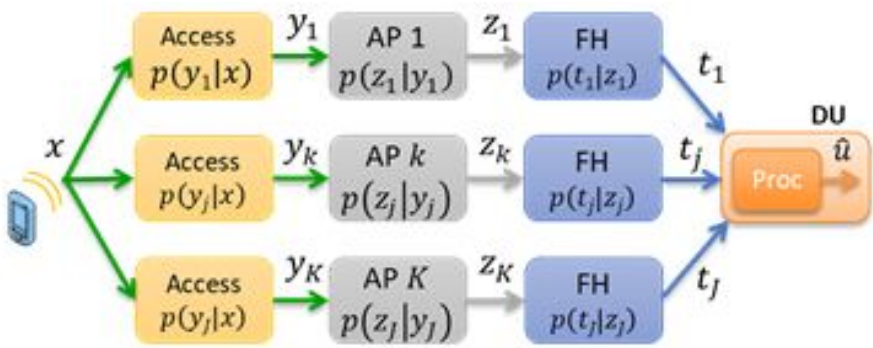

Figure 8.5 Distributed communication system with **J** access points forwarding compressed messages to the DU.

mention an independent design of the local quantizers Q_j per branch j such that the mutual information (MI) $I(x;;z_j)$ between the source symbol x and the quantizer output z_j per AP is maximized for a given source distribution $p(x)$

$$Q_j^\star = \underset{Q_j \in \mathcal{Q}}{\operatorname{argmax}} I(x;;z_j) \quad \text{s.t. } |\mathcal{Z}_j| \leq N_j \tag{8.1}$$

$\mathcal{Q}$ is the set of all possible quantizer mappings and N_j denotes the upper bound on the cardinality of the set $\mathcal{Z}_j$. By limiting the cardinality N_j, the FH rate of AP j is bounded by $R_j \leq \log_2 N_j$ such that rate limitations of individual FH links can be considered by choosing N_j. The objective in (8.1) is a special case of the IBM [48].

Forward-Aware Vector Information Bottleneck (FAVIB): If the FH links are not only rate-limited, but also introduce transmission errors such that the message t_j received by the DU on the FH link j can deviate from the transmitted message z_j, it is favourable to incorporate the statistic of the FH link already in the design of the quantizers. To this end, the objective function is adapted by maximizing the MI $I(x;;t_j)$ between the source symbol x and the receive signal t_j per AP at the DU. The FAVIB method presented in [60] achieves a generalization of the IBM method by e2e data rate optimization considering error-prone FH by the objective function

$$Q_j^\star = \underset{Q_j \in \mathcal{Q}}{\operatorname{argmax}} I(x;;t_j) \quad \text{s.t. } |\mathcal{Z}_j| \leq N_j. \tag{8.2}$$

With increasing FH error rate, the number of clusters in $\mathcal{Z}_j$ carrying most of the information about the source decreases and some clusters are allocated with vanishing probability. This trend can be interpreted as a type of inherent error protection performed by the quantization scheme. Similarly, the impact of error-prone FH links can be incorporated in the joint design of distributed quantizers [61].

Relative Entropy based Message Combining (REMC): The choice of each individual quantizers Q_j depends on the access statistic $p(y_j \mid x)$, the cardinality N_j and the FH channel statistic $p(t_j \mid z_j)$. Thus, even if same messages arrive at the DU on two different FH links, their individual meaning regarding the source message can be different. Consequently, the combining step in the DU needs to incorporate the actual meaning of the messages t_j in order to fully exploit the spatial diversity. The REMC approach [62] performs

a clustering of messages with similar meaning $p\left(c_\nu \mid t_1, t_2, \ldots, t_J\right)$ regarding a given decoder design distribution $p^*(c|r)$ by

$$r_\nu = Q_{C,\nu}\left(t_1, t_2, \ldots, t_J\right) = \arg\min_{r\in\mathcal{R}} D_{\mathrm{KL}}\left(p\left(c_\nu \mid t_1, t_2, \ldots, t_J\right) || p^*(c|r)\right). \tag{8.3}$$

Performance Evaluation: A comparison between the 3-bit LUT-MP and the 4-bit LUT-MP decoders from a previous section for a 6-bit channel quantization is shown in Figure 8.6. The 4-bit LUT-MP achieves at a BER of 10^{-3} a performance gain of ≈ 1 dB for $J = 1$ and ≈ 0.6 dB for $J = 2,\ 3$. The performance improvement can be further increased by increasing the number of bits of the LUT-MP. Hence, the e2e performance by using a low-bit resolution for the forwarding of I/Q data via the FH and the joint processing at the DU (REMC and LUT-MP decoding) is very close to the benchmark without quantization and floating-point implementation of the sum product algorithm (FP-SPA). Thus, distributed APs with joint receiver processing has been demonstrated to realize high-reliable communication by exploiting spatial diversity. The IBM-based compression for distributed APs allows for separated compression at APs while meeting the e2e requirements with low total FH data rate (only 6 bits per receive signal) and only 3 or 4 bit-resolution of the decoder.

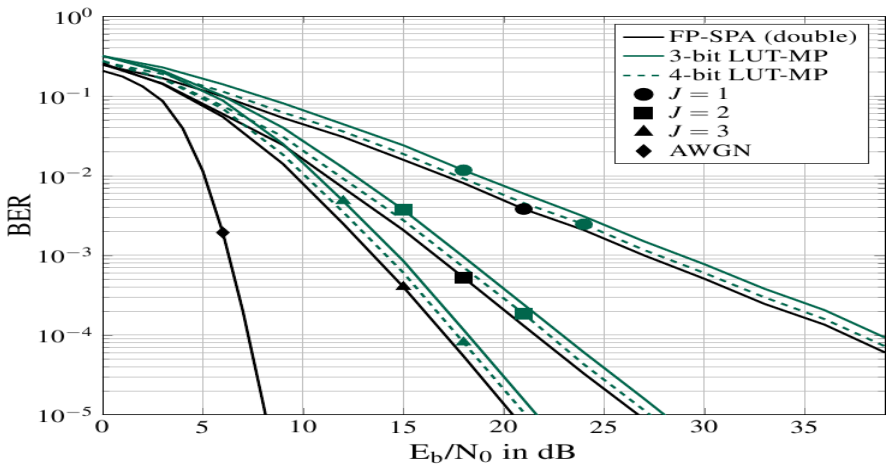

Figure 8.6 BER performance for 16-QAM with RAPs applying SNR-adapted 6-bit quantizer per AP and REMC in DU for $\mathbf{J} \geq \mathbf{1}$.

8.5 Communications in an "Embodied Artificial Intelligence" Future

By 2030 we can expect wireless networks with terabits-per-second connectivity, paired with compute power equivalent to that of the human brain. Machines will independently offer and consume complex services on Internet platforms that operate according to platform-economic business rules. These human-like capabilities will also lead to completely new possibilities in the way machines communicate with humans and other machines. In this section we discuss which opportunities and technical requirements will arise from these future requirements and possibilities. It is argued that there will be a strong transformation from constant networking to the principle of "conversations", where context and experience are considered. At the same time, future wireless technology will offer new functions in addition to communications, which will allow to optimize the use of limited resources like energy, raw materials, space, time and frequency per application.

Many companies in industrial markets, such as capital goods, are undergoing a fundamental transformation from sellers of machines to providers of services, offering their customers integrated solutions consisting of goods and services as integrated value propositions [63]. Driven by synergies between technological advances and the widespread use of mobile devices, data science and the IoT, the ability to connect remotely to physical devices has spawned radically new types of services [64]. Smart products have become enablers for the delivery of smart services. They can both collect and analyse field data and make decisions and act autonomously, thus changing the design of services and business models [65].

Establishing a platform business model currently represents a particularly promising strategy for achieving market leadership. The pipeline business model – "creating value by controlling a linear series of activities" [66], traditionally implemented by many manufacturers, is being fundamentally challenged. At the same time, digital platforms go beyond the co-creation of value with customers propagated in service theory by using two- or multi-sided marketplaces that enable different types of users to interact with each other and carry out transactions. Given the success of platform business models, it is not surprising that companies with product-oriented business models, as well as manufacturers looking to evolve into smart service providers, are considering adopting platform business models. Companies' interest in this topic also stems from the observation that competition between platforms on the same market can lead to a winner-takes-all outcome under certain

conditions [67] and those early movers can gain a significant advantage [68]. In the future, users will mainly be end consumers and machines that are able to autonomously offer services on a platform like human users. These so-called ***embodied intelligence (EI) machines*** act as providers of intelligent services.

8.6 Embodied Artificial Intelligence

According to Cangelos [69], EI is the manifestation of intelligent behaviour in embodied and situated agents in conjunction with a strict coupling between the agent and its environment (situatedness), mediated by the constraints of the agent's own body, perceptual and mobile systems, and brain (embodiment).

According to Klocke [70], intelligent agents are autonomous systems that perceive, decide and act on their own. They are characterized by properties such as the ability to learn, logical reasoning, creativity and sometimes also initiative, which are more like human intelligent behaviour than functionalities of conventional computer programs. In human-computer interaction, so-called interface agents increasingly operate to mediate between humans and computer systems, often unnoticed by the user. One of the most important tasks of intelligent agents is to search for and store information in the world in which they operate. Every decision, just as with humans, is based on information and knowledge. Every agent, whether human or SW, must have distinctive capabilities and algorithms to search for information and store it as knowledge, the human in the brain, the SW in the computer memory.

Given this background, the ability to learn and the associated expandability of the functional and action space is of particular interest. For this purpose, it is important to understand the learning process or the life cycle of cognitive systems, which is depicted in Figure 8.7. Such systems should be able to capture the environment and the respective situation with the help of embodiment, for example through suitable sensor technology or the body itself. In the further course, the captured information and data points must be processed appropriately and provided with meaning and semantics. The transformed knowledge is then transferred into models and possible options for action, strategies and solution spaces are derived and evaluated. From the different options, depending on the own objectives, the most promising variant for the system is selected and the implementation or the interaction with the environment is started. Finally, the essential step of learning from one's own behaviour and the actions and reactions of the environment begins,

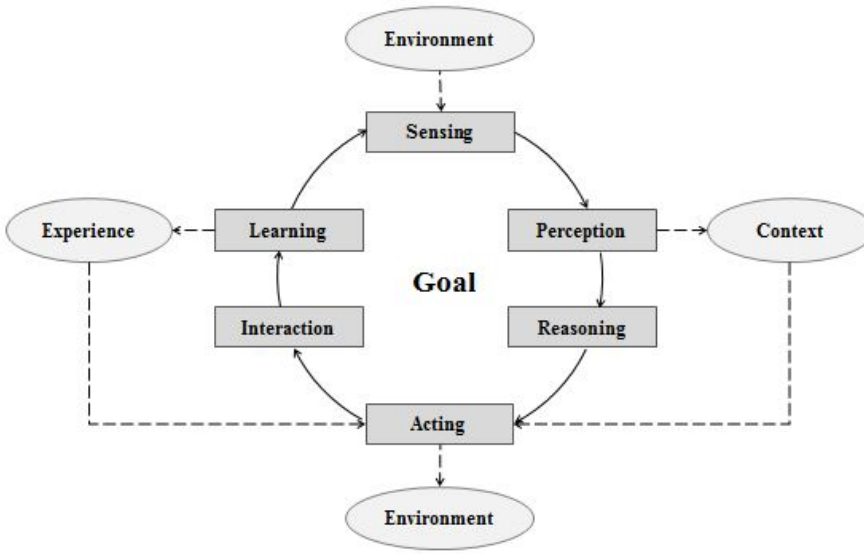

Figure 8.7 The cognitive cycle of an embodied intelligence agent.

which are first observed to learn from them and to reflect on what has been experienced. In this way future EI Things will interact in and with the platform ecosystems, build up a knowledge base and realize their goals better and better.

Wireless connectivity, and in particular, device-to-device links (in context of cellular networks also referred to as "sidelinks" [71]) will be key facilitator for local distribution of information needed to make ML agents work together autonomously. However, transmitting raw sensor data (e.g., from cameras) to agents running in a centralized data centre will unlikely be sustainable on large scale, given the steady growth of the number of ML systems in professional and private environments. To address future needs, communication networks will push the performance boundaries and expand into new frequencies. Supplementary, each EI agent will collect a-priori information specific to its task, physical and communication environments, which can be used to reduce the amount of exchanged information between collaborating autonomous IoT systems. Federated ML and means for model sharing are first steps in this direction, as touched in previous sections [72–73]. Due to their distributed nature, these approaches are a good match for edge architectures. However, limitations of the underlying communications network also need to be considered, when deciding how information is

represented, what is shared and how it is propagated through a network [74–77]. In this context, key research directions are i) how to collect and represent context information, i.e., knowledge about an application and its physical and wireless environment, and ii) how to build, represent and share experience for collaborating EI agents under dynamic, constrained, and unreliable communication conditions.

8.7 High Integration as a Central Technological Driver

An EI agent is usually a highly integrated system, i.e., a system that tightly integrates various previously independent components into one physical body. In addition to a purely physical integration, these components are also strongly coupled with each other in terms of energy and communication. However, the inter-connection of the components is not rigid, but flexible, mostly depending on the realized application. The installed components can therefore also serve purposes that are different form the ones conceived at system design time. This is facilitated by generously overprovisioning the components in terms of performance and capabilities, rather than them being derived from a limited set of fixed features in the sense of a "design to cost". This design approach leads to minimal functional costs in the overall view of all applications realized with the system. As a result, the high integration of machines will displace various existing solutions or even make them obsolete. Ultimately, a system with integrated functions will prevail over a composite system with subsequently added function groups, in which synergies can usually only be created at considerable expense, while performance will remain the same or even improve. The logical next step of high integration is therefore EI. In the Stanford Encyclopaedia of Philosophy, the once insignificant movement of embodied cognition is now said to be well known. Unlike, for example, ecological psychology [78], which has had to fight an uphill battle for acceptance by the public, embodied cognition has gained a large following. EI has been the subject of numerous articles in popular media. Moreover, there is no area of cognitive science-perception, language, learning, memory, categorization, problem solving, emotion, social cognition, that has not been given a makeover by EI [79].
One example of high integration of functionality can be observed in the millimetre waves (mmW) frequency bands shown in Figure 8.8.

The frequency range above 100 GHz holds the potential for channels with large, aggregated bandwidth. For communication systems, large bandwidths carry the promise of increased data rates, higher traffic capacity

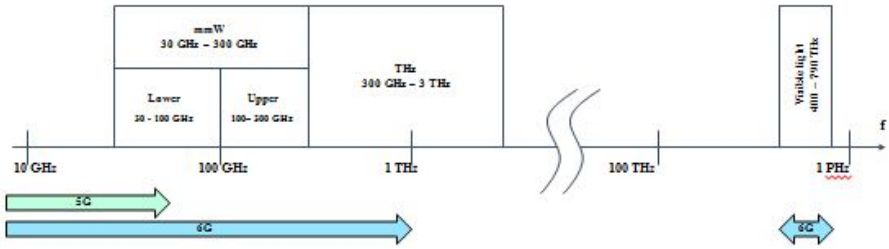

Figure 8.8 Overview of mmW frequencies. 5G bands expand up to 50 GHz, 6G is expected to reach 1 THz and also include visible light communications.

and connection density, finer frequency and time resolution for environment sensing and potentially a lower latency. Shorter wavelengths bring altered properties for the interaction of radio waves with the matter in our environment and make trade-offs between smaller form-factor steerable antenna arrays and link budget possible. This brings also great opportunities for capturing the (physical) environment with radio waves, which in future will no longer be a by-product but a design target. High resolution of multipath signal components and fine-grained beamforming are the foundation for better localization, mapping and tracking of devices and objects. Covering a large range of frequencies with a radio brings us closer to be able to explore the physical properties of our environment with spectroscopy. (More details can be found in [80–84]). The functionality needed from the underlying wireless technology to achieve this can be broadly categorized into the four functional areas "short range wireless connectivity", "long range wireless connectivity", "sensing with radio waves" and "wireless energy transfer". An overview is given in Figure 8.9.

The traditional small-cell scenario with typical cell size below 100 m is considered as **short-range wireless connectivity** for mmW frequencies (30 – 300 GHz). In contrast to previous generations of cellular systems, emphasis on differentiated optimizations for smaller ranges is expected in 6G. Short-range transceivers capable of operating in the upper mmW frequencies will allow future communications systems to expand into new frequencies. In addition to data rate, also traffic and connections per area (i.e., capacity and density) will generally benefit from access to these new frequencies. Additionally, the increasing signal attenuation at higher frequencies gives the opportunity to deploy dense networks of smaller cells. High directivity of the transmissions with narrow beams allows to further optimize the utilization of communication resources. Altogether, these properties will also provide

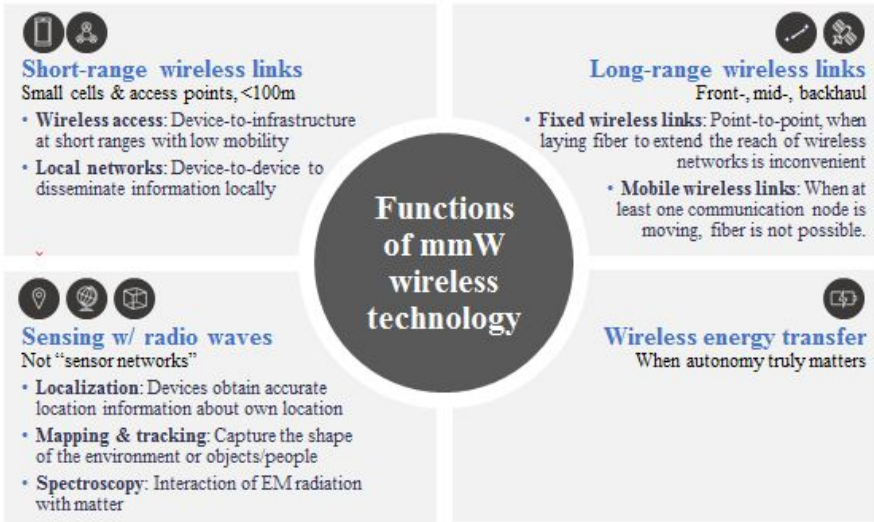

Figure 8.9 Overview of the functions of mmW wireless technology.

the means to transport data from sensors/displays/actors to the processing and back and hence help facilitate the integration of services offered by local compute nodes.

Communication links at distances beyond 100 m are considered as **long-range wireless connectivity** for mmW frequencies. Traditional applications include directional radio (point-to-point) links across a few kilometres, while emerging scenarios might necessitate link distances of up to 1000 km. In general, more available bandwidth for wireless x-haul (fixed/integrated) will increase achievable and peak data rates and capacity**.** Additionally, the mmW frequencies are expected to play an increasing role for wireless backhaul links from and between moving entities like satellites, high-altitude platforms, or swarm-networks, which will be integral for extending the global reach (coverage) of cellular networks [85].

With respect to location accuracy and **integrated sensing capabilities**, large signal bandwidth leads to better resolution of multipaths. The rapidly steerable antennas with strong directivity, necessary at frequencies beyond 100 GHz to overcome path loss, bring the benefit of increasing the spatial resolution for localization purposes. And lastly, decreasing the wavelength changes how radio waves interact with matter in the physical world. This can be exploited for 3D mapping of the environment and for detecting human gestures in a manufacturing domain.

EI systems will only become truly autonomous when energy is always available everywhere and. Already today, energy harvesting from the environment can complement the traditional wired charging of batteries. **Wireless energy transfer** (at distances beyond a few millimetres) from infrastructure to devices and among devices will become increasingly important in future. Advances of mmW technology will pave the way towards ubiquitous wireless energy transfer, as the size of antenna arrays shrinks, and the number of antenna elements grows inverse to the operating frequency. This opens new possibilities to focus the emitted electromagnetic radiation in a single direction with beam-/spot-forming algorithms.

These functional areas can also be addressed with optical communication technology operating in the visible light spectrum, which will play a complementary role in the advancement of wireless communication networks.

8.8 Conclusion

The trend towards platform economies continues to disrupt traditional business models. In future, platforms will not only serve humans but also machines. The communication behaviour of such machines will change from long range and broadband to short range and context-based, from permanent data collection to focused and directed information exchange. This will be facilitated by additional non-communication functions integrated in future wireless technology and will impact broadly all manufacturing related scenarios.

Acknowledgements

Part of this work was partly funded by the German ministry of education and research (BMBF) under grant 16KIS1180K (FunKI) and 16KIS1012 (IRLG).

Part of this work received funding from Germany's Federal Ministry for Economic Affairs and Energy under grant agreement No. 01MT20005 (BIG – The next big thing in Embodied Intelligence).

Part of this work has been funded by the European Commission through the H2020 project Hexa-X (Grant Agreement no. 101015956).

References

[1] Number of internet of things (IoT) connected devices worldwide. Statista, 2021. Available online at: https://www.statista.com/statistics/802690/worldwide-connected-devices-by-access-technology/

[2] S. Goodman, "Industry 4.0: How Cisco is Enabling the Future of Manufacturing". White paper, 2019. Available online at: https://www.cisco.com/c/dam/en_us/solutions/industries/manufacturing/white-paper-c11-742529.pdf
[3] U.S. Shanthamallu, et al. "A brief survey of machine learning methods and their sensor and IoT applications" 8^{th} International Conference on Information, Intelligence, Systems & Applications (IISA), 2017.
[4] Z. Zhou et al., "Learning-Based URLLC-Aware Task Offloading for Internet of Health Things," in IEEE Journal on Selected Areas in Communications, vol. 39, no. 2, pp. 396-410, Feb. 2021, doi: 10.1109/JSAC.2020.3020680.
[5] M. Merenda, C. Porcaro, D. Iero. "Edge machine learning for AI-enabled IoT devices: A review" Sensors 20.9 (2020): 2533.
[6] Gartner Identifies Four Trends Driving Near-Term Artificial Intelligence Innovation. Gartner, 2021. Available online at: https://www.gartner.com/en/newsroom/press-releases/2021-09-07-gartner-identifies-four-trends-driving-near-term-artificial-intelligence-innovation.
[7] ESP32. Available online at: http://esp32.net.
[8] Raspberry pi. Available online at :https://www.raspberrypi.org.
[9] TensorFlow. Available online at: https://www.tensorflow.org/lite.
[10] Pytorch. Available online at: https://pytorch.org/mobile/home/ .
[11] Onnx. Available online at: https://onnxruntime.ai.
[12] Coral. Available online at: https://coral.ai/products/.
[13] Movidius-VPU. Available online at: https://www.intel.com/content/www/us/en/products/details/processors/movidius-vpu.html.
[14] A. Katona, P. Panfilov, B. Katalinic "Building predictive maintenance framework for smart environment application systems". Proceedings of the 29^{th} DAAAM International Symposium. 2018.
[15] N. Jazdi "Cyber physical systems in the context of Industry 4.0". 2014 IEEE international conference on automation, quality and testing, robotics (AQTR), 2014.
[16] U. T. Gamze, C. Davutoğlu, M. N. Durakbasa "Automated quality assurance applications in the rise of IoT". Proceedings of the International Symposium for Production Research 2019. Springer, Cham, 2019.
[17] G. Plastiras, et al. "Edge intelligence: Challenges and opportunities of near-sensor machine learning applications". 2018 IEEE 29^{th} International conference on application-specific systems, architectures and processors (ASAP), 2018.

[18] C. Dwork, et al. “Calibrating noise to sensitivity in private data analysis”, Theory of cryptography conference. Springer, Berlin, Heidelberg, 2006.

[19] T. Li, et al. “Federated learning: Challenges, methods, and future directions” IEEE Signal Processing Magazine 37.3 (2020): 50-60.

[20] J. Koneèn$\imath$, et al. “Federated learning: Strategies for improving communication efficiency”. arXiv preprint arXiv:1610.05492 (2016).

[21] A. C. Yao “How to generate and exchange secrets”. 27^{th} Annual Symposium on Foundations of Computer Science (sfcs 1986), 1986.

[22] D. Ramage, “Federated Analytics: Collaborative Data Science without Data Collection”, Google Research, 2020. Available online at: https://ai.googleblog.com/2020/05/federated-analytics-collaborative-data.html.

[23] S. P. Karimireddy, et al. “Scaffold: Stochastic controlled averaging for federated learning”. International Conference on Machine Learning. PMLR, 2020.

[24] E. Bagdasaryan, et al. “How to backdoor federated learning”. International Conference on Artificial Intelligence and Statistics. PMLR, 2020.

[25] A. N. Bhagoji, et al. “Analyzing federated learning through an adversarial lens” International Conference on Machine Learning. PMLR, 2019.

[26] G. A. Reina, et al. “OpenFL: An open-source framework for Federated Learning” arXiv preprint arXiv:2105.06413 (2021).

[27] D. J. Beutel, et al. “Flower: A friendly federated learning research framework”. arXiv preprint arXiv:2007.14390 (2020).

[28] TensorFlow Federated: Machine Learning on Decentralized Data. Available online at: https://www.tensorflow.org/federated.

[29] PYGRID: A Peer-to-Peer Platform for Private Data Science and Federated Learning. 2020. Available online at: https://blog.openmined.org/what-is-pygrid-demo/.

[30] Q. Li, et al. “A survey on federated learning systems: vision, hype and reality for data privacy and protection” IEEE Transactions on Knowledge and Data Engineering (2021).

[31] Y. Liu, et al. “FATE: An industrial grade platform for collaborative learning with data protection”. Journal of Machine Learning Research 22.226 (2021): 1-6.

[32] NVIDIA Clara Documentation. Available online at: https://docs.nvidia.com/clara/ (accessed 25. February 2022).

[33] H. Ludwig, et al., "IBM federated learning: an enterprise framework white paper v0. 1". arXiv preprint arXiv:2007.10987 (2020).
[34] N. Samuel, T. Diskin, and A. Wiesel, "Learning to Detect", IEEE Transactions on Signal Processing, vol. 67, no. 10, pp. 2554-2564, May 2019.
[35] M. Khani, M. Alizadeh, J. Hoydis, and P. Fleming, "Adaptive Neural Signal Detection for Massive MIMO", IEEE Transactions on Wireless Communications, vol. 19, no. 8, pp. 5635-5648, May 2020.
[36] M. Hummert, D. Wübben, and A. Dekorsy, "DeEQ: Deep Equalization for Large MIMO Systems", 24th International ITG Workshop on Smart Antennas (WSA), Hamburg, Germany, Feb. 2020.
[37] E. Beck, C. Bockelmann, and A. Dekorsy, "CMDNet: Learning a Probabilistic Relaxation of Discrete Variables for Soft Detection with Low Complexity", IEEE Transactions on Communications, vol. 69, no. 12, pp. 8214-8227, Dec. 2021.
[38] T. Gruber, S. Cammerer, J. Hoydis, and S. ten Brink, "On deep learning-based channel decoding", 51st Annual Conference on Information Sciences and Systems (CISS), Baltimore, MD, USA, March 2017.
[39] E. Nachmani, et al., "Deep Learning Methods for Improved Decoding of Linear Codes", IEEE Journal of Selected Topics in Signal Processing, vol. 12, no. 1, pp. 119-131, Feb. 2018
[40] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes", IEEE International Conference on Communications (ICC), Geneva, Switzerland, May 1993
[41] Y. Jiang, et al., "DEEPTURBBO. Deep Turbo Decoder", IEEE 20th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), Cannes, France, July 2019.
[42] Y. Jiang, H. Kim, H. Asnani, S. Kannan, S. Oh, and P. Viswanath "Turbo autoencoder: Deep learning based channel codes for point-to-point communication channels", 33rd Conference on Neural Information Processing Systems (NeurIPS), Vancouver, Canada, 2019.
[43] J. Clausius, et al.,, "Serial vs. Parallel Turbo-Autoencoders and Accelerated Training for Learned Channel Codes", 11th International Symposium on Topics in Coding (ISTC), Montreal, QC, Canada, Aug. 2021.
[44] M. Hummert, et al., "Neural Network-based Forecasting of Decodability for Early ARQ", 17th International Symposium on Wireless Communication Systems (ISWCS), Berlin, Germany, Sept. 2021.

[45] J. K.-S. Lee and J. Thorpe, "Memory-Efficient Decoding of LDPC Codes", International Symposium on Information Theory (ISIT), Adelaide, SA, Australia, Sept. 2005.

[46] C. Kestel, M. Herrmann, and N. When, "When Channel Coding Hits the Implementation Wall", IEEE 10th International Symposium on Turbo Codes & Iterative Information Processing (ISTC), Hong Kong, China, Dec. 2018.

[47] F. J. C. Romero and B. M. Kurkoski, "LDPC Decoding Mappings that Maximize Mutual Information," IEEE Journal on Selected Areas in Communications, vol. 34, no. 9, pp. 2391–2401, Aug. 2016.

[48] N. Tishby, et al., "The Information Bottleneck Method", 37^{th} Allerton Conference on Communication, Control, and Computing, Monticello, IL, USA, 1999.

[49] J. Lewandowsky and G. Bauch, "Information-Optimum LDPC Decoders Based on the Information Bottleneck Method", IEEE Access, vol. 6, pp. 4054-4071, 2018.

[50] M. Meidlinger, G. Matz, and A. Burg, "Design and Decoding of Irregular LDPC Codes Based on Discrete Message Passing", IEEE Transactions on Communications, vol. 68, no. 3, pp. 1329-1343, March 2020.

[51] R. Ghanaatian, A. Balatsoukas-Stimming, T. C. Muller, M. Meidlinger, G. Matz, A. Teman, and A. Burg, "A 588-Gb/s LDPC Decoder Based on Finite-Alphabet Message Passing", IEEE Transactions on Very Large Scale Integration Systems, vol. 26, no. 2, pp. 329-340, Feb. 2018.

[52] T. Monsees, et al., "Information Optimized Finite-Alphabet Message Passing Decoders using only Integer Operations", 23rd IEEE International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), Oulu, Finland, July 2022.

[53] B. Raaf et al., "Key technology advancements driving mobile communications from generation to generation", in Intel Technology Journal 18 (1), 2014.

[54] D. Wübben, et al., "Benefits and Impact of Cloud Computing on 5G Signal Processing", IEEE Signal Processing Magazine, vol. 31, no. 6, pp. 35-44, Nov. 2014.

[55] P. Rost, et al., "Benefits and Challenges of Virtualization in 5G Radio Access Networks", IEEE Communications Magazine, vol. 53, no. 12, pp. 75-82, Dec. 2015.

[56] ITU-T, "5G wireless fronthaul requirements in a passive optical network context", ITU-T Series G Suppl. 66, September 2020.

[57] J. Bartelt, et al., "Fronthaul and Backhaul Requirements of Flexibly Centralized Radio Access Networks", IEEE Wireless Communications Magazine, vol. 22, no. 5, pp. 105-111, Oct. 2015.

[58] J. Demel, et al., "Cloud-RAN Fronthaul Rate Reduction via IBM-based Quantization for Multicarrier Systems", 24th International ITG Workshop on Smart Antennas (WSA), Hamburg, Germany, 2020.

[59] S. Hassanpour, et al., "Generalized Distributed Information Bottleneck for Fronthaul Rate Reduction at the Cloud-RANs Uplink", IEEE Global Communications Conference (Globecom), Taipei, Taiwan, Dec. 2020.

[60] S. Hassanpour, et al., "Forward-Aware Information Bottleneck-Based Vector Quantization for Noisy Channels", IEEE Transactions on Communications, vol. 68, no. 12, pp. 7911-7926, Dec. 2020.

[61] S. Hassanpour, D. Wübben, and A. Dekorsy, "Forward-Aware Information Bottleneck-Based Vector Quantization: Multiterminal Extensions for Parallel and Successive Retrieval", IEEE Transactions on Communications, vol. 69, no. 10, pp. 6633-6646, Oct. 2021.

[62] T. Monsees, et al., "Relative Entropy based Message Combining for Exploiting Diversity in Information Optimized Processing", 25^{th} International ITG Workshop on Smart Antennas (WSA), France, Nov. 2021.

[63] K. R. Tuli, A. K. Kohli, S. G. Bharadwaj "Rethinking customer solutions: from product bundles to relational processes", Journal of Marketing, Vol. 71 No. 3, pp. 1-17 (2007).

[64] N. V. Wunderlich, et Al. "Futurizing' smart service: implications for service researchers and managers", Journal of Services Marketing, Vol. 29 Nos 6-7, pp. 442-447 (2015).

[65] N. V. Wunderlich, et al., "High tech and high touch: a framework for understanding user attitudes and behaviors related to smart interactive services", Journal of Service Research, Vol. 16 No. 1, pp. 3-20 (2013).

[66] M. W. van Alstyne, et al., "Pipelines, platforms, and the new rules of strategy", Harvard Business Review, Vol. 94 No. 4, pp. 54-62 (2016).

[67] T. R. Eisenmann, G. G. Parker, M. W. van Alstyne "Strategies for two-sided markets", Harvard Business Review, Vol. 84 No. 10, pp. 92-101 (2006).

[68] S. Park "Quantitative analysis of network externalities in competing technologies: the VCR case", The Review of Economics and Statistics, Vol. 86 No. 4, pp. 937-945 (2004).

[69] A. Cangelosi, et al., "Embodied Intelligence", Springer Handbook of Computational Intelligence, pp. 697-714, 2015. Available online

at: https://www.researchgate.net/publication/283812826_Embodied_Intelligence
[70] H. Klocke "Intelligente Agenten", Fachhochschule Köln, Campus Gummersbach (2011-2012), available online at http://www.gm.fh-koeln.de/~{}hk/lehre/ki/ws1112/bilder/ki_ws1112_welcome.html.
[71] M. H. C. Garcia et al., "A Tutorial on 5G NR V2X Communications", in IEEE Communications Surveys & Tutorials, vol. 23, no. 3, pp. 1972-2026, thirdquarter 2021, doi: 10.1109/COMST.2021.3057017.
[72] Federated Learning: Collaborative Machine Learning without Centralized Training Data, available online at https://ai.googleblog.com/2017/04/federated-learning-collaborative.html
[73] 3GPP TR 22.874, "5G System (5GS); Study on traffic characteristics and performance requirements for AI/ML model transfer".
[74] M. Kountouris, N. Pappas "Semantics-Empowered Communication for Networked Intelligent Systems", 2020. Available online at https://arxiv.org/abs/2007.11579.
[75] H. Seo, J. Park, B. Mehdi, M. Debbah "Semantics-Native Communication with Contextual Reasoning", 2021, available online at https://arxiv.org/abs/2108.05681.
[76] A. Das, et al. "TarMAC: Targeted Multi-Agent Communication", 2018, available online at https://arxiv.org/abs/1810.11187.
[77] Lazaridou, Angeliki & Baroni, Marco. (2020). Emergent Multi-Agent Communication in the Deep Learning Era, available at https://arxiv.org/abs/2006.02419
[78] M. A. Wirtz "Dorsch - Lexikon der Psychologie", Available online at https://dorsch.hogrefe.com/stichwort/oekologische-psychologie.
[79] L. Shapiro, S. Spaulding "Embodied Cognition", Stanford Encyclopaedia of Philosophy (2021), available online at: https://plato.stanford.edu/entries/embodied-cognition/.
[80] T. S. Rappaport et al., "Wireless Communications and Applications Above 100 GHz: Opportunities and Challenges for 6G and Beyond", in IEEE Access, vol. 7, pp. 78729-78757, 2019, doi: 10.1109/ACCESS.2019.2921522.
[81] N. Rajatheva, et al., "White paper on broadband connectivity in 6G", arXiv preprint arXiv:2004.14247, 2020.
[82] A. Bourdoux, et al., "6G white paper on localization and sensing". arXiv preprint arXiv:2006.01779, 2020.
[83] V. Frascolla et al., "Challenges and opportunities for millimeter-wave mobile access standardisation", 2014 IEEE Globecom Workshops,

Austin, TX, 2014, pp. 553-558. doi: 10.1109/GLOCOMW.2014.7063490.

[84] V. Frascolla et al., "MmWave use cases and prototyping: A way towards 5G standardization", 2015 European Conference on Networks and Communications (EuCNC), Paris, 2015, pp. 128-132. doi: 10.1109/EuCNC.2015.7194054.

[85] M. Shariat, et al., "Enabling wireless backhauling for next generation mmWave networks", 2015 European Conference on Networks and Communications (EuCNC), Paris, 2015, pp. 164-168, doi: 10.1109/EuCNC.2015.7194061.