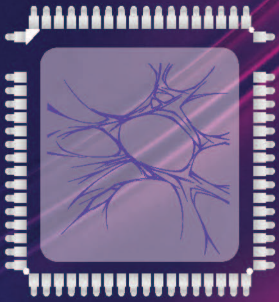
River Publishers Series in Communications

Intelligent Edge-Embedded Technologies for Digitising Industry



Editors:
Ovidiu Vermesan
Mario Diaz Nava

River Publishers

Intelligent Edge-Embedded Technologies for Digitising Industry

RIVER PUBLISHERS SERIES IN COMMUNICATIONS AND NETWORKING

The “River Publishers Series in Communications and Networking” is a series of comprehensive academic and professional books which focus on communication and network systems. Topics range from the theory and use of systems involving all terminals, computers, and information processors to wired and wireless networks and network layouts, protocols, architectures, and implementations. Also covered are developments stemming from new market demands in systems, products, and technologies such as personal communications services, multimedia systems, enterprise networks, and optical communications.

The series includes research monographs, edited volumes, handbooks and textbooks, providing professionals, researchers, educators, and advanced students in the field with an invaluable insight into the latest research and developments.

Topics included in this series include:

- Communication theory
- Multimedia systems
- Network architecture
- Optical communications
- Personal communication services
- Telecoms networks
- Wi-Fi network protocols

For a list of other books in this series, visit www.riverpublishers.com

Intelligent Edge-Embedded Technologies for Digitising Industry

Editors

Ovidiu Vermesan

SINTEF, Norway

Mario Diaz Nava

STMicroelectronics, France

River Publishers

Published, sold and distributed by:
River Publishers
Alsbjergvej 10
9260 Gistrup
Denmark

www.riverpublishers.com

ISBN: 978-87-7022-611-0 (Hardback)
978-87-7022-610-3 (Ebook)

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made.

Printed on acid-free paper.

Dedication

"Wise thinkers prevail everywhere."

- Sophocles

"Intelligence is what you use when you don't know what to do."

- Jean Piaget

"The intelligence is proved not by ease of learning, but by understanding what we learn."

- Joseph Whitney

"A computer would deserve to be called intelligent if it could deceive a human into believing that it was human."

- Alan Turing

Acknowledgement

The editors would like to thank all the contributors for their support in the planning and preparation of this book. The recommendations and opinions expressed in the book are those of the editors, authors, and contributors and do not necessarily represent those of any organizations, employers, or companies.

Ovidiu Vermesan
Mario Diaz Nava

Contents

Preface

Intelligent Edge-Embedded Technologies for Digitising Industry

Industrial intelligent edge systems are designed with more computing power and sensors to enable analytics, AI inferencing, and natural user interfaces. These new capabilities enhance their behaviour and provide new functionalities based on sensing, actuating, programming, and connectivity to dynamically interact and autonomously function.

Intelligent edge architectures are complementary to embedded systems, bringing scalable computing nearer to resource-constrained embedded systems and enabling these systems to leverage more complex, computing-intensive processes (including machine and deep learning), and local processing of real-time and historical data.

Intelligent edge devices are often resource-constrained by design. Such fixed-function systems are highly optimised for performance (speed, reliability, safety) and cost.

By making additional computing resources available to these systems, intelligent edge deployments enable diverse decision-making processes in the local industrial environment. These include system-level optimisations across devices, changes to the programming of specific devices, and other forms of control.

AI algorithms are processed locally, directly on the device, on the gateway, or on-premises servers near the edge devices. The algorithms utilise the data generated by the devices themselves. Industrial edge IIoT devices can make independent decisions in a matter of milliseconds without having to connect to the cloud.

As the computing and microcontroller architectures evolve, they support edge AI on embedded industrial systems and make the most of the limited computing resources there. The ARM Cortex cores, and AI accelerator's developments are pushing forward AI in resource-constrained environments. Several chip manufacturers are directly enabling machine learning-based AI on their microcontrollers. The increased hardware support for AI, including tools for edge AI, opens new opportunities for industrial edge AI implementations and deployments with new AI configurations that can operate in real-time and be integrated into the industrial manufacturing process.

This book provides a valuable resource for researchers working with intelligent edge-embedded technologies for digitising industry and industry professionals, machine and deep learning engineers, front-end developers, IIoT developers, and back-end developers looking to deploy intelligent solutions at the industrial edge.

List of Figures

List of Tables

List of Contributors

Ali, Rashid, *Fraunhofer IIS, Germany*

Arsalan, Muhammad, *Infineon, Germany*

Bahr, Roy, *SINTEF AS, Norway*

Bierzynski, Kay, *Infineon, Germany*

Borggreve, David, *Fraunhofer EMFT, Germany*

Bröring, Arne, *Siemens AG, Germany*

Brederlow, Ralf, *Technical University of Munich, Germany*

Calandra, Davide, *Politecnico di Torino, Italy*

Coppola, Marcello, *STMicroelectronics, France*

Döricht, Volkmar, *Siemens AG, Germany*

De Luca, Cristina, *Silicon Austria Labs GmbH, Austria*

Debacker, Peter, *imec, Belgium*

Debaillie, Björn, *imec, Belgium*

Dekorsy, Armin, *University of Bremen, Germany*

Diaz Nava, Mario, *STMicroelectronics, France*

Frascolla, Valerio, *Intel Deutschland GmbH, Germany*

Geiser, Florian, *Motius GmbH, Germany*

Höß, Alfred, *Ostbayerische Technische Hochschule Amberg-Weiden, Germany*

Han, Bin, *Technische Universität Kaiserslautern, Germany*

Hummert, Matthias, *University of Bremen, Germany*

John, Reiner, *AVL List GmbH, Austria*

Kämpfe, Thomas, *Fraunhofer IPMS*

Kaiser, Joachim, *Siemens AG, Germany*

Kundu, Bijoy, *Fraunhofer IIS, Germany*

Laleni, Nellie, *Fraunhofer IPMS*

Lamberti, Fabrizio, *Politecnico di Torino, Italy*

Maen, Mallah, *Fraunhofer IIS, Germany*

Mateu, Loreto, *Fraunhofer IIS, Germany*

Michailow, Nicola, *Siemens AG, Germany*

Monsees, Tobias, *University of Bremen, Germany*

Muir, Dylan, *SynSense, Switzerland*

Narduzzi, Simon, *CSEM, Switzerland*

Niedermeier, Christoph, *Siemens AG, Germany*

Ocket, Ilja, *imec, Belgium*

Pétrot, Frédéric, *University Grenoble Alpes, CNRS, Grenoble INP, TIMA, France*

Richerzhagen, Björn, *Siemens AG, Germany*

Schneider, Mathias, *Ostbayerische Technische Hochschule Amberg-Weiden, Germany*

Schotten, Hans, *Technische Universität Kaiserslautern, Germany*

Sifalakis, Manolis, *imec, The Netherlands*

Soliman, Taha, *Robert Bosch GmbH*

Traferro, Stefano, *imec, The Netherlands*

Urlini, Giulio, *STMicroelectronics, Italy*

Valentian, Alexandre, *CEA, France*

Van Leuken, Rene, *TU Delft, The Netherlands*

Vanselow, Fank, *Fraunhofer EMFT, Germany*

Vardar, Alptekin, *Fraunhofer IPMS*

Vermesan, Ovidiu, *SINTEF AS, Norway*

Vijayan, Preetha, *imec, The Netherlands*

Villnow, Michael, *Siemens AG, Germany*

Viseras, Alberto, *Motius GmbH, Germany*

Wübben, Dirk, *University of Bremen, Germany*

Wessel, Daniel, *Motius GmbH, Germany*

Wissel, Matthias, *Motius GmbH, Germany*

Yousefzadeh, Amirreza, *imec, The Netherlands*

Zhang, Lei, *Fraunhofer EMFT, Germany; Technical University of Munich, Germany*

List of Abbreviations

AP	Access point
ARQ	Automatic repeat request
ASIC	Application specific integrated circuit
CPU	Central processing unit
CU	Central unit
DL	Deep learning
DSO	Distribution system operator
DU	Distributed unit
DT	Digital twin
e2e	End to end
FEC	Forward error correction
FH	Fronthaul
FL	Federated learning
FPGA	Field-programmable gate array
GPU	Graphic processing units
HW	Hardware
IBM	Information bottleneck method
IIoT	Industrial internet of things
IoT	Internet of things
LUT	Look-up-table
MedTech	Medical technology
MIMO	Multi-input multiple-output
ML	Machine learning
NN	Neural network
QA	Quality assurance
RAN	Radio access network
RU	Radio unit
SDK	Software development kit
SotA	State of the art
SW	Software
TSO	Transmission system operator

1

Industrial AI Technologies for Next-Generation Autonomous Operations with Sustainable Performance

Ovidiu Vermesan[1], Frédéric Pétrot[2], Marcello Coppola[3], Mathias Schneider[4], and Alfred Höß[4]

[1]SINTEF AS, Norway
[2]University Grenoble Alpes, CNRS, Grenoble INP, TIMA, France
[3]STMicroelectronics, France
[4]Ostbayerische Technische Hochschule Amberg-Weiden, Germany

Abstract

This book lays down the technological foundation for and introduces key artificial intelligence (AI) concepts and technologies for the digitising industry. While this chapter does not exhaustively cover all types of AI, it comprehensively prioritises the features of AI-based industrial applications and designs and defines the reference terminology used in the other chapters of the book.

AI integrates several interrelated technologies to solve problems and perform tasks to achieve defined objectives; hence, AI can be approached from many viewpoints, such as mathematics and computer science, linguistics, psychology, neurology, and philosophy. The approach in this chapter is from a technological and industrial perspective, and concepts and functions are presented intuitively and visually, focusing on AI, as it is applied to embedded systems, with industrial automation, interactivity, and sustainability in mind.

This already reflects the next-generation deployment of AI into edge devices (called edge AI) and the emergence of different edge layers (i.e., micro-, deep- and meta-edge), which contrasts existing solutions that are currently deployed in the cloud. The edge processing continuum includes

sensing, processing and communication devices (micro-edge) close to the physical industrial assets under monitoring, gateways and intelligent controller processing devices (deep-edge) and on-premise multi-use computing devices (meta-edge).

Furthermore, instead of attempting to present a definition of AI that is common to all industries, the chapter relies on a framework of classifications and continuums along various dimensions, including the industrial intelligence spectrum, the intelligent capabilities spectrum, the edge-cloud continuum, the symbolic reasoning – pattern recognition continuum and, not the least, the problem-solving spectrum. The chapter introduces some of the main pillars of problem solving, such as expert systems, genetic and evolutionary computation, intelligent agents, machine learning (ML) and more.

This chapter, in particular, will detail ML approaches and neural networks. During the past decades, the trends and developments in AI have followed a recurring pattern, where the focus has moved back and forth between logic (symbolic reasoning) and pattern recognition (neural networks), driven by the varying abilities of technologies to acquire data, learn, derive new information and reason to reach decisions. In the last years, ML and neural network models have been the primary focus due to advances in hardware development and processing capabilities. Furthermore, embedded ML has been increasingly gaining popularity in industrial applications.

This chapter introduces several contributions. First, it gives a high-level overview of how AI works. Second, it shows how AI methods and techniques can be incorporated into an industrial design workflow. Finally, it provides a valuable intuitive understanding of how AI methods and techniques work when deployed in edge devices and how they operate in industrial settings.

Keywords: artificial intelligence, industrial AI, sustainable AI, intelligent embedded systems, symbolic AI, logic-based technologies, machine learning, AI-based problem-solving, AI technology stack, neural network architectures, embedded ML development.

1.1 Industrial AI

A recent report [1] predicted that the smart manufacturing market is expected to reach $446.24 billion by 2029, growing at a compound annual growth rate (CAGR) of 21.5% from 2022 to 2029. The smart manufacturing market is segmented into industrial Internet of Things (IIoT), cloud computing and storage, robotics and automation, industrial cybersecurity, additive manufacturing, augmented reality (AR), virtual reality (VR), digital twin, artificial intelligence (AI) and blockchain-based technology. In 2022, the IIoT segment is expected to account for the largest share of the smart manufacturing market. The large market share of this segment is attributed to factors such as the consistent declining cost of IIoT sensors, the significant rise in overall equipment effectiveness (OEE) through IIoT usage and increasing government initiatives to promote digital transformation. According to another market research report [2], the industrial AI market is expected to grow from $1,482.50 million in 2021 to 17,925.50 million by 2028 at a CAGR of 51.50% during the forecast period of 2022–2028.

Industrial AI refers to the application of AI in various industrial sectors and is considered a game changer in the manufacturing industry. The transition to Industry 5.0 is likely to drive the market's growth in the next few years. In manufacturing plants, the information obtained from various sensors, software and IIoT-driven systems may become too complex for humans to analyse. The use of AI is an efficient solution that can assist the manufacturing sector in transforming completely through ML and pattern recognition. The use of AI in manufacturing plants allows users to analyse and predict user behaviour, to perform predictive maintenance to prevent unwanted shutdowns, detect abnormalities in the production process and much more. AI also facilitates the use of real-time information, which could improve decision-making time, lead to better fabrication quality and yield, and boost organisational growth. The increasing volume of data gathered through various devices together with the widespread availability of high-speed communication networks and the upcoming implementation of wireless technologies will contribute to the increased use of AI in manufacturing in the future.

Although embedded in an increasing variety of products, processes and services in many industrial sectors, AI remains difficult to define. In scientific terms, AI is for example defined as "The designing and building of intelligent agents that receive precepts from the environment and take actions that affect that environment." [8]. AI and machine intelligence can also be defined as follows: "Artificial Intelligence is [...] the study of the computations that make it possible to perceive, reason, and act." [4]; "[Intelligence is] the capability of a system to adapt its behaviour to meet its goals in a range of environments." [5]; "Intelligence measures an agent's ability to achieve goals in a wide range of environments." General definition: "A very general and flexible capacity to succeed when faced with a wide range of problems and situations." [6];

"Intelligence is the computational part of the ability to achieve goals in the world." [7]. In more general terms AI refers to the ability for machines, systems, models, computers, to be able to mimic and improve intelligence in general, and human intelligence in particular.

Currently, neither the industry nor the scientific community has agreed on a particular definition. The presently available definitions are too vague, or too broad or too narrow. This is largely because of the growing variety and specific properties of AI technologies and partly because of the convergence of multiple technologies in the last years into AI, such as semiconductor technologies, cyber-physical systems (CPSs), internet of things (IoT), IIoT supervisory control and data acquisition (SCADA), programmable logic controllers (PLCs), 5G, distributed ledger technologies, edge computing, etc. The AI ecosystems are extending to related fields, such as edge computing, to address the challenges and requirements of various industrial sectors, and each field defines AI from its own perspective [11].

In this chapter, AI is approached from a computer science and information technology perspective, encompassing numerous technologies and frameworks, and focusing largely on embedded hardware/software systems that use searching algorithms, logic-based procedures or ML methods. ML represents a paradigm shift in computing - a change from explicitly modelling solutions to modelling systems that approach such solutions, which drives one to think in a new framework. ML - both software and hardware - is therefore addressed in several sections.

1.1.1 Challenges of Industrial AI versus Consumer AI

Although industry stakeholders have different perceptions of AI technologies and their industrial applications, industrial AI poses unique challenges that are absent from consumer AI or are present but of less importance or differ from the challenges related to the latter. Some of these challenges are described below.

Industrial training data are in short supply. AI-based models require large amounts of data, and their performance relies strongly on training data set availability. These data sets exhibit tremendous potential for optimising industrial processes in cases in which traditional approaches, such as stochastics and analytical or numerical models, can no longer be used. However, for many industrial sectors, it is not easy to create training data sets that are sufficiently large and cover common aspects that would allow them to be used by different industrial stakeholders to benchmark similar AI models.

Industrial training data are often noisy or inaccurate. While data coming from consumers are hard to misinterpret, this is not the case with industrial data, which is frequently captured from sensors and IoT/IIoT that produce noisy data sets. Sensor data can also be voluminous, and not all data is relevant. Data can also be inaccurate when generated by "digital twins" models that are not always created and maintained in tandem with the real system. Furthermore, the actual deployment of sensors close to production environments that are generally ungracious of the sensors (higher likelihood of sensor malfunction) and redundancy to alleviate this problem introduces additional challenges and costs. Nevertheless, despite the high volume of noisy, incomplete, or faulty data, industrial AI needs to be highly accurate.

Industrial AI runs mostly on the edge. Consumer data are processed on computers with seemingly infinite capacities, and current AI tools are optimised for cloud services and therefore do not always fulfil the stringent requirements of industrial applications, such as real-time processing, low latency, high reliability, safety, data privacy and guaranteed QoS. To be successfully implemented in industry, AI must be deployed on the edge to support distributed on-site data processing with state-of-the-art AI components, algorithms, techniques, and methods.

Industrial AI can be subject to compliance with industry standards and other regulations. While consumer AI can at most be subject to direct consumer scrutiny, industrial AI is subject to compliance requirements, including technical, legal and corporate requirements, as well as local and governmental regulations, which may impact operations, particularly when large budgets are at stake.

Industrial AI involves high costs. The development, implementation, deployment, repair, and maintenance of AI-based solutions necessitate vast investments. AI-based systems require frequent upgrades to meet the needs of changing environments and to make machines more intelligent day by day. In severe breakdowns, the recovery of lost codes and the restoration of AI-based systems may require considerable amounts of time and costs. Maintenance of the sensor part of the AI solution also contributes to overall costs.

Industrial AI must be explainable. Industrial AI applications must be able to explain and justify their predictions and decisions, especially when the consequences of wrong decisions can be disastrous.

Industrial AI systems are difficult to validate and test due to the costs and complexity involved. The complexity of AI tasks has increased steadily to address new paradigms for automating, conceptualising, designing, and implementing AI-based systems that include sensors, hardware, software, models, and algorithms. In many cases, industrial AI systems are trained and tested using simulations and virtual validations.

1.1.2 Sustainable AI

The search for improved accuracy on large-scale problems is driving the use of new AI techniques and increasingly deeper neural networks, thereby increasing energy consumption and climate-changing carbon emissions.

Advances in scientific computing have demonstrated the advantages of modelling and simulation across industrial and scientific domains. However, energy consumption is a feasibility constraint for computational modelling, and AI must reduce the energy computation costs associated with high-performance computing in front of trends such as declining Moore's Law.

The evolution and expansion of AI-based technologies require moving towards sustainable AI and using AI for sustainability in various industrial sectors. To build and strengthen sustainable AI technologies and applications, new solutions need to be developed to move AI processing from the cloud to the edge, optimise and reduce the need for data sets and amount of data for training and learning and address the analytics close to the data sources.

Sustainable AI (or AI sustainability) requires stimulating change in the entire lifecycle of AI technologies and applications (e.g., AI function generation, AI technology stack, HW/SW platforms, training/learning, re-tuning, re-training/learning implementation, governance) towards more efficient ecological integrity and economic efficiency. Sustainable AI technologies are compatible with sustainable environmental resources for current and future generations, and new digital economic models are aligned with industrial and societal values.

Research and developments in industrial edge AI incorporate two essential elements: sustainable edge AI (e.g., edge AI technologies development for optimised resource processing consumption, resource consumption for AI models, reduction of carbon emissions, computing power, etc.) and edge AI for sustainability (e.g., the use of edge AI to address sustainability goals in different applications and industrial sectors). These elements can be viewed from the perspective of the different pillars of sustainability (e.g., social, economic, and environmental).

Sustainable industrial edge AI focuses on developing AI HW/SW/algorithms and resource-efficient edge AI technologies to reduce carbon emissions and computing power consumption of AI models.

Industrial edge AI for sustainability and sustainable computing leverages intelligent processing technologies to address environmental and climate problems and ameliorate the accelerating trend towards high-performance computing in modelling and simulation.

Leveraging hardware modules and platform characteristics to generate compact and accurate models that require less computational resources is essential for sustainable edge AI. Combining different techniques, including knowledge distillation, AI HW/SW co-optimisation for power efficiency and energy-aware model compression, can result in models with negligible loss of accuracy.

Sustainable edge AI implies less data for model training to achieve high-accuracy model performance, thereby reducing the expensive data collection and annotations, accelerating model training when faced with a new problem and reducing the resource-intensive process of training a new model from scratch.

The development of semi-supervised methods [16] by incorporating external knowledge, active learning, transfer learning and short learning approaches, such as meta-learning and unsupervised representation learning, are elements of the AI technology stack that supports sustainable edge AI developments. These methods facilitate domain adaptation across problems, including natural language processing and predictive maintenance in different industrial sectors.

Sustainable edge AI requires enabling more accurate modelling techniques, reducing computing costs by reducing time-to-solution, decreasing the need for high-resolution models where possible and leveraging resource/data-efficient AI developments to ensure that the application of AI is not energy/resource-consuming.

Sustainable edge AI requirements advance the development of new embedded hardware modules, platforms, and accelerator architectures (e.g., system on module (SoM), system on chip (SoC), a system in package (SiP), neuromorphic, hybrid, tensor-based, etc.).

1.2 Capabilities Spectrum of Industrial AI

This chapter relies on a framework of classifications and continuums along various dimensions, including the industrial intelligence spectrum, the intelligent capabilities spectrum, the edge granularity, the edge continuum, the symbolic reasoning-pattern recognition continuum and not the least, the problem-solving spectrum.

The foundation supports the ongoing projects and stakeholders across the industrial sectors with a common methodology and roadmap. Meanwhile, it prioritizes the right features for AI-based applications and designing them in the right way in different use cases across various industrial sectors using synergies among different solutions, methods, or techniques.

The foundation assists in choosing state-of-the-art AI technologies and having a clear overview over the existing state of play in the field for optimal selection and trade-off of these technologies, methods, and techniques for use cases in different industrial sectors.

AI empowers computers to mimic human intelligence, such as decision-making, text processing and visual perception. In this context, AI is a broad field, encompassed by multiple contributing branches, such as ML, robotics and computer vision.

AI can be understood in the context of the tasks that we expect an intelligent machine, IoT/IIoT device to be capable of performing.

An intelligent machine, IoT/IIoT device is any system whose behaviour could be interpreted as reflecting human intelligence, which may be demonstrated in basic capabilities, such as perceiving, comprehending, acting, and learning. For example, the three-dimensional classification scheme for evaluating an AI-based systems in [37] differentiates four capabilities: perception, understanding, action and communication.

In the context of the European projects contributing to this book, four capabilities are differentiated as shown in Figure 1.1, but the list is extended with more capabilities, which are elaborated below from the perspective of their application to industry.

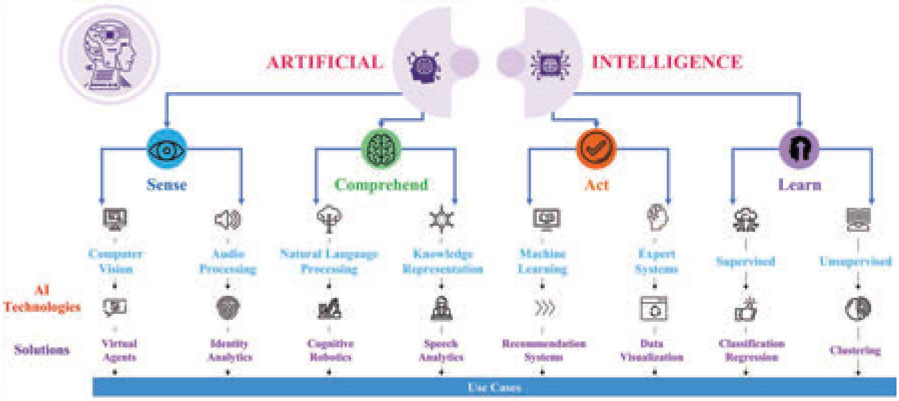


Figure 1.1 AI systems capabilities.

The **perceiving** or **sensing** capability allows industrial machines, IoT/IIoT devices to scan their environment using various sense devices, and to collect and process data streams (images, sounds, speech, text, and other data) from diverse sources, such as radar, light detection and ranging (LiDAR), cameras, ultrasound sensors, etc.

The processing is often complicated, as it involves great numbers of distinct appearances over multiple occasions, varying by view and angle, as well as scenes suggesting objects that may be hidden. Mechanisms, such as data, information, and sensor fusion are employed to assimilate various sources of information, often imperfect and uncertain, and deal with multiple dimensions of remote sensing (spatial, temporal, spectral, and radiometric resolution).

The **comprehending** capability enables industrial machines, IoT/IIoT devices to recognise patterns and context in the information it collects, just as humans interpret data/information by understanding patterns and context in their perceptions of the environment, but it is important to note that comprehending does not have the same meaning for machines as for humans. Machines do not actually "comprehend" the world around them; rather, they are trained to "learn" how to recognise patterns.

An industrial machine's and IoT/IIoT device's **learning** capability enables it to continually improve its performance by learning from the success or failure of its actions. Like humans, machines learn in various ways, for example, by trial and error. A machine tries various solutions to a problem until it achieves the desired results. It records all the steps actions that produced those results in its memory for use the next time it is given the same problem.

The **reasoning** capability enables industrial machines and IoT/IIoT devices to draw relevant inferences from the situation at hand. Reasoning has become an essential component of AI only in the past decades before which the ability was limited to humans.

Logic employs two broad methods of reasoning: the deductive and inductive approaches. Deductive reasoning works from the "top down", moving from a theory to its confirmation (or rejection) by collecting observations to address the hypotheses and narrowing down the possibilities.

By contrast, inductive reasoning works from the "bottom up", moving from specific observations to broader generalisations and theories by detecting patterns and then formulating testable hypotheses.

The **problem-solving** capability enables industrial machines and IoT/IIoT devices to move from a problem's initial state to the final goal state through a stepwise gradual reduction of the difference between the current state and any intermediate goal state.

This involves using several techniques, such as algorithms and heuristics, to solve a problem. The ability to solve problems, a highly prized skill in both humans and machines, involves two distinct, possibly conflicting processes: creativity and decision making. The former, creativity, generates options and possible solutions, and then the latter, decision making, selects the optimal one.

The **acting** capability enables industrial machines and IoT/IIoT devices to act (inspired by their perception or comprehension) in the physical or digital environment. It is implicitly assumed that machines will act rationally and determine the best and safest course of action for achieving their goals.

The **interacting** capability enables industrial machines and IoT/IIoT devices to connect to the environment and to everything and collaborate with humans, other machines, and infrastructure (physical and digital, edge/cloud, etc).

This emerges primarily when industrial machines have to interact with people, which assumes the ability to understand language. For example, when an AI system explains how it came to its decision, it must adopt the normal conventions of human interaction to make itself understood.

The **locating** capability enables industrial machines and IoT/IIoT devices to determine (relative) positions very precisely and accurately on network, dynamic maps, GPS, GNSS, etc., that help in identifying the context of actions. These capabilities could also apply to locating the state within a state machine or for temporal locating (e.g., clock or relative clocks between devices used for synchronisation) that can improve the initial state for action.

1.3 The Industrial AI Spectrum

In the previous section, a classification of AI was given in the context of the tasks that we expect an intelligent industrial machine to be capable of performing.

Additionally, AI can be understood from the perspective of the (theoretical) ability of an intelligent machine situated on a continuum, from specific to general intelligence or from basic to super intelligence. Some forms of AI within this continuum can be distinguished by names, such as Narrow AI, General AI, Weak AI, Strong AI [35].

These various forms of AI differ primarily in their range of abilities/capabilities and the level of training required to implement them. In the following, they are described in contrasting pairs from the perspective of their relevance to industrial applications.

1.3.1 Narrow AI vs. General AI

General AI defines an AI system that parallels human intelligence. As such, it is considered an ultimate vision of AI that can handle a wide variety of cognitive tasks across multiple domains.

General AI is the basis for future human-like autonomous systems and robots, which will implement hundreds of systems working in parallel while communicating with each other in a manner that mimics human reasoning. While the development of technology pushes the abilities of AI ever closer to General AI (e.g., consciousness, exhibit common sense, ability to reason, solve a puzzle, identify needs and emotions, adapt to conditions as the context is changing, use knowledge and experience to plan, etc.), most AI surrounding us today is Narrow AI.

Broadly speaking, Narrow AI can be thought of as anything that is not General AI. Narrow AI defines an AI system capable of performing a particular task that any human would ordinarily perform. Narrow AI systems are designed to precisely execute a well-defined task. These systems are optimised to excel in controlled environments with a limited set of parameters, demonstrating capabilities that match, or even surpass, those of a human. However, their capabilities are narrow (e.g., e-commerce suggestions based on user search patterns, weather prediction, predictive maintenance, etc.), and they cannot do anything that is not explicitly stated in their programming.

A comparison of the features of Narrow AI and General AI is illustrated in Figure 1.2. The primary difference between Narrow and General AI comes down to adaptability. For AI to be generally intelligent, it must be able to adapt rapidly to changing surroundings in the same way that humans do. In practice, this would mean to pass the Turing test repeatably and consistently. Turing defined intelligent behaviour as the ability to achieve human-level performance in all cognitive tasks sufficient to fool an interrogator.

1.3.2 Weak AI vs. Strong AI

The terms Weak AI and Strong AI are sometimes used in place for Narrow AI and General AI, respectively. Weak and Strong AI were coined in [9] to differentiate the performance levels in different kinds of AI machines.

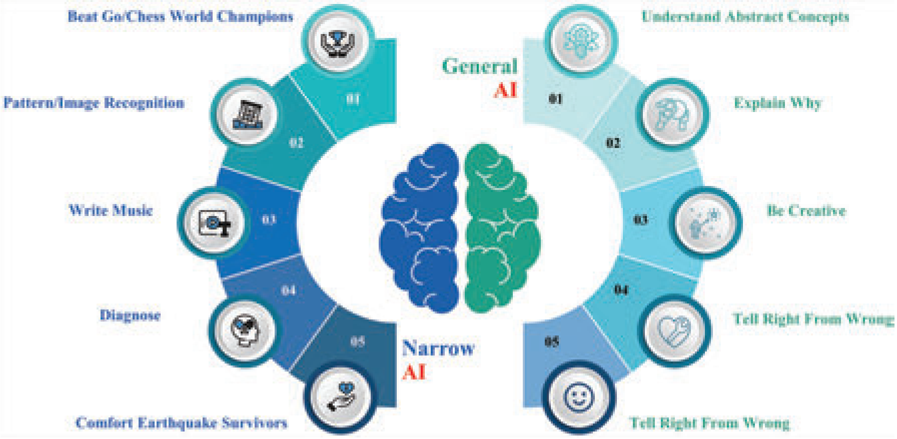


Figure 1.2 Narrow AI vs General AI.

Strong AI systems (e.g., advanced robotics, intelligent robotic things, etc.) behave intelligently, think as humans do, and have a conscious, subjective mind; they know who the AI systems are, what they are doing, and why. Strong AI systems are represented by an AI-based application with a larger scope, using high-level clustering and association to process data, information, and knowledge.

In contrast, Weak AI defines the simulated thinking of the brain processes with the help of a computer. It behaves intelligently (e.g., chatbots, Siri, Alexa, etc.) but does not exhibit any kind of consciousness about what it is doing.

Weak AI systems are represented by Narrow AI-based applications with a limited scope that are optimised by using supervised and unsupervised learning to process data collected from different sources (e.g., real-time or from databases, etc.).

Although Weak AI systems never attain the breadth of a General AI, most Narrow AI systems are very powerful and focused. It is therefore important to not conflate Narrow AI, which deals with specific tasks, with weak AI.

1.3.3 Basic AI vs. Super AI

The term Super AI refers to the combination of General AI and Strong AI at the point at which it surpasses the intelligence and ability of the human brain. This is made possible primarily due to the amount of memory and instantaneous access to data, which far exceeds human limits. In addition, this AI will improve self-capabilities to feel things and emotions.

Nevertheless, since Strong AI is still theoretical, the realization of Super AI lays far in the future, relying strongly on technological advancements in hardware (quantum computing), software, and other fields (biomimicry).

Basic AI, in contrast, can be considered for any AI that is under the threshold of Super AI. It is an all-encompassing term that denotes the simplest tasks and technologies used and is mentioned here merely as a foil to Super AI.

1.3.4 Red AI vs. Green AI

The term Green AI [12] defines AI research that yields novel technological results while considering the financial cost of developing, training, and operating, as well as encouraging a positive impact both on the environment and inclusiveness.

Green AI includes the optimisation of the use of data/information, the processing across the edge-cloud continuum, the transfer and exchange of data/information, and storage.

The term Red AI defines AI research that seeks to achieve progress regardless of the huge computational power required and environmentally unfriendly impact involved.

While Red AI research has made valuable scientific contributions to the field, making AI both greener and more inclusive will lead to wider acceptance of AI in industry.

Nevertheless, ensuring a smooth transition from Red AI to Green AI is not straightforward. For instance, the type of energy sources used for powering the data centres, edge computing facilities, or the intelligent devices at the edge is part of the efficiency equation associated with training/learning/reasoning algorithms. Even if powered by renewable sources, massive power consumption for stronger results from the algorithms may not be considered an improvement step towards Green AI.

To conclude the discussion on the industrial AI spectrum, the degree to which more General, Strong, Super and Green AI can be achieved will largely depend on the abilities of the particular AI system to continuously learn how to solve problems from multiple application domains without requiring extensive retraining for each, to learn in a self-supervised manner, and to adapt the knowledge and skills acquired to new situations with minimal training.

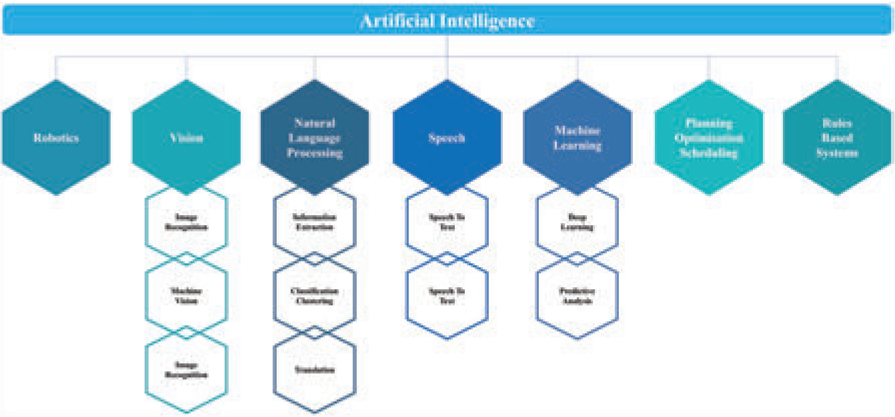


Figure 1.3 AI problem solving domains.

1.4 AI Problem Solving Domains

Problem-solving is a method used to reach a desired goal or find a solution to a problem. In the context of computer science, problem-solving refers to various techniques, such as forming efficient algorithms and heuristics, to find desirable solutions. A single problem can have many different solutions, and these can be achieved by different methods. Also, some problems have unique solutions, depending on the nature of the given problem and the domain.

AI has always been beneficial for solving complex problems and challenges that cannot be solved by other means. This section presents some of the major AI problem solving domains that are most used in industrial problem-solving and/or that have great potential for sustainable developments. The various branches of AI and AI problem-solving domains are illustrated in Figure 1.3. For a more complete overview of problem-solving techniques, we refer the interested reader to [4][8].

1.4.1 Expert Systems

Expert systems (ES) are computer programs designed to act as experts in a particular domain or area of expertise. In other words, they are designed to model human expertise in that specific knowledge area. Problem-solving relies on organising considerable amounts of knowledge and then systematically searching through them when selecting the path to go with each decision, ultimately leading to a solution. A typical architecture is shown in Figure 1.4.

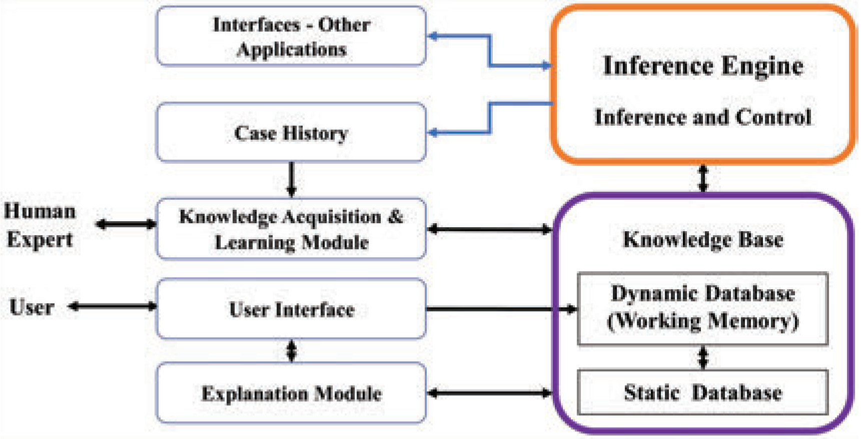


Figure 1.4 Typical expert system architecture.

There are two basic components in an expert system: a knowledge database and an inference engine. The knowledge base mainly consists of facts about that domain (declarative knowledge) and rules for applications to those facts (procedural knowledge). The most common representation of human expert knowledge is in the form of rules, for example, an ‘if A, then B’ structure.

The inference engine processes the input information (for example, that A is true) and draws the deductions based on the rules (for example, B). It consists of algorithms, which, via step-by-step inferences, draw deductions based on the knowledge rules. Depending on the application, ES may also have a user interface to interact with users.

In the absence of generalised knowledge-based systems, the industry embraced the idea of practical ES for specific tasks, and there are many successful applications of ES in medicine, agriculture, and other areas, where ES assist or even replace human experts with specialised knowledge. ES remain important tools for decision support or decision-making; nevertheless, they have evolved in both the technological and business directions. ES can now be embedded into applications and can be designed to handle uncertainty. Furthermore, new knowledge representation and reasoning tools have been developed for ES: MYCIN [13] for disease diagnosis, DENDRAL [30] for chemical analysis to predict molecular structure, R1 for configuring orders for new computer systems [14], Fuzzy Logic UAV (Unmanned Aerial Vehicle) Motion Planning [31], etc. There are other application areas such as environment, manufacturing, diagnostic tools for vehicles, and machinery.

ES remain a feasible solution when there is a lot of human expert knowledge and experience that can be modelled but not enough data to build other problem-solving systems. ES can also be the preferred solution because of their unique capability, namely explainability, which other systems lack in spite of advanced problem-solving capabilities. This could be an obstacle in some application areas, such as autonomous vehicles, where unexpected decisions need to be understood.

On the downside, knowledge bases take time to acquire and represent on computers, and if some knowledge is missing or incomplete, a less reliable result will be produced. Hence, verification and validation methods and techniques aimed at ensuring quality are fundamentally important.

Initially, ES were built around rules established by humans, but gradually the rules are being set by computers, which can interpret and extrapolate from large volumes of data. In this respect, the AI learning process can be implemented using top-down approaches (e.g., expert systems) or bottom-up approaches (e.g., machine learning).

ES and its technology have been one of the most important and widely used parts of AI and goes back to the beginning of AI, so they have been used in business for decades. This is an area that will continue to be important in the future, either independently or in combination with other major branches of AI.

As a concluding remark, there is a lot to learn from the earlier generation of AI in our pursuit of the development of explainable and verifiable AI.

1.4.2 Machine Vision

Machine vision (MV) is a branch of AI that enables machines to imitate the human visual system and perform various tasks, such as image classification and segmentation, object detection and recognition, and object tracking, using information collected from various sources including IIoT image sensors. MV enables intelligent vision devices to grasp their visual surroundings and to process, analyse and understand digital images. For example, in the case of autonomous vehicles, MV detects traffic signs, buildings, vehicles, pedestrians and other participants in the traffic.

Machine vision and computer vision (CV) are sometimes used interchangeably, but they are different. Both are used for image processing, so they both use similar components, such as cameras, IIoT image sensors to capture images and software to handle the data. However, CV uses systems with PC-based processors to analyse the imaging data it collects, so it has a lot of processing power and is commonly applied in the medical, financial and security industries. CV can be used alone, without needing to be part of a larger machine system.

MV, by contrast, is integrated into perception systems in industrial sectors, such as autonomous vehicles, manufacturing, food processing and semiconductors. An MV system uses algorithms to process and interpret an image, and it instructs other components in the system to act upon that data. MV systems are therefore designed to quickly analyse image data and make simple, automated decisions on different tasks, such as quality control, inspection, and guidance. The image could be obtained from a thermal or infrared sensor, IIoT image sensors, motion detectors or other sources.

Analysis of reams of images produced by sensors requires that the machines be able to see and understand images, and this is where AI comes into the picture because its methods and techniques permit the automatic extraction of information from images.

Machine vision is one of the areas that has greatly benefitted from the rapid advances in AI/ML, and implementation of MV's capabilities is now possible at all micro-, deep-, and meta-edge levels. Modern MV systems are usually built using different types of neural networks, including deep learning (DL). DL allows machines, robots and intelligent IIoT devices to recognise objects with close to human-like ability. At the lower levels, ML algorithms perform processing techniques on the image, extract features from the image and access and intertwine multiple views. At the higher levels, they perform more advanced tasks, such as image classification, and they make inferences about whether the object in the image belongs to a specific class of objects. The highest level is where DL is employed to build intelligent, scalable MV systems that can recognise/identify and react/respond to objects in images and videos.

From the multitude of neural network architectures, Convolutional Neural Networks (CNNs) have become increasingly powerful in large-scale image recognition by combining the feature extraction processes and classifying the extracted features in the same algorithm. When DL technology is deployed in IIoT devices, it relies on pretrained DL models, and transfer-learning techniques are employed to retrain an existing image classifier into a custom classifier by retraining a small image data set using minimal resources. An intuitive illustration of CNN-based MV is shown in Figure 1.5.

Nonetheless, some challenges arise when deploying MV on IIoT edge devices. Most deep NNs are too complex to be created and trained on most present-day microcontrollers, but if optimised in terms of memory,

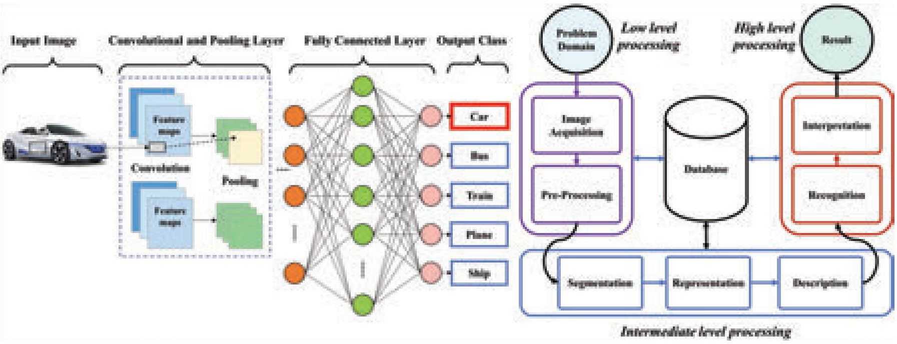


Figure 1.5 Typical CNN-based machine (left) and workflow (right).

processing, and power capabilities, they can run on them. The optimisation can be done either by rewriting the models in low-level languages or by quantising to improve the latency and the model size.

Real-time object detection [38] on edge devices with live video analytics using YOLO (You Only Look Once) are widely used for video surveillance and are important for mobile robots, including self-driving vehicles.

The machine vision system uses embedded edge AI for use-case applicability and autonomous optimisation in industrial manufacturing visual inspection and are extensively used in various industrial applications and sectors.

1.4.3 Robotics

Robotics is a branch of AI that deals with creating machines that can perform some actions like humans. AI capabilities enable robots to act intelligently in certain situations by solving problems in a limited sphere or even learning in controlled environments. Many industries are implementing robotics solutions to overcome critical issues related to production and execution by eliminating the potential for human error while reducing redundancy in manual labour.

In recent years, there has been consistent progress in intelligent robotics, driven by an increased availability of complex and intelligent sensor systems, powerful computing and communication capabilities, and software platforms. Progress in deep learning in particular is opening up new opportunities in industrial robotics – leveraging improvements in MV, robotic grippers that can pick up randomly placed objects and stack them, and other agile and dynamic robotics systems that operate at speeds essential for many industrial applications. Thus, implementing inference at the edge, without connecting to the Internet, enables robots to make decisions independently.

The greatest impact has been on automobile industry and the use of autonomous vehicles. The design of self-driving vehicles requires the integration of technologies such as sensor fusion, AI decision-making, vehicle-dynamics prediction, on-the-fly rerouting, and inter-vehicle communication to carry out tasks such as adaptive cruise control, to safely adjust speed, and lane-keeping assistance, to keep vehicles centred on the road. A schematic illustration of an end-to-end deep learning for self-driving vehicles is shown in Figure 1.6.

Training data contains single images sampled from video, paired with the corresponding steering command. Training with data from the human driver is insufficient. The network must likewise learn how to recover from any mistakes, or the vehicle can slowly drift off the road. In this case, the training data is augmented with additional images that show the vehicle in different shifts from the centre of the lane and rotations from the direction of the road. The images for two specific off-centre changes can be obtained from the right and left cameras [33].

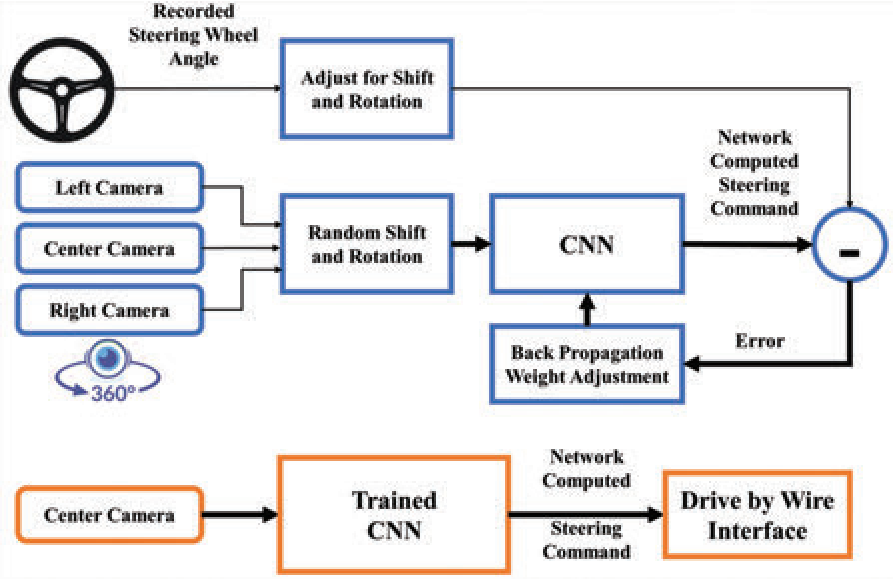

Recorded Steering Wheel Angle
Adjust for Shift and Rotation
Network Computed Steering Command
Left Camera
Center Camera
Right Camera
Random Shift and Rotation
CNN
-
Back Propagation Weight Adjustment
Error
360°
Center Camera
Trained CNN
Network Computed
Steering Command
Drive by Wire Interface

Figure 1.6 Self-driving vehicles: Training and inference (generate steering commands). Adapted from [33].

Images are fed into a CNN which computes a proposed steering command. The proposed command is compared to the desired command for that image, and the weights of the CNN are adjusted to bring the CNN output closer to the desired output. During inference, the trained model generates steering commands from the input video images.

1.4.4 Biomimicry

The term AI typically connotes emulating, mimicking, or replicating human intelligence in machines. However, AI also encompasses biological intelligence, including that of plants, animals, and other living organisms. Plants, for instance, do not possess brains, but they have senses. Hence, one of the many lessons to be learned from billions of years of evolution, natural engineering, and natural design is that embedding more processing power close to the sensors and actuators will improve intelligent functioning in IIoT technologies in different industrial applications.

There are many examples of AI problem-solving architectures and techniques that incorporate insight from nature into their solutions, e.g., ML, robotic vision, and path-planning. Biomimicry is an approach to problem-solving that produces innovative sustainable solutions by learning from and replicating the natural patterns observed in living systems and beings (e.g., plants, animals, humans) to create remarkably intelligent technologies and products. These patterns, which appear in nature with varying degrees of frequency, can be found not only in forms and shapes, but also in processes (chemical, physical, and behavioural) and ecosystems. Highly fundamental patterns observed across species that appear very frequently are known as Life's principles [25].

Life's Principles are lessons from nature and can serve as inspirational lessons for designers and developers and can be applied to various industrial applications. Being locally attuned and responsive is one of the six primary Life's Principles, and it refers to how an organism fits into and integrates with it surrounding environment. The use of feedback loops is a sub-principle or strategy for being locally attuned and responsive. Feedback loops are cyclical information flows that allow organisms to adequately modify their reactions to environmental stimuli and situations.

We can find hundreds of unique strategies and mechanisms in nature for sending and receiving signals in a feedback loop. The white clover is a good example of the information flow that occurs between a prey and the formation to which it belongs, finalised by feedback to the predator. To survive, this

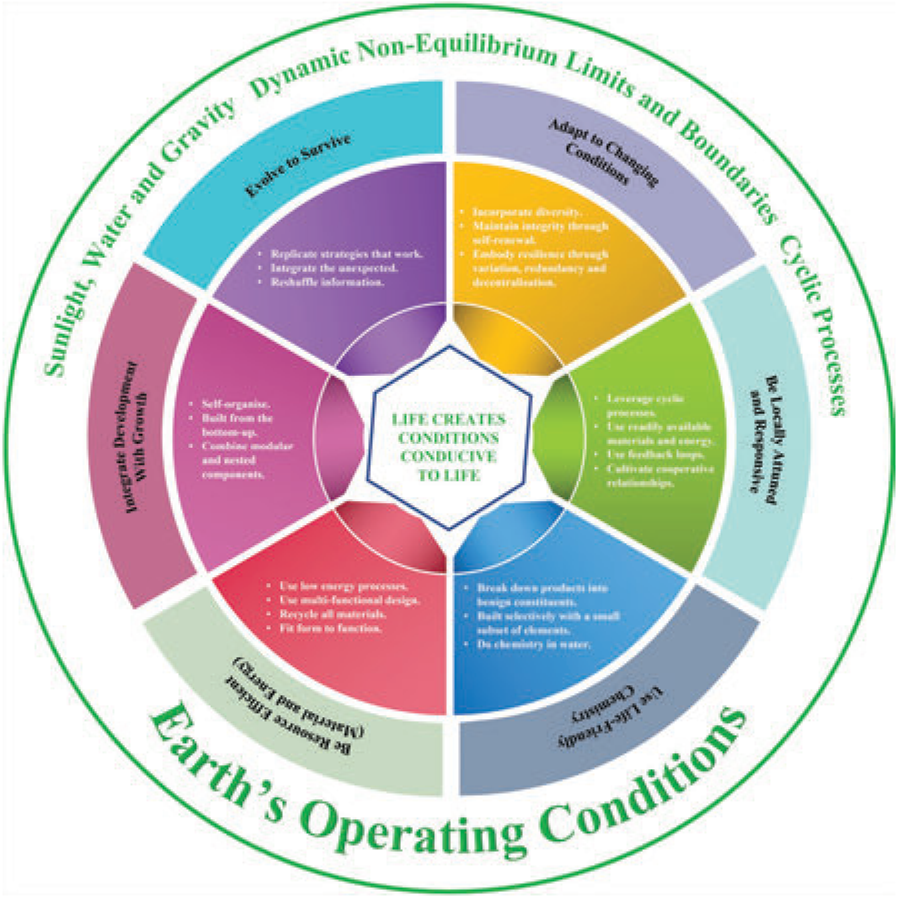

Dynamic Non-Equilibrium
Limits and Boundaries
Cyclic Processes
Sunlight, Water and Gravity
Earth's Operating Conditions
Evolve to Survive
Adapt to Changing Conditions
Be Locally Attuned and Responsive
Use Life-Friendly Chemistry
Be Resource Efficient (Material and Energy)
Integrate Development With Growth
• Replicate strategies that work.
• Integrate the unexpected.
• Reshuffle information.
• Incorporate diversity.
• Maintain integrity through self-renewal.
• Embody resilience through variation, redundancy and decentralization.
• Leverage cyclic processes.
• Use readily available materials and energy.
• Use feedback loops.
• Cultivate cooperative relationships.
• Break down products into benign constituents.
• Built selectively with a small subset of elements.
• Do chemistry in water.
• Use low energy processes.
• Use multi-functional design.
• Recycle all materials.
• Fit form to function.
• Self-organise.
• Built from the bottom-up.
• Combine modular and nested components.
LIFE CREATES CONDITIONS CONDUCIVE TO LIFE

Figure 1.7 Life's principles. Adopted from [25].

plant employs a chemical defence mechanism to ward off herbivores. When white clover leaves are damaged (chewed), two chemicals mix to form hydrogen cyanide, a bitter substance that makes the leaves less tasty.

For an AI system, feedback loops are essential because they enable intelligence. A feedback loop entails the assessing and leveraging of its output (predictions or recommendations) to retrain and improve the model over time. Feedback loops are used in ML and DL, especially in neural networks. A good example is the object recognition technology in self-driving vehicles and its ability to recognise traffic lights, road signs, pedestrians, automobiles, and various types of objects, with feedback loops improving its accuracy.

Through feedback from the output layer in a neural network model, the variations of weights in the hidden layer(s) are adjusted to fit the expected outputs. Positive feedback increases the change or output, while negative feedback decreases the change or output.

Returning to the white clover example, when a white clover leaf is attacked, this action triggers signals in every direction, making the other leaves harder to chew and upgrading the mechanical and chemical resistance of the entire formation. This is made possible by its network infrastructure composed of runners, commonly found in many plant species.

Runners are stems growing just at or below the soil surface. They form roots at the nodes, new plants grow from their buds, and they are part of a propagation strategy. Above ground, these plants most often appear to be distinct individuals, but underground, they are interconnected, such that when one member of the formation senses something, a signal is sent to every other member, facilitating a quick response to a predator.

Feedback loops also occur in ecosystems, with connections within the formation allowing for rapid information circulation, information processing, and reaction. To function like ecosystems, AI systems must be strongly interconnected and equipped with built-in IIoT technologies that continually capture stimuli from the environment. These stimuli are then converted into information that is circulated and processed rapidly, resulting in an almost instantaneous self-regulation and adaption to any change, along with feedback on the origin of the change.

AI systems functioning like ecosystems will foster collaborative infrastructure design and sustain innovation, enabling these systems to evolve and rapidly learn how to evolve. AI systems with *collaborating sensors* reminiscent of such collaborative infrastructure would behave almost organically.

1.4.5 Genetic and Evolutionary Algorithms

Genetic algorithms (GAs) represent a branch of AI searching for a range of potential solutions to find one which solves a particular problem. GAs save information about the paths traversed during the search, simulating an evolutionary process and in this way overcoming known issues such as inefficient searches, and convergence to local optimums rather than global optima.

The idea of GAs can be traced back to Alan Turing's paper from 1950 [26], where concepts derived from natural evolution are used to evolve AI machines. These include mutation, hereditary material, survival of the fittest, and keeping track of the different genetical combinations that have been tried and tested, to avoid trying the same ones again.

GAs are a subset of a much larger branch of computation known as evolutionary computation. The main concept behind GAs and evolutionary algorithms (EAs) is inspired by the natural selection principle in biological evolution [27], in which organisms evolve and adapt to thrive in the surrounding environmental conditions.

According to this principle, new candidates can be produced from a current population of individuals using crossover and mutation, which perform different roles. Mutation is a divergence technique, driving the population to discover new regions and enlarge the search space. Crossover is a convergence technique, driving the population towards a local optimum. The fitness of individuals is evaluated against a fitness function related to the optimisation problem being solved, subsequently the stronger candidates are selected to breed, the rest are 'discarded'. Since the ultimate 'goal' is to bring the population to a state of convergence, selection/crossover occur more frequently than mutation.

This process is iterative, in that the new generation of candidate solutions becomes the current population in the next iteration. The cycle terminates after the maximum number of iterations has been executed, or earlier if the fitness functions reach a satisfactory level. The advantages of GAs include their relatively simple application to new problems – merely requiring redefinition of the fitness function, and they are also effective and scalable, due to the "survival of the fittest" principle, according to which the unfit candidates are eliminated during the process.

GAs and EAs have a wide range of applications, such as in robotics, evolutionary machine learning, generative design applications and evolvable hardware. For example, GAs can accelerate the NN learning process to solve a certain problem, by learning the best hyper-parameters. This is illustrated in Figure 1.8.

Evolvable hardware (EH) is another field focusing on the use of EAs and GAs to create specialised hardware and electronics without manual engineering. Although it started out as a branch of electrical engineering and computer science, EH now brings together reconfigurable hardware, evolutionary computation, fault tolerance, sensors; connectivity and processing modules; and autonomous systems. In a broader sense, EH refers to any form of hardware that can change its architecture and behaviour dynamically and autonomously by interacting with its environment.

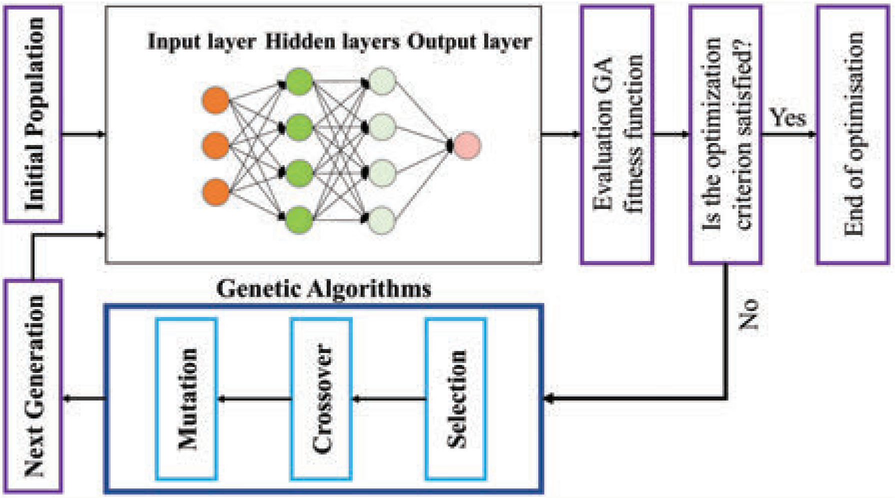

Initial Population
Input layer Hidden layers Output layer
Evaluation GA fitness function
Is the optimization criterion satisfied?
Yes
End of optimisation
No
Genetic Algorithms
Selection
Crossover
Mutation
Next Generation

Figure 1.8 Using Genetic Algorithms in the iterative process of fine-tuning NN hyperparameters.

Regardless of the industry, the generation and testing of different solutions is a critical part of every design process, including generative design. The generative design algorithm creates and tests different configurations, diverging to explore a large variety of solutions based on the pre-set requirements, and then converging on the best solution. Often such processes are cyclical/iterative, where initial requirements are adjusted, leading to another cycle/iteration of generating candidate solutions. Such processes can become very complex, hence the need for AI systems, such as GAs. Thus, one of the most powerful benefits of generative design is the speed with which new candidates can be generated and evaluated because the entire cycle is automated.

1.4.6 Generative AI

Generative AI is a branch of AI that enable computers to learn underlying patterns related to their input, which can be text, audio files or images, and then use these to create similar content [23]. While advances in ML have mostly been the result of discriminative modelling, the most significant advances in AI in recent years have been attributed to generative modelling, not least due to its ability to create new things.

In contrast to discriminative techniques that learn to classify data, generative AI techniques are mostly involved in creating new data based on training data. Discriminative modelling is focussed on learning a function that maps an input to an output using a labelled data set, a notion synonymous with supervised learning. Generative modelling is usually performed with an unlabelled data set, that is, as a form of unsupervised learning.

The differences between discriminative and generative modelling are best visualised in Figure 1.9. Discriminative models draw boundaries in the data space, focusing on predicting the data labels, while generative models try to model how data are placed throughout the space, focussing on explaining how the data were generated.

There are three main classes of generative AI techniques: general adversarial networks (GANs), autoregressive convolutional neural networks (AR-CNN), and transformer-based models.

GANs are a breakthrough, empowering deep networks with the ability to produce artificial content that passes for the real thing. GANs consist of two competing components – the generator network, which learns the distribution of classes, and the discriminator network, which learns the boundaries between those classes. Each network can be any type of neural network, such as artificial neural network (ANN). The discriminator must have fully connected layers with a classifier at the end.

GANs are the cutting-edge technology of AI, not least due to two essential key advantages: they solve the problem of generating data when there is not enough to start with, and they require no human supervision.

One of the practical applications of GANs can be seen in anomaly detection. Anomaly detection is known for identifying signal behaviours that

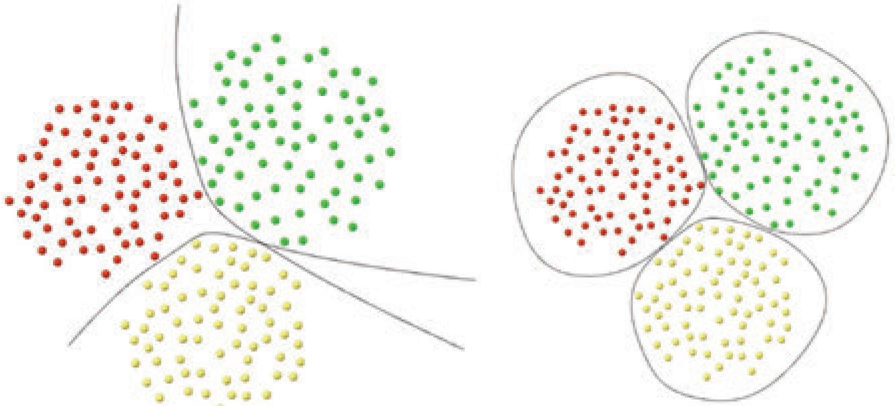


Figure 1.9 Discriminative (left) vs Generative (right) Models in ML.

do not fit the normal patterns and which can be addressed as a supervised learning problem. Depending on the application, this may require large, labelled data sets. However, in many industrial applications, samples from abnormal class may be insufficient for effective modelling. This is a challenge that can be addressed by another approach, using GANs (i.e., training only on samples considered ‘normal’ and then identifying the unusual, insufficiently available samples (abnormal) that differ from the learned sample distribution of normal).

An example of the network architecture of the generator and discriminator based on deep convolutional GAN is shown in Figure 1.10. In the training stage, only normal samples are involved. In the testing stage, abnormal samples can be discriminated by a higher anomaly score. The generator is trained only using the extracted features from normal samples. Anomaly scores are designed for anomaly detection.

Another practical application is GAN-based robotics control. Generative modelling helps reinforce ML models, so they are less biased and comprehend more abstract concepts, both in simulations and the real world.

GANs generate data that are like real data; therefore, they are widely used in industrial applications. They also have advantages over methods of supervised and unsupervised learning. A GAN is a semi-supervised learning

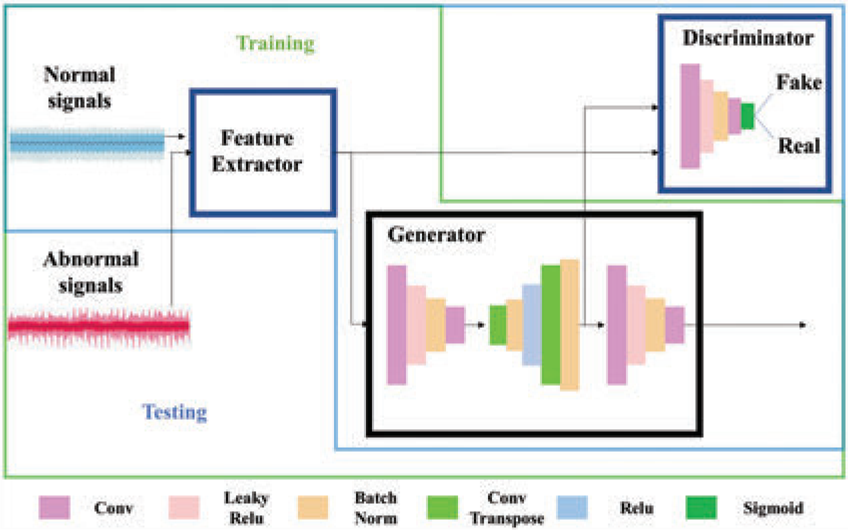


Figure 1.10 Network architecture of generator and discriminator based on deep convolutional GAN. Adapted from [24].

framework that uses manually labelled training data for supervised learning and unlabelled data for unsupervised learning to build models that can make predictions beyond the labelled data by leveraging the same.

The other two classes of generative AI techniques are AR-CNN and Transformer-based models. AR-CNN are used to examine systems that evolve, predicting future outcomes of a sequence from the previously observed results of that sequence. They rely on previous time-series data to generate accurate new data as an autoregressive model is a feed-forward model which predicts future values from past values. Transformer-based models are used to analyse data with a sequential structure and have become a standard tool for processing sequential input data, such as natural language. Core to their architecture is the ability to identify and learn context within an input sequence and thus refine the meaning of the other part of the sequence (the so-called attention mechanism).

1.4.7 Artificial Swarm Intelligence

Swarm intelligence is a branch of AI that is based on an extrinsic type of intelligence inspired from nature and biological systems and is connected to collective behaviour of decentralized and self-organised systems.

Swarm intelligence systems typically consist of many independent but similar individuals that follow very simple rules without a centralised control system. These systems' overall behaviour is a result of the interactions of the individuals, with each other and with their environment, but globally they act quite intelligently.

For example, ant colonies can optimise routes and shortest paths, and bee colonies can find the location of their nest in an extremely efficient manner. Swarm logic is a behaviour demonstrated by many animals, and while each individual is less capable of independently making decisions or solving problems, in a swarm they communicate, coordinate, organize, and seemingly problem solve, seemingly without a central command.

The essence of swarm logic is the sharing of information, along with interaction with other individuals and the surroundings, to derive new information as a basis for global actions. Adopting a broader perspective, swarm intelligence is the action of having decentralised agents swarm collectively towards a goal. These agents can be ants, bees, cars, robots, among other things. An IIoT system can be seen as multiple agents, where the intelligence lies both within agents and in their interaction with each other.

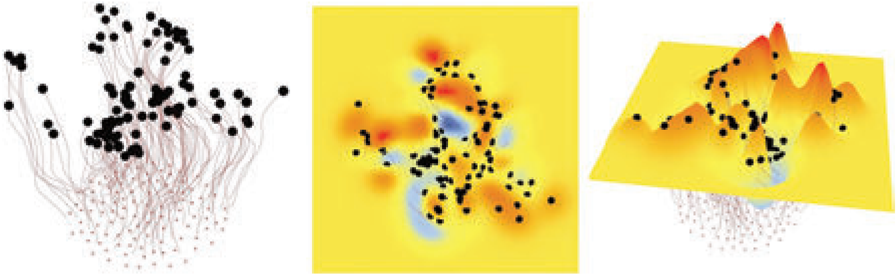


Figure 1.11 Swarm intelligence visualized: population of agents searching for a destination (left) and search space represented by a nonlinear regression generated surface (right).

Figure 1.11. intuitively illustrates the concept behind swarm intelligence, the starting point of which is a population of agents (like birds or bees) searching for a destination (left). Complicated intelligent behaviour to solve complex tasks emerges from simple agents following simple rules such as keeping diverging trajectories, avoiding collisions and interaction with near neighbours (rule is known as self-organization). These agents will simultaneously know when the destination is reached, based on the goal parameters of the destination. Swarm intelligence's aim is to optimise the goal parameters and minimise the search space, represented by a surface generated using nonlinear regression (right).

1.4.8 Natural Language Processing

Natural language processing (NLP) is a branch of AI that focuses on developing algorithms to enable computers to understand speech and text. NLP systems are developed to imitate the human capacity to use language. NLP is used in a variety of tasks, including text understanding, text summarization, information extraction, machine translation, and speech recognition and synthesis. Examples of AI techniques include support vector machine (SVM), for text classification (such as spam detection); hidden Markov models (HMMs) for speech or text generation; neural networks, for machine translation; and logic-based methods, for text summarisation.

NLP technology has made major progress in recent years, leading to the development of network architectures better able to learn from complex and context-sensitive data. These advances have been supported by the constantly increasing data resources from intelligent sensors and computing power.

Current challenges include obtaining quality data and detecting and removing data biases. Future applications are expected to meet these challenges, as well as to improve human–AI interactions across diverse languages and situations.

NLP is too computationally expensive to run on microcontrollers, so applications running on edge devices are often limited to looking for keywords in speech, such as short commands for executing some actions. Identifying non-voice sounds is also extremely useful. NLP based on embedded machine learning will make edge devices more intelligent in future applications.

1.4.9 Machine Learning

Machine learning is a branch of AI that provide systems with the ability to automatically learn and improve their performance in some tasks through experience without being explicitly programmed. The rules of ML programs are not determined in the same way as those of normal computer programs are; instead, ML uses specialised algorithms to *learn* rules from data, in a process known as *training*.

This training process starts with feeding data and then training the machines by building various models using different algorithms. The choice of algorithms depends on the kind of task we are attempting to automate. Most ML tasks are narrowly specified to optimise specific functions using particular data set.

Inference is the process of taking a trained model and deploying it into a device, which will then process incoming data to look for and identify whatever it has been trained to recognise.

During the inferencing phase, predictions and decisions are made concerning new data, based on the learned parameters. Prediction is the process of using a model to make a prediction about something that has yet to happen. Inference is the process of evaluating the relationship between the predictor and response variables.

DL and neural networks are examples of **ML techniques** frequently used today. DL systems learn from large amounts of data to subsequently recognise and classify related, but previously unobserved, data. For example, **neural networks**, often described as being loosely modelled after the human brain, consist of thousands or millions of processing nodes generally organised into layers. Advances in hardware have allowed these networks to have many layers, which is what puts the “deep” in deep learning. What differentiates DL from ML techniques is the former’s ability to extract features automatically.

Humans and machines both acquire knowledge in the process of learning based on experience; however, the former do so based on either direct or shared experience, while the latter do so through experience shared in the form of past data. With respect to which input data an ML process receives and how it handles this data, three types of **ML training methods** can be distinguished: supervised (labelled data required), unsupervised (no labelled data; these attempt to discover patterns) and reinforcement learning (actions taken to maximise cumulative rewards).

Supervised-learning algorithms learn from labelled input data and are widely used for classification and regression tasks. The system learns which components of the data are useful for classifying it correctly and uses that information to correctly classify data it has not encountered before. Such algorithms can also detect patterns in data and then use the uncovered patterns to predict future data or other outcomes of interest. By contrast, **unsupervised-learning** algorithms seek to discover hidden patterns and other underlying structures in unlabelled data and are used in clustering tasks.

Reinforcement learning algorithms enable computer programs to learn from experience and to be rewarded for reaching specified objectives – both immediate actions and long-term goals. Reinforcement learning is akin to how humans learn from their own mistakes over time through trial and error. This means that the algorithm decides the next action by learning behaviours that are based on its current state and that will maximise the reward in the future.

More detailed description of the learning algorithms can be found in Section 1.8.1.

1.4.10 Neural Networks

This section presents a high-level overview of neural networks (NNs) thus providing a valuable and intuitive understanding of how the models work when deployed in edge devices and operated within industrial settings.

Neural networks simulate the learning capacity of biological neurons in the human brain. The fundamental unit of a neural network is a perceptron (Figure 1.12).

The perceptron model multiplies all inputs with a weight parameter, whose value is representative of how important each feature is in the calculation of the results. The resulting values are added together with a bias term, resulting in the so-called weighted sum, on which an activation function

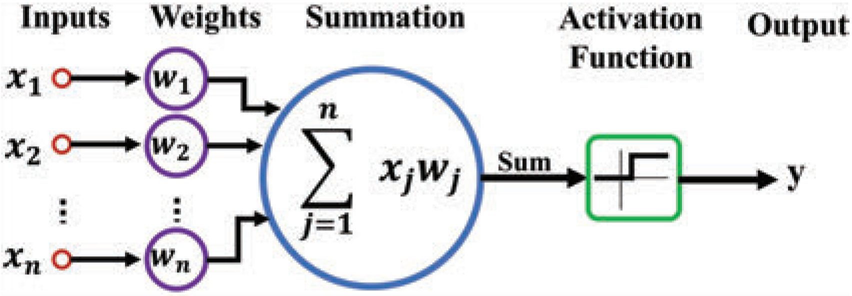


Figure 1.12 Perceptron illustration.

is finally applied. The activation functions introduce non-linearity in NN models, thus differentiating them from linear regression models.

During training, model parameters are gradually calibrated so that the NN output comes increasingly nearer to the desired one, when given a certain input. A loss function keeps track of how far the model is from predicting the correct output, meaning that the higher the loss value, the less efficient the model is at predicting. Accuracy is another metric, inversely proportional to the loss.

Anyone interested in more detailed reading regarding training is directed to [4][8] and other sources. Generally, the first step is to forward feed one signal sample or a batch of samples through the network. Feedforward is the process of passing input values, through the hierarchical layering of neurons, to produce an output in the final layer. Network loss is then calculated by comparing the predictions with the actual outputs and this is then used to update the model's parameters in the next step known as backpropagation. Backpropagation is the process by which the error contribution of each neuron is calculated and passed backwards through the network. The weights and biases are adjusted proportionally to this error contribution, and this is how the machine learns.

This stepwise procedure is run several times with the aim of improving the output of feedforward, ultimately optimising the network's predictions. One feedforward and backward pass is called an iteration, while a pass of the entire data set is called an epoch. After each epoch, the algorithm will perform a forward pass of each validation sample, looking at loss and accuracy. Usually, accuracy improves over time as loss drops.

The number of epochs and the learning rate, i.e., how much the model's internal parameters are updated during each training step, are known as hyperparameters, and can be configured to make a model more efficient.

1.4.11 Automated Planning and Plan Recognition

Automated planning is a branch of AI that concerns providing goal-oriented, deliberative behaviour to both physical and virtual intelligent agents [29]. It takes as inputs a planning domain, an initial state and a goal, and it employs optimisation algorithms to return a sequence of actions that guides the agent's behaviour. The correct representation of states, conditions and actions and the suitable algorithms all contribute to the agent reaching its goals and optimising performance.

In many industries, automation is an emergent trend that requires efficient automated planning, such as robotic and autonomous systems. Mobile and fixed robotic systems can perform various tasks in the industrial application domain without the need to acquire knowledge, relying on only the accuracy of the model. The model encompasses explicitly represented domain knowledge acquired from human experts.

Incorporating AI capabilities in automated production planning in manufacturing also has significant potential. Embedded industrial AI optimisation

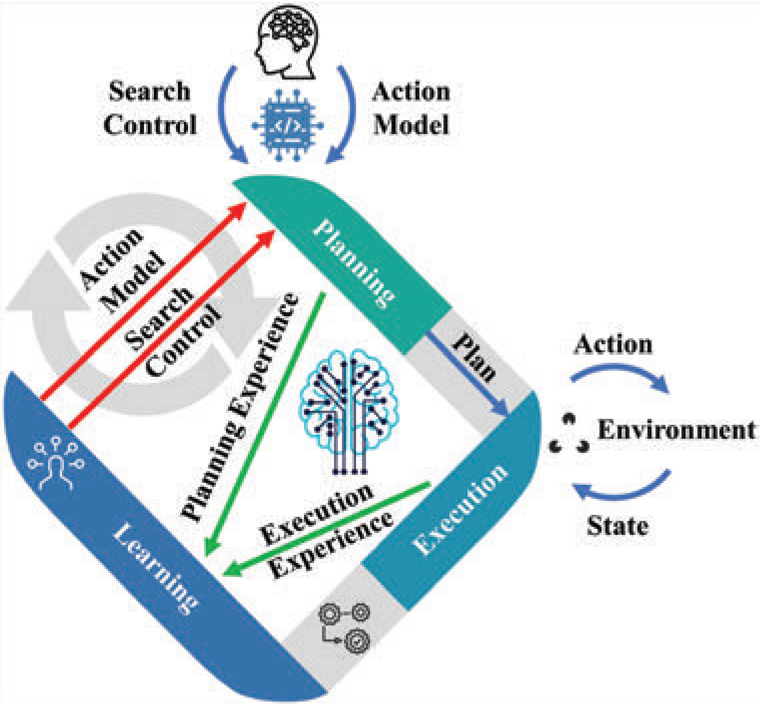


Figure 1.13 Automated planning, states, and actions.

algorithms can balance the product result and the resources used during production and can learn by collecting considerable amounts of "cause and effect" data that can be used for what-if simulations and analysis.

Embedded edge AI solutions for path planning for swarms in mobile autonomous systems are evolving, and they have been applied in several manufacturing optimisation and logistics applications. Swarm automated planning algorithms are used as planning methods based on planning graph technology to improve the searching efficiency using swarm intelligence in fleets of autonomous devices operating on the manufacturing floor.

AI planning techniques are widely used when explainability is necessary, i.e. the planner can explain why a specific course of action has been chosen.

On the downside, there are some challenges to AI planning techniques when they are employed in real-time applications, due to slow response time. The more complex the planning domain, the larger the search space becomes, thus increasing the response time to find a proper sequence of actions. This is especially critical when planning and acting are intertwined.

An alternative to acquiring knowledge from human experts is to learn the model in time. Architectures relying on ML have the advantage of not requiring much prior knowledge about the domain; once trained, they act quickly. Nevertheless, they need a large amount of data for the training. They are usually limited to the industrial application domain they were trained for and presenting them with new situations might be challenging.

The two approaches can be incorporated into the same agent architecture, thus achieving better trade-offs than if only one approach were used. More about the synergistic benefits of combining symbolic AI and ML can be found in Section 1.6.

Plan recognition deals with inferring the goals or plans that explain the observed actions of an agent; as such, it is considered the opposite of planning. Plan recognition algorithms require knowledge about the potential behaviours of the agent and how the agent makes its decisions. When this knowledge is unavailable, neural networks can be employed to learn the decision model automatically.

1.4.12 AI for the Metaverse

Metaverse is a term formed by combining meta and universe and has been introduced as a shared virtual world that is fuelled by many emerging technologies, such as virtual reality, and AI. AI has shown the great importance

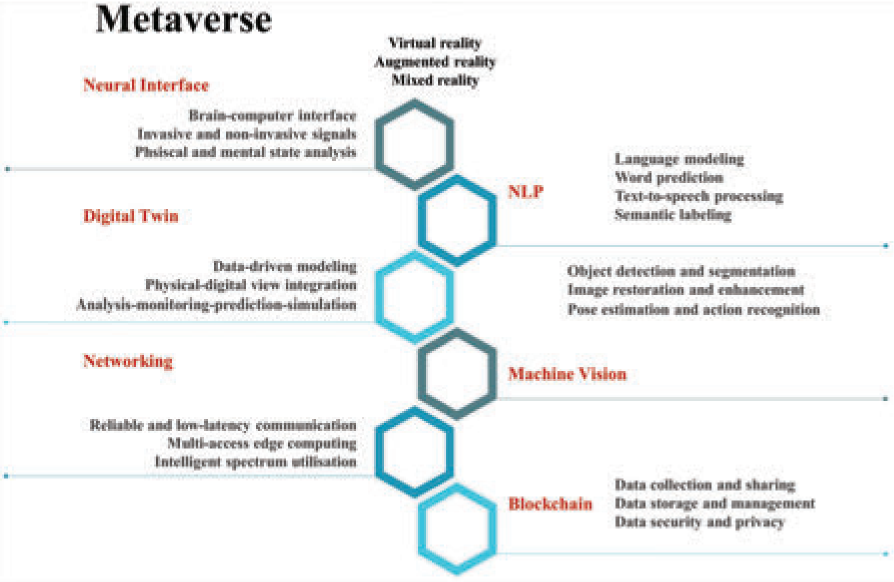

Metaverse
Virtual reality
Augmented reality
Mixed reality
Neural Interface
Brain-computer interface
Invasive and non-invasive signals
Phisical and mental state analysis
NLP
Language modeling
Word prediction
Text-to-speech processing
Semantic labeling
Digital Twin
Data-driven modeling
Physical-digital view integration
Analysis-monitoring-prediction-simulation
Object detection and segmentation
Image restoration and enhancement
Pose estimation and action recognition
Networking
Machine Vision
Reliable and low-latency communication
Multi-access edge computing
Intelligent spectrum utilisation
Blockchain
Data collection and sharing
Data storage and management
Data security and privacy

Figure 1.14 Application of Metaverse.

of processing large amounts of data to enhance immersive experience and enable human-like intelligence of virtual agents.

ML algorithms, DL architectures and other emerging technologies such as swarm intelligence have had a role in the foundation and development of the metaverse, such as AR, VR, mixed reality (MR). These are now ready to be employed in applications such as machine vision, blockchain, networking, digital twin, and in different industrial applications (Figure 1.14).

1.5 Edge AI Continuum

Edge processing can redefine the landscape of interconnected devices by moving data processing and analytics to the edge and employing AI techniques and embedded security. Edge AI computing and processing allow for the development of new real-time applications due to the processing being performed close to the data source. It can reduce the amount of transmitted data by transforming extensive amounts of raw data into essential insight data. It can also decrease communication bandwidth and data storage requirements while reducing energy consumption and increasing security, privacy, and data protection.

Edge AI technology developments are used to implement applications that benefit from AI-based technology advances across the edge continuum.

Various forms of AI have already been adopted by multiple industries, governments, and society. However, a breakthrough is needed in several industrial sectors to bring the intelligence close to the data source and implement it in industrial processes. However, this breakthrough may face several hurdles that challenge its advancement.

Leveraging AI methods and techniques at the edge is essential for increasing the performance and capabilities of the intelligent sensor systems and IIoT devices used in various industrial applications. The edge AI processing concept is reflected in the emergence of micro-, deep-, and meta-edge layers for several industrial intelligent applications.

The edge processing continuum includes sensing, processing, and communication units close to physical industrial assets (micro edge), gateways and intelligent controller processing (deep edge), and on-premises multi-use computing (meta edge). This computing continuum creates a multi-level structure that advances processing, intelligence, and connectivity capabilities.

The edge AI processing concept for intelligent applications is mirrored in the development of different edge-processing levels. Figure 1.15 shows an all-encompassing edge AI architecture incorporating the computing

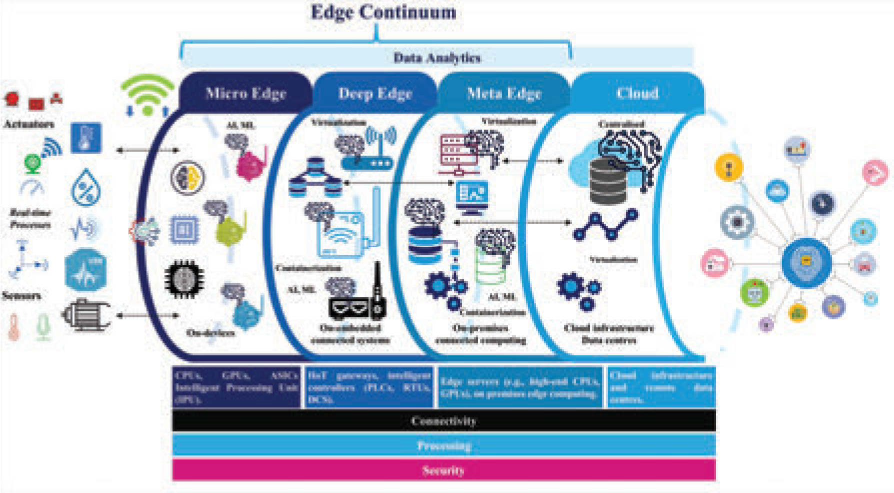

Edge Continuum
Data Analytics
Micro Edge
Deep Edge
Meta Edge
Cloud
Actuators
Real-time Processes
Sensors
AI, ML
On-devices
Virtualization
Containerization
AI, ML
On-embedded connected systems
Virtualization
AI, ML
Containerization
On-premises connected computing
Centralised
Virtualization
Cloud infrastructure
Data centres
CPUs, GPUs, ASICs, Intelligent Processing Unit (IPU).
IIoT gateways, intelligent controllers (PLCs, RTUs, DCS).
Edge servers (e.g., high-end CPUs, GPUs), on-premises edge computing.
Cloud infrastructure and remote data centres.
Connectivity
Processing
Security

Figure 1.15 Edge AI Architecture.

and intelligence continuum from sensors and actuators, processing, units, controllers, gateways, and on-premises servers to multi-access, fog, to cloud computing interfaces.

Edge AI computing and processing device functions cover edge computing, communication, and data analytics capabilities, which make it smart/intelligent. An edge AI computing and processing device is designed around the computing units (CPUs, GPUs, FPGAs, application specific integrated circuits (ASICs), AI accelerators/processing), communication networks, storage infrastructures, and applications (workloads) that run on it. Single- and multi-core microcontrollers (MCUs) are based on ARM Cortex-M cores or on cores using new open-source RISC-V instruction set architecture (ISA) and high-performance embedded processors with varying capabilities. The memory footprint, computing time, transmission, and power consumption requirements always depend on whether the device operates at the micro, deep, or meta edge. ML and DL models need to be converted into efficient formats before compiling and flashing them into edge devices.

AI building blocks are optimised for the type of processor, the amount of RAM, and the number and types of sensors. The solutions are usually provided as a C library, which can be embedded into the main microcontroller program and compiled and downloaded into the embedded system.

The edge can scale from a few devices to tens of thousands of devices distributed in various locations with unique identities. Edge AI computing and processing devices are physically separated, yet they can be connected by wireless/wired topology connections, such as mesh topologies. Edge AI computing and processing devices can operate independently, and local decisions can be supported by inference actions, including the unexplored evolution of training on edge devices.

AI models increase various potential industrial applications; however, developing AI functionalities for the edge continuum is complex and presents several challenges, such as scalability, interoperability, and performance optimisation versus the resource constraints of the edge devices. Overall, implementing AI models on edge-embedded devices has advantages for different use cases and applications in various industrial sectors.

A key element for the transition of AI processing to the edge is the capabilities of the developer edge environment, covering the hardware, interfaces, platforms, training/learning, applications, and services. The intelligent infrastructure at the edge refers to the tools, platforms, and techniques used to run store data, build, and train AI/ ML algorithms, and the algorithms themselves.

Micro-edge

The micro edge includes intelligent sensor systems (physical, chemical, environmental parameters, perception, etc.) with processing and connectivity capabilities that use IIoT devices that generate insight data and analytics. Micro-edge devices are implemented using microcontrollers built around ARM Cortex M0, M0+, M3, M4, M7X, ASICs and RISC V. The distance from the data source (sensors) is minimised, and the micro-edge devices have cost and power consumption constraints. Micro-edge hardware devices implement analytics and intelligent functions by integrating AI-based components and algorithms and running the AI algorithms for inference, training, and self-training. The intelligent micro edge makes IIoT real-time applications ubiquitous and merges them with the industrial environment.

Deep-edge

The deep edge comprises intelligent controllers, PLCs, SCADA elements, connected machine vision systems, networking equipment, gateways, and computing units that aggregate data from the sensors/actuators and IIoT devices. Deep-edge processing capabilities are implemented with performant processors and microcontrollers, such as Intel i-series, Atom, ARM Cortex M7+, etc., including CPUs, GPUs, TPUs, FPGAs and ASICs. The system architecture, including the deep edge, relies on foreseen functionality and deployment options. These functions include cognitive capabilities that can acquire, aggregate, understand, react to data, exchange, and distribute information.

Meta-edge

The meta edge integrates processing units, typically on-premises, implemented with high-performance embedded computing units, edge machine vision systems, and edge servers (e.g., high-performance CPUs, GPUs, FPGAs, etc.), which are designed to handle compute-intensive tasks (e.g., data series, image, and video processing), advanced analytics, AI-based functions, networking, and data storage.

Fog computing extends the computing capabilities of meta-edge industrial systems and interfaces cloud computing capabilities with the edge of the network. Fog computing enables repeatable structures in the edge computing concept, so enterprises can push computing out of centralised systems or clouds for a better and more scalable performance. A Fog computing implementation is a virtualised platform located between cloud data centres (hosted within the Internet) and meta-edge that provides support for edge processing and is complementary to cloud computing platforms.

1.6 Symbolic AI – ML Continuum

Human intelligence comes in two distinct but complementary forms of arriving at conclusions, one based on structured and rational decisions and the other on perception and understanding patterns. Machine intelligence also comes in two similar forms, one based on symbolic knowledge representation and reasoning (the symbolic AI approach) and the other on deep learning and the interpretation of data patterns (the ML approach).

Therefore, when faced with an AI problem, one can look for a solution combining technologies in the symbolic AI – ML/DL continuum, instead of choosing between the symbolic or ML/DL approach in solving it. Nevertheless, it is essential to understand the difference between and the advantages and disadvantages of these two approaches.

Generally, the symbolic AI approach is suitable when the AI problem is abstract, no large amounts of data about the AI problem are available (for example, data coming from sensors, such as images, sounds, etc.) but the steps to the solution are commonly known so that this knowledge can be modelled explicitly.

On the contrary, ML is useful when the steps to the solution are not known, but the large amount of data allows us to look for larger patterns, which may ultimately lead to the likely solution. This approach requires several iterations and massive computational power to arrive at a conclusion. Nevertheless, as computing hardware becomes faster and cheaper and ML algorithms become more powerful, ML becomes more inclusive (i.e., available not only to actors with strong computational resources).

The concept is easier to grasp if we consider a simple use case of the automated diagnosis of a malfunctioning motor problem. In the case of symbolic AI approaches, this would require that a human expert fully describe the motor and its features, functioning and malfunctioning situations. This knowledge would then be represented in a form that could be processed by machines. With the help of algorithms and step-by-step inferences from this knowledge base, it is possible to arrive at a diagnostic for the motor problem when fed real-time sensor data.

The advantage of this approach is that it does not rely on massive data and might work for most motors. However, the knowledge base takes time to acquire and set up, and if some knowledge (about a particular motor) is missing or incomplete, it will yield no result.

The ML approach would be to feed a neural network with many signals/data of the motor, vibration data and audio data in both functioning and malfunctioning situations. The trained network would then be able to accurately guess the motor diagnosis when fed real-time sensor data. The advantage of this approach is that it does not rely on a motor expert's knowledge to be made explicit, and it allows for automation due to its ability to handle large amounts of real-time sensor data.

It is technologically possible to combine symbolic AI and ML, for example, by using symbolic AI to generate answers (constraints) and then feeding these answers to ML to generate predictions. A balance between the two can be achieved based on experimentation.

In short, with symbolic AI, the rules of the AI algorithms are decided by a human. These rules and some data are provided as input to the AI algorithms, and data are processed according to these rules to produce answers in the output. With ML, on the other hand, the inputs to the ML algorithms during the training process include some data and answers, while the rules are the output. These rules are then used during inference to produce predictions about input data that have not been seen before (i.e., data that was not part of the training).

Therefore, AI can also be understood from the perspective of combining technologies in the symbolic AI – ML continuum and balancing them to achieve better trade-offs than otherwise achieved if only one technology were used.

1.7 Logic-based AI: Knowledge Representation and Reasoning

As full-scale AI applications increase in number and complexity, accelerating digital innovation across industries and boosting productivity, so does the need for AI to be more comprehensible, explainable, and therefore trustworthy. Thus, symbolic approaches to classical AI are re-gaining momentum. This section summarises some of the logic-based approaches that are likely to be adopted by various industrial sectors and discuss future perspectives for exploiting logic-based technologies.

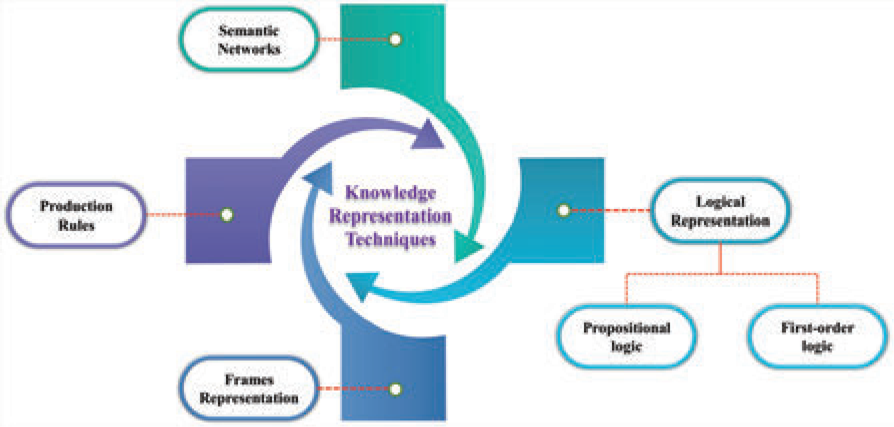


Figure 1.16 Knowledge representation.

Intelligent machines require knowledge to make intelligent decisions, the same way as humans do. This usually entails that expert knowledge need to be acquired and represented in a form that machines can process, called a knowledge base. Predicate logic and propositional logic are representative ways to reflect knowledge; semantic networks, rules, frames, or programming languages are also good examples (Figure 1.16). Languages that are designed specifically for AI include LISP and Prolog.

A knowledge representation should have specific properties, for example be unambiguous, easy to use, inferential adequate and efficient, and able to represent all types of knowledge: declarative, procedural, heuristic, structural, meta-knowledge (Figure 1.17). The choice of knowledge representation method largely depends on the problem to solve.

Inference is a term representing the derivation of new knowledge from existing knowledge and axioms (i.e., rules of derivation) within a single step, using logical constructions. The rule of derivation can be one of many kinds, such as, induction, deduction, and abduction. Modus ponens (*if A is true, then B is true. A is true. therefore, B is true*) and modus tollens *(if A is true, then B is true. B is not true. therefore, A is not true*) are two such logical argument constructions.

Reasoning is a term used in the context of a goal (e.g., proof whether a propositional statement is satisfiable or not) and is carried out via a search process involving multiple inferences. Choices during such search have to be made such as which axiom to “fire” along with which knowledge in order to

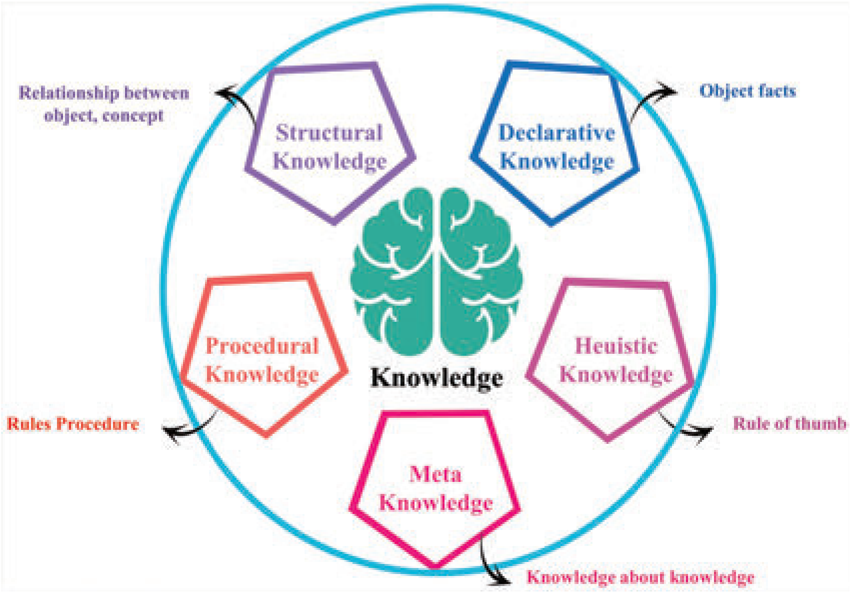


Figure 1.17 Type of knowledge.

derive new knowledge. Resolution is a particular kind of reasoning involving the “resolution rule”.

Reasoning from premises to logical consequences, have been a major part of AI since its beginnings. Inferences are steps in reasoning, moving from premises to logical consequences. One motivation behind is the utilization of knowledge of a domain for obtaining answers for given problems. In this case knowledge must be available in a formal form like propositional or first-order logic. Reasoning and mechanical theorem proving is used for computing an answer using the formalized knowledge directly. It is worth noting that this kind of application of logic comes with several advantages, i.e., making knowledge explicit (and thus understandable for humans), allowing to use the same knowledge for various problems, and allowing to explain solutions based on knowledge. On the downside, logical theorem proving requires high computational resources, but which are widely available today.

For more information about the foundations of logic (and in particular propositional and first-order logic) we refer the interested reader to [8] (with the direct context to AI).

To solve this problem, other classes of (non-monotonic) logic has been proposed like default logic [15], and abduction, which is also non-monotonic.

In the past decades, research in non-monotonic logics and their applications has been a very active part of AI. This includes model-based reasoning [17][18] with a strong relationship to default logic, and also more recently answer set programming (ASP) [19]. All these inference mechanisms can be used to solve practical challenges, like diagnosis and fail-operational behaviour. More about these topics and reasoning from first principles for self-adaptive and autonomous systems can be found in [20].

Logical inference has been an active research area of AI since its beginnings, ranging from expert systems to more recent developments on non-monotonic inference. Due to the increased available computational power and the availability of efficient reasoning and inference engines the direct use of knowledge formalized in ontologies and knowledge bases for solving various tasks can be achieved. Recent work describing a mapping from neural networks to a logical representation can be found in [21][22].

1.8 Hardware/Software Technology Stack

Technology stacks are widely used to structure technologies in a particular area. AI is no exception, as it is possible to conceptualise AI as a technology stack with various layers. A five-layer stack is presented in Figure 1.18.

During the past decades, the focus has moved back and forth between logic (symbolic reasoning) and pattern recognition (neural networks), driven by the varying abilities of technologies to acquire data, learn, derive new information and reason to reach decisions. In the last years, machine learning and neural network models have been the primary focus due to advances in hardware development and processing capabilities. Hence, the technology stack is illustrated by machine and deep learning, covering topics such as learning/training and inference.

The foundation of the stack is represented by the hardware layer, which contains at least three sets of components that reflect the processing units responsible for performing specialised AI operations. The neuromorphic hardware components consist of new ultra-low-power silicon chip architectures (e.g., neuromorphic modules and chips, analogue NN, spike NN) that incorporate different chip designs and algorithms to mimic how the human brain works. The accelerator set of components consists of silicon chips designed to perform the highly parallel functions required during training and inference, such as GPUs, FPGAs, or ASICs. The head node components are units that coordinate computations among accelerators.

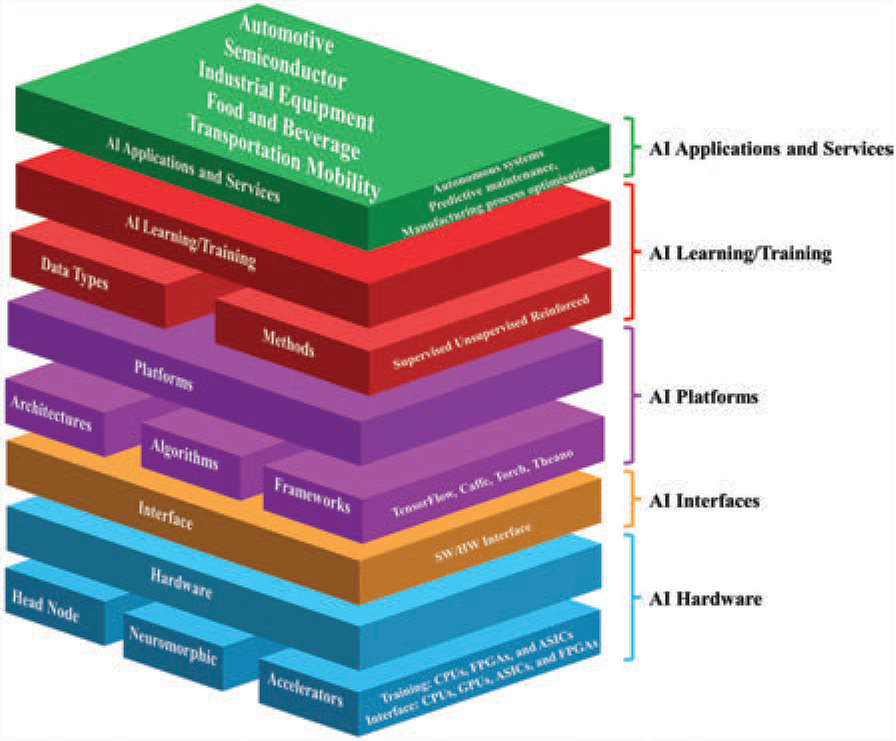


Figure 1.18 Five-layer (with sublayers) AI technology stack.

The platforms layer is used for AI and ML/DL deployment and consists of three sublayers, the goal of which is to abstract firmware from the underlying hardware. The frameworks sublayer consists of packages that trigger HW algorithms, such as Caffe, Torch, Theano, etc. This happens through the interface layer, which connects the hardware and platform layers and is in charge of facilitating communication between them. The algorithms sublayer consists of rules to achieve optimal inference according to the training method employed, such as backpropagation, evolutionary, and contrasted divergence. The architectures sublayer consists of many continuously evolving neural network architectures, such as CNN, RNN, etc.

The AI training/learning layer consists of two sublayers. The methods sublayer involves techniques for optimising the model for specific domain data, such as supervised, unsupervised and reinforcement learning. The data type sublayer consists of categories of domain input data, such as labelled and unlabelled data.

Finally, the applications and services layer incorporate ready-to-use AI functionality into solutions to real industry problems and use cases, such as autonomous vehicles and object recognition. The solutions can be customised based on generic data or on customer-specific training data.

The AI technology stack provides a common understanding of the AI layers and components when implementing and benchmarking various AI technologies and applications. The elements presented in the different sections - spectrum, continuums, methods, techniques, concepts, and others - are all connected through the AI technology stack defined by European projects such as AI4DI [3].

This section briefly introduces the industry-adopted ML terms and the ML methods such as supervised, unsupervised, and reinforcement learning, and neural networks architectures. Specifically, the focus is on embedded ML, for which the advances in hardware architectures opened an entirely new space of applications and opportunities. The new hardware architectures make possible to run complex ML workloads on microcontrollers, with limited compute and memory profiles.

1.8.1 ML Methods and Techniques

There are a multitude of methods and techniques that depend on the type of learning, and the type of learning – supervised, unsupervised, or reinforcement depends on the data available for the application. A taxonomy is shown in Figure 1.19.

Supervised Machine Learning

Supervised learning algorithms learn from a training set of data that is labelled with the correct description; the system subsequently learns which components of the data are useful for classifying it correctly and uses that information to correctly classify data it has never encountered before. These algorithms are widely used for classification and regression tasks, as detailed below.

Regression is considered the fundamental ML paradigm. The process of regression connects outputs to inputs. It shows an output for a given input, and the regression component creates a transfer function to best fit that

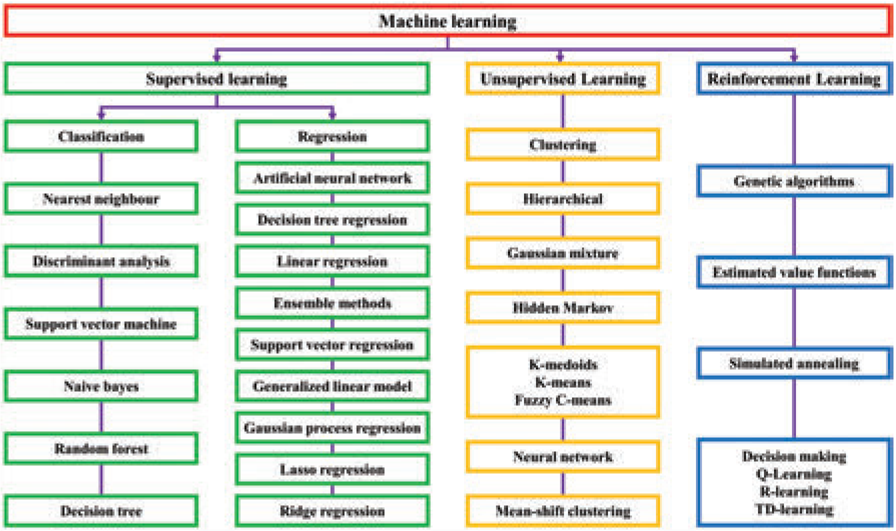


Figure 1.19 ML taxonomy.

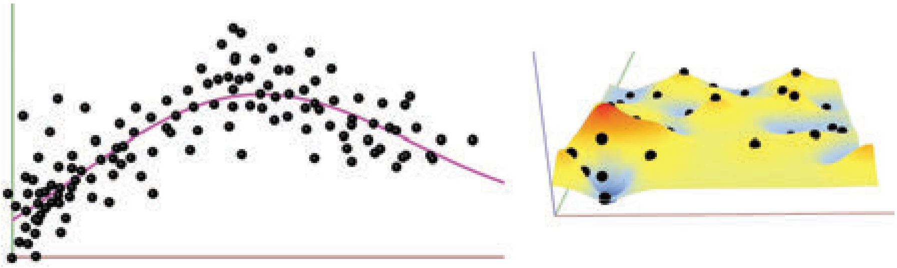


Figure 1.20 Regression visualized 2D (left), 3D (right).

data. This transfer function then provides a method to predict an output for an untested input. In other words, if the independent variable is time, then the model forecasts future values; otherwise, the model predicts present but unknown values. Typically, when selecting a regression strategy, the number and type of inputs and the type of transfer functions need to be considered. The transfer functions can be represented as curves (Figure 1.20 left) and surfaces (Figure 1.20 right).

There is a wide variety of regression strategies employed in industrial applications such as simple linear regression, polynomial regression, logistic regression, support vector for regression (SVR), decision tree regression, random forest regression. Figure 1.21 shows an example of logistic regression

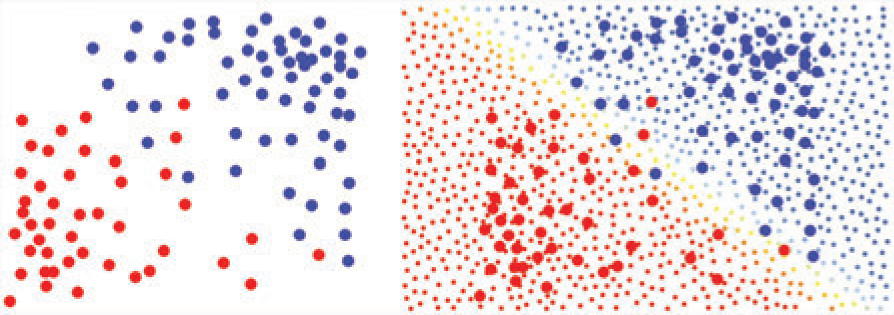


Figure 1.21 Normal(blue) - abnormal(red) (left). Predicted values using logistic regression (right).

that predicts a binary outcome, such as normal or abnormal, based on observations of the data set, which could be motor vibration measurements. The large dots are the learning data, while the small dots are the data to test against learning data.

Classification addresses the problem of determining the class that a given data instance belongs to. It requires more input, i.e., training data must be provided for the definition of classes. The more training, the more accurate the classification algorithm. Given sufficient training data, classification tools can distinguish between classes as well as or better than humans.

Many of the most powerful applications of ML are classification systems. Neural networks based on the layered architecture of biological brains have emerged as a common classification technique because they are able to group explicit, visible features into abstract or inferred features that correlate closely to the predefined classes in the training data.

Classification methods are widely used in machine vision with the classification of images, e.g., to determine whether an image contains specific objects. Another example is with time series, e.g., motor classification in predictive maintenance. Among their other benefits, classification tools can extend automation to incorporate the ability to differentiate inputs automatically, alleviating one of the most time-consuming manual steps in the generative-design workflow.

The intuitive images in Figure 1.22 show how the two classification and regression can be distinguished. Regression searches for a line or plane that fits the given input points, while classification searches for a line or surface to separate the classes.

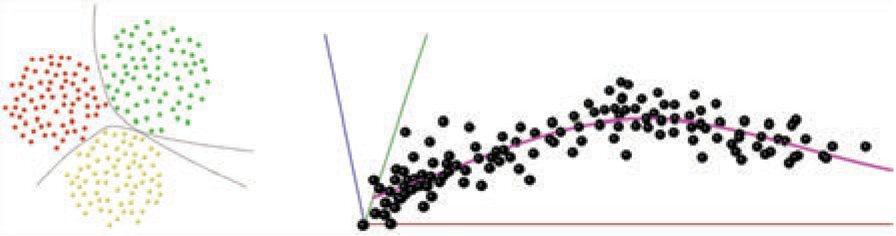


Figure 1.22 Classification (left) vs Regression (right)

Unsupervised Machine Learning

In contrast to supervised learning, unsupervised learning algorithms search for underlying structures in unlabelled data. Unsupervised learning is where there is only input data and no corresponding output variables. The goal for unsupervised learning is to model the underlying structure or distribution in the data to learn more about the data. These algorithms are widely used for clustering and dimension reduction tasks, as detailed below.

Clustering is one of the most flexible techniques in ML: it is easy to apply and requires no sample data or predetermined classifications. Clustering algorithms group together data with similar characteristics without any prior training or guidance on how to distinguish between groups. This is very powerful precisely because it is so flexible. The raw data and the number of groups are given as inputs, and the clusters are generated as outputs. K-means is one of the most used methods for clustering, where K is the number of clusters to be created. In short, the algorithm places the centres of the K clusters in the data set, assigns the closest points to the K cluster and recalculates the centre of the cluster iteratively. Another powerful clustering algorithm is the Gaussian mixture models (example in Figure 1.23). The better the data describes different features within the data, the better or likelier the grouping result.

Clustering is powerful in its flexibility and simplicity. Often, clustering is the starting point when organising poorly structured data or sorting continuous data into useful groups. On the downside, the results are difficult to control precisely and depend on the resolution of the input data.

Dimension reduction is used to simplify the model by removing the less important or redundant information from the data set to make it manageable while maintaining relevance and performance. Data sets can sometimes have hundreds of features, and by extracting fewer independent features, the

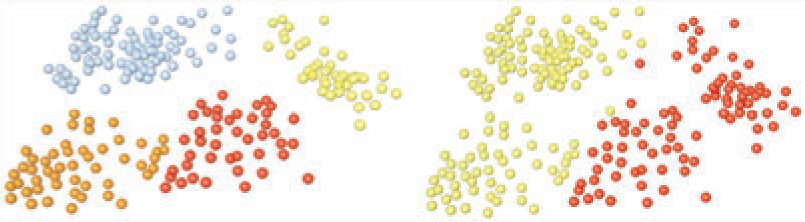


Figure 1.23 Cluster (Gaussian mixture models) 4 clusters (left) vs 2 clusters (right).

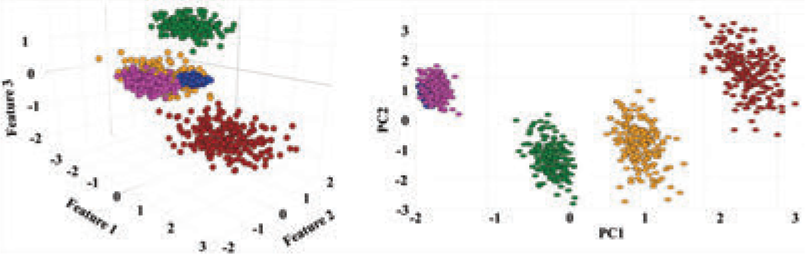


Figure 1.24 Principal component analysis. Intuitive visualisation, select variables that capture the largest variability in data.

complexity of the model can be greatly reduced. The most used algorithm is Principal Component Analysis (PCA), which finds new vectors that maximise the linear variation of the data by drastically reducing the size of the data without losing too much information (Figure 1.24). Another commonly used method is t-Stochastic Neighbour Embedding (t-SNE), used for automatic learning by reducing the space of functions.

There are two types of dimensionality reduction techniques: feature selection and feature extraction. Feature selection techniques are backward elimination, forward selection, bidirectional elimination, score comparison and more. Feature extraction techniques are, Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), Kernel PCA, Quadratic Discriminant Analysis (QDA).

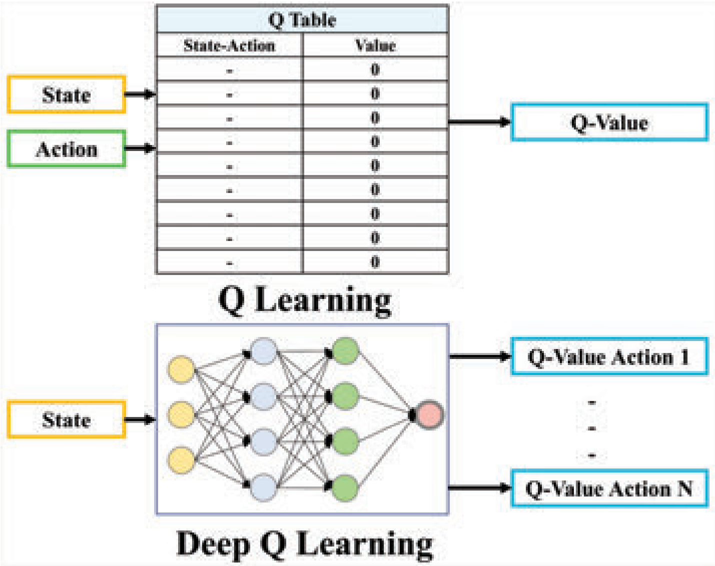
Reinforcement Learning

Reinforcement learning (RL) enables computer programs to learn from experience by trial and error and to be rewarded for reaching specified objectives – both immediate actions and long-term goals. The two main components are the environment, which represents the problem to be solved, and the agent, which represents the learning algorithm.

Different from other ML approaches is RL's emphasis on simulated motivation and learning from direct interaction with humans and the environment, without requiring explicit examples and models. RL is akin to how humans learn from their own mistakes over time through trial and error. This means that the algorithm decides the next action by learning behaviours that are based on its current state and that will maximise the reward in the future. RL shifts the focus of machine learning (ML) from pattern recognition to experienced-based sequential decision-making and execution. Many applications in robotics and machine vision use RL to perform tasks.

One of the core concepts in RL is the Q-Learning, which is about learning an action-value function, representing the measure of the overall expected reward assuming the agent performs the action. A simple data structure such as a table can be used to keep track of the states, actions, and their expected rewards. In case of an infinite state space, this function is implemented with DNNs, hence the term deep Q-learning illustrated in Figure 1.25.

Deep RL has demonstrated great potential for addressing the challenges of real-time decision-making based on information captured by sensors. The increased complexity of sensor-intensive systems with expensive subsystems and costly repairs requires efficient real-time control and decision-making



State-Action	Value
-	0
-	0
-	0
-	0
-	0
-	0
-	0
-	0
-	0

Figure 1.25 Q-Learning vs Deep Q-Learning.

approaches. Thus, many research efforts have recently been devoted to applying deep RL to the field of predictive maintenance [28].

1.8.2 Neural Networks Architectures

An artificial neural network (ANN) encompasses any form of a DL model and can have one hidden layer connecting the input and the output. DL is a class of machine learning algorithms that uses multiple stacked layers of processing nodes to learn high-level representations from data, such as images, audio, and text. ANN can have many hidden layers, in which case they are called “deep”, hence the term deep neural network (DNN). By adding more hidden layers, the model gets more parameters, which in turn allows the model to fit more complex functions.

A DNN consists of a series of stacked layers, and each layer is made up of nodes that are connected to the previous layer’s nodes through a set of weights. By stacking layers, the nodes in each subsequent layer can represent increasingly sophisticated aspects of the original input. Understanding how each layer changes the shape of the data as it flows through the network is a key aspect of truly understanding the mechanics of DL. There are many different types of layers, but one of the most common layers is the dense layer that connects all units in the layer directly to every unit in the previous layer.

The DNN architecture is forward in nature, i.e., the information does not shift between two consecutive layers, i.e., the layers give no feedback to the previous layers. A feed forward neural network (FFNN) is the most basic type of multi-layer NN, and as the name suggests, information is passed in the forward direction. Data flows from the input layer to the output layer without going backwards, and the links between the layers move one way, which is in the forward direction. FFNNs are the foundations of deep networks, such as CNN and RNN. Other architectures include LSTMs.

CNN is an FFNN that is generally used for image/object recognition and classification and for other complex classification problems, such as predictive maintenance. CNNs can extract the local features of the input data and combine them layer by layer to generate high-level features. As illustrated in Figure 1.26, a typical CNN has two phases. The first phase is a series of convolutions of layers, usually followed by pooling layers, while the second phase is a series of dense layers. CNNs can be used for deep learning with a few parameters; thus, there are fewer parameters to learn as compared to dense layers.

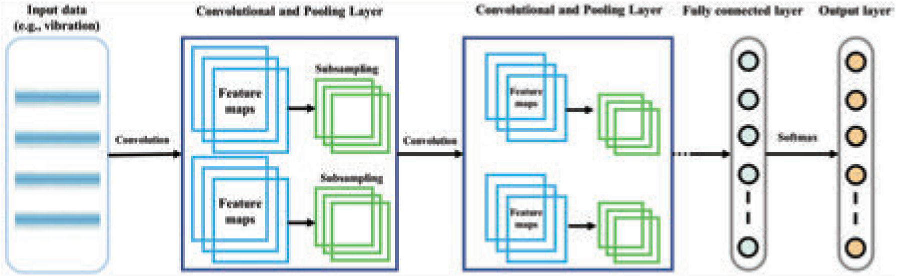


Figure 1.26 Typical CNN architecture.

RNN is the time-series version of an FFNN. It has connections between passes and through time. The connections form a directed graph along a sequence of features that link one layer to previous layers, allowing information to flow back into the previous parts of the network. Thus, each model in the layers depends on past events, allowing information to persist. The key idea behind RNN is to share parameters over time so that decisions can be made at each point in a sequence of events about what has happened so far in the sequence. In short, it ends up with a network that has a relatively simple repeating pattern, with part of the classifier connecting to the input at each time step and another part, called the recurrent connection, connecting you to the past at each step, as shown in the following Figure 1.27. On the downside, training RNNs can often be a challenging task due to their memory associated with the recurrent aspect (i.e. signals travel both forward and back and may contain loops, thus adding to their complexity).

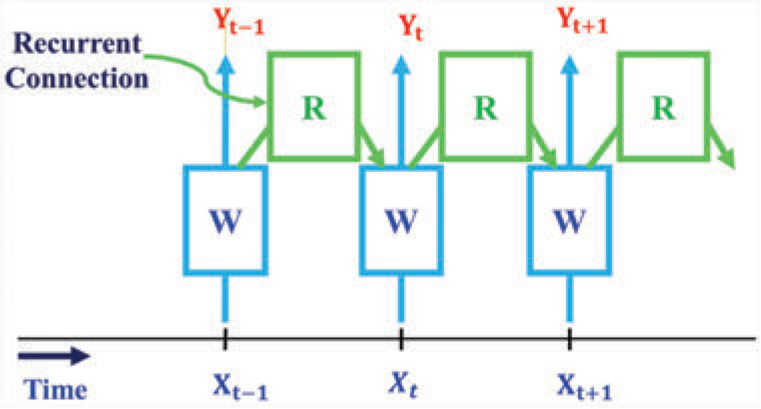


Figure 1.27 The repeating module underlying RNN architecture.

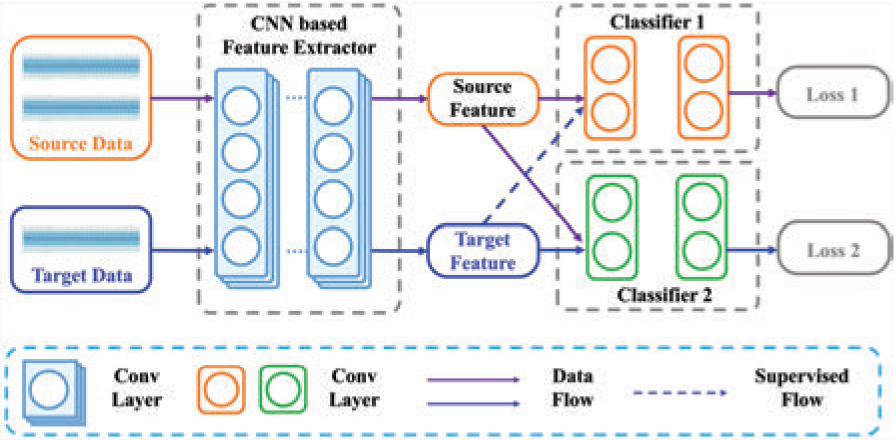


Figure 1.28 Example of an architecture useful for fault diagnosis. Adapted from [28].

LSTM is a type of RNN that, in addition to standards cells, also includes memory cells that can retain information for long periods of time. The enhanced architecture allows LSTMs to learn about long-term dependencies, which makes them smart at remembering things that have happened in the past and finding patterns across time.

Deep architectures are continuously evolving. Thus, the number of industrial applications in which DL is employed has grown steadily over the last decade. Many reported architectures have proven their superior ability in specific tasks, such as fault classification and fault prediction. An example of an architecture useful for fault diagnosis is shown in Figure 1.28.

It uses source domain-labelled data sets (such as vibration signals) to pre-train a CNN model, and a discriminator with two independent classifiers (fully connected layers) to optimize the CNN-based feature extractor parameters by minimizing distributions between the source and target domains.

1.8.3 Industrial Embedded AI/ML

Embedded AI is the application of AI at the embedded device level. While there are many examples of intelligent devices in the consumer space, embedded AI may have far higher potential in industrial applications. There are many contexts in which embedded AI may be very useful for collecting and understanding important phenomena in industrial settings, right where the sensors are located.

Embedded ML is the field of ML when applied to embedded systems such as microcontrollers. An embedded system is a combination of computer hardware and software, and additional parts, either mechanical or electronic, designed to perform a dedicated function.

The trend has been to connect the embedded devices via the Internet, collect the data and run the inference on servers in the cloud. However, according to the demand of the industry, the processing is moving from the cloud to the edge by using embedded ML, where lots of application can be designed having features of low cost, low power consumption, low bandwidth, secure and intelligent processing.

Embedded ML and DL techniques enable electronic systems to learn from real-time sensor data, audio and video and use the acquired knowledge to make standalone assessments, predictions, and decisions locally rather than in the cloud. Even more potential lies in combining real-time data from multiple sensors and thus deriving new types of information, leading to a continuous refinement and improvement of the ML/DL techniques. These techniques are applied on low power devices at the edge, hence the terms "edge ML and DL" are used interchangeably with "embedded ML and DL".

Edge AI refers to processing the data at the edge using AI methods and techniques, including ML and DL; however, edge AI has much more potential to accomplish edge intelligence than ML and DL alone. Edge AI equips sensor data with "the what" and "the how" to drive problem-solving processes, design, and development; hence, edge AI can be seen as the edge ML/DL of the future, encompassing architectures, frameworks, applications and edge intelligence and concepts, such as meta-learning and meta-intelligence.

The applications of embedded ML span many market segments and applications, for some of which the best pathways to development and deployment, such as time- and safety-critical applications, have yet to be found. The chapter seeks to cover a wide range of terms and concepts, not only with the aim of achieving a broader understanding of ML/DL applications but also to provide a valuable vantage point of where ML/DL are heading in the near future.

Many industrial applications target embedded ML and DL into edge devices, addressing the challenges and solving the problems posed by the gap between the advanced state-of-the-art models developed in and for the cloud and the limited capabilities of edge devices. The memory, processing, transmission and power consumption capabilities and limitations always depend on whether the device is micro-, deep-, or meta-edge device, but the challenges are the same. The AI/ML model needs to be converted into an efficient format, before compiling and flashing it into the device.

Benchmarking experiments are needed to demonstrate that state-of-the-art models with the right design and optimisation are compatible with the stringent resource requirements of edge devices and to suggest areas of improvement for the AI models.

Edge devices are typically single- and multi-core microcontrollers, with varying capabilities and limitations and unique identities. The edge can scale from a few devices to tens of thousands of devices distributed in different locations, so the devices are able to operate independently, with an unexplored evolution to training and inference actions. Although physically separated, the edge devices can be connected using wireless/wired connections in topologies such as mesh, with an unexplored potential for communication and distributed learning across devices inspired by recent advances in emergent intelligence.

ML model architectures can allow for highly interactive flows, starting with capturing the data straight from the embedded device all the way to production and deployment. This entails gathering sensor data directly from the products and environments and turning that data into useful data sets to be applied to ML algorithms and signal processing, instead of relying on predefined data sets. Furthermore, interactivity involves the verification, validation, and testing (VV&T) of algorithms, so that the most optimal solution given the device's capabilities and limitations is finally deployed.

The data, hardware/software platforms and more are the ingredients to design vertically integrated AI stacks, ensuring that edge AI is optimised for its hardware and its target application with optimised performance and efficiency.

The inference is performed on static models implemented on edge devices or other types of devices depending on the application. The inference requires many mathematical operations such as matrix multiplications and dot product operations and the processing run on a CPUs, GPUs, FPGAs, DSPs, ASICs depending on the processing power, energy efficiency, speed, and memory requirements.

Edge inference requires optimised hardware acceleration and when the process is connected to other performance-critical functions there is a need to provide interfaces by tightly coupling other accelerators or processing units into a common dynamic architecture.

1.8.4 On-device ML Applications Enabling True Edge Computing

The typical ML workflow takes advantage of several tools and frameworks, such as TensorFlow, TensorFlow Lite, and PyTorch, as shown in Figure 1.29.

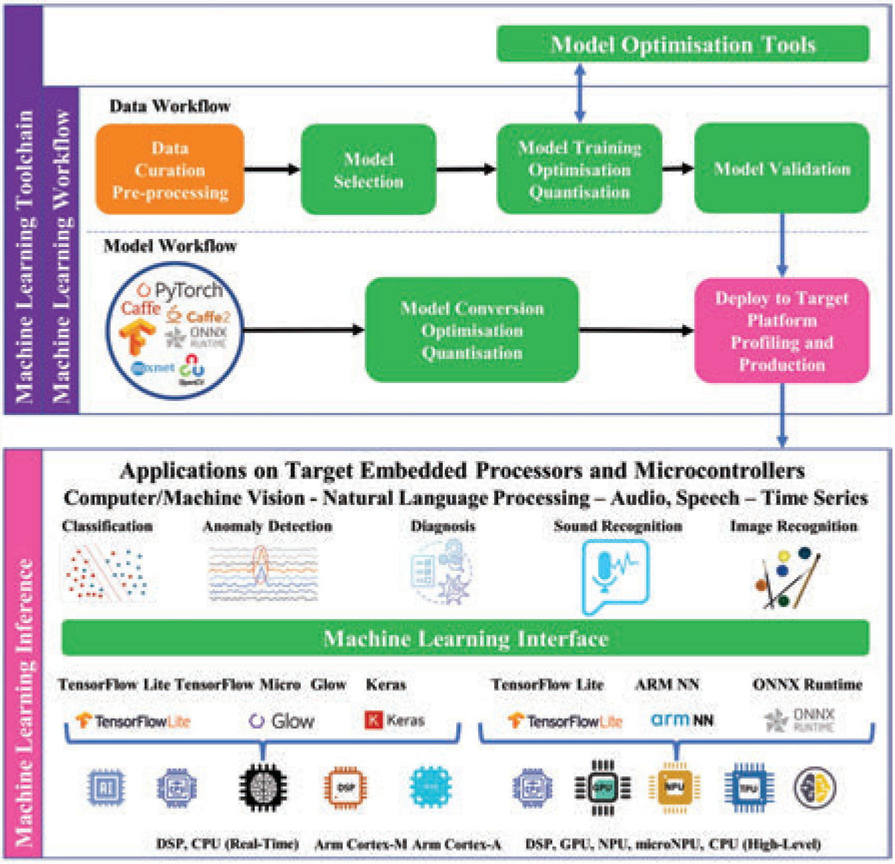

Machine Learning Toolchain
Machine Learning Workflow
Model Optimisation Tools
Data Workflow
Data Curation Pre-processing
Model Selection
Model Training Optimisation Quantisation
Model Validation
Model Workflow
PyTorch
Caffe
Caffe2
ONNX RUNTIME
mxnet
OpenCV
Model Conversion Optimisation Quantisation
Deploy to Target Platform Profiling and Production
Machine Learning Inference
Applications on Target Embedded Processors and Microcontrollers
Computer/Machine Vision - Natural Language Processing – Audio, Speech – Time Series
Classification
Anomaly Detection
Diagnosis
Sound Recognition
Image Recognition
Machine Learning Interface
TensorFlow Lite TensorFlow Micro Glow Keras
TensorFlow Lite
ARM NN
ONNX Runtime
TensorFlowLite
Glow
Keras
TensorFlowLite
arm NN
ONNX RUNTIME
DSP
GPU
NPU
TPU
DSP, CPU (Real-Time) Arm Cortex-M Arm Cortex-A
DSP, GPU, NPU, microNPU, CPU (High-Level)

Figure 1.29 Embedded ML design and development ecosystem view. Adapted from [40].

Some of them are optimised to run in very small footprints of memory and processing cycles, and thus can be employed in industrial embedded systems at the edge.

Most industrial embedded systems can be loosely classified into three main categories:

- *Vibration and motion* include industrial systems with sensors that allow not only for the control of the device but also for its predictive maintenance.
- *Voice and sound* include industrial systems with microphones for voice keyword detection and speech recognition.

- *Vision* includes industrial systems recognizing objects to sort them or spot defects, or systems identifying, for example, faces to unlock devices.

The problems that may arise in industrial embedded systems worth investigating to solve with the help of ML are many and multi-folded, but three general main categories can be identified:

- Detecting anomalies in the operation of edge devices before something breaks because industrial equipment can be expensive to produce and even costlier to repair or replace.
- Classifying things, behaviour, or objects from any variety and combination of sensors, either internal or external to the edge device.
- Forecasting, such as what the signal will look like in the near or far future, based on historical data.

All the potential use cases will have different workload performance and scalability requirements, depending on the application:

- For the prediction and maintenance of machines, it is essential to predict and give feedback on their health status as early as possible to avoid instant shutdown.
- For security systems, it is essential to implement features such as facial and voice recognition on edge devices to ensure they effectively contribute to providing security, through their use with security locks for home, offices, vehicles, and so forth.
- For autonomous vehicles, it is important that the devices installed on the car analyse local surroundings to recognize traffic lights, pedestrian roads, and people to make smart decisions.
- For surveillance and monitoring, it is crucial that any suspicious activities are monitored on edge devices and in real time by, for instance, recognizing human movements.
- For robots and robotic things, it is essential to make decisions independently without the need to connect to the internet.

While ML can be used to arrive at innovative solutions, it is important to note that embedding AI on the edge has limitations and that ML alone cannot always solve complex problems. Many industrial applications require other technologies to work in tandem with ML to achieve effective, low-power solutions to be deployed close to the sensor, thus enabling true edge computing.

More in-depth insights into use cases implementing industrial AI applications at the edge and the transition to Industry 5.0 can be found in [36].

1.8.5 Machine Learning on Embedded Devices

Most AI frameworks have been developed for desktops, servers, and laptops with large resources. By contrast, embedded edge AI frameworks run on smaller but efficient devices, such as single-board computers and microcontrollers. Single-board computers usually have a powerful microprocessor with a separate memory, can run a full operating system, and can provide a full-user interface; hence, they can adapt ML algorithms (such as Scikit-learn, TensorFlow, PyTorch, Keras, and Caffe) that use high-level programming languages such as Python, provided that they have enough power to fulfil the task effectively and efficiently.

The situation is rather different for microcontrollers, which are usually less expensive and require much less power, with only a few buttons or a simple LCD screen of the user interface. Hence, the adaption of the existing AI frameworks to run on microcontrollers has started to show results only recently.

Software Platforms

TensorFlow Lite was the first AI software framework specifically designed for micro controllers that allows running simple NNs without manually programming the matrix operations and with only a few kilobytes of memory. Since it was introduced, many AI software tools have been developed to address the different requirements for designing and implementing ML on edge devices. However, it was the optimisation of both hardware and software in tandem that allowed for the use of more complex ML algorithms in microcontrollers, which led to industries embracing the application of embedded ML.

Optimisation can be multi fold: enable more complex models to be deployed, meet real time latency constraints, extend the battery life of edge devices. The important point is that even the smallest optimisation anywhere in the system can make a difference, be it in hardware, software algorithms, framework, libraries, as shown in Figure 1.30.

Hardware Platforms and Hardware-software Co-design for ML

Embedded edge AI can be defined from the perspective of both hardware and software, depending on whose capabilities are focused on. From the

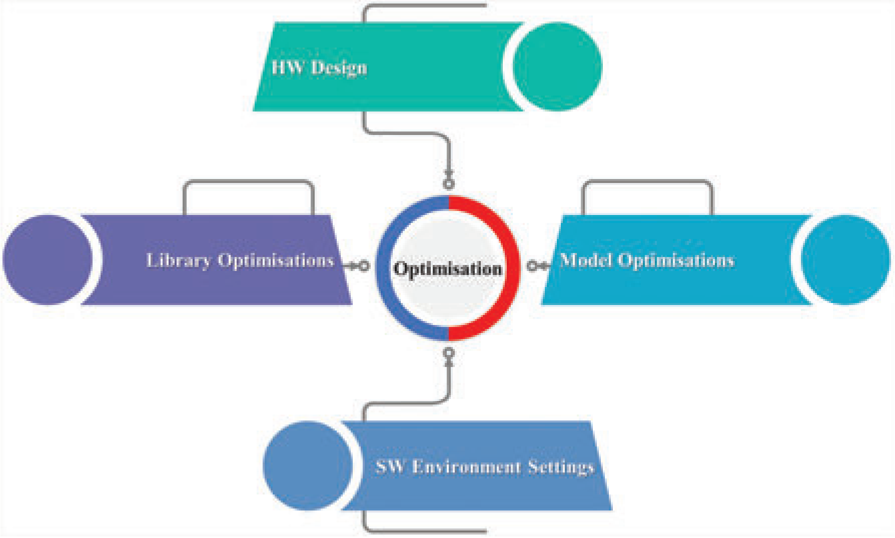


Figure 1.30 Embedded ML optimisation.

hardware perspective, embedded edge AI is defined as the capability of low-power, resource-constrained devices such as sensors and actuators to execute AI algorithms. From the software perspective, embedded edge AI is defined as the capability of AI algorithms to adapt and run effectively and efficiently on devices with limited resources.

The ability to embed AI in low-end devices is highly dependent on the availability of automated frameworks with easy-to-use design flows that can generate optimised AI models for the hardware targets. Thus, all hardware components (microcontroller, communication- modules, sensors, actuators, etc.) are part of the design flow. Hence, regardless of whether embedded edge AI is defined from the hardware or software perspective, a hardware-software co-design is key to embedding AI in edge devices.

Embedded edge ML has changed the way microprocessors and microcontrollers are used. AI can be embedded by augmenting development boards with components such as sensors and additional chips, all geared towards executing AI programs spanning from simple ML algorithms to resource intensive DNNs.

Until recently, most of the chips developed only supported a subset of functions used in modern DNNs, imposed by the memory restrictions and computing capabilities of the hardware; not even specialised hardware could execute DNNs.

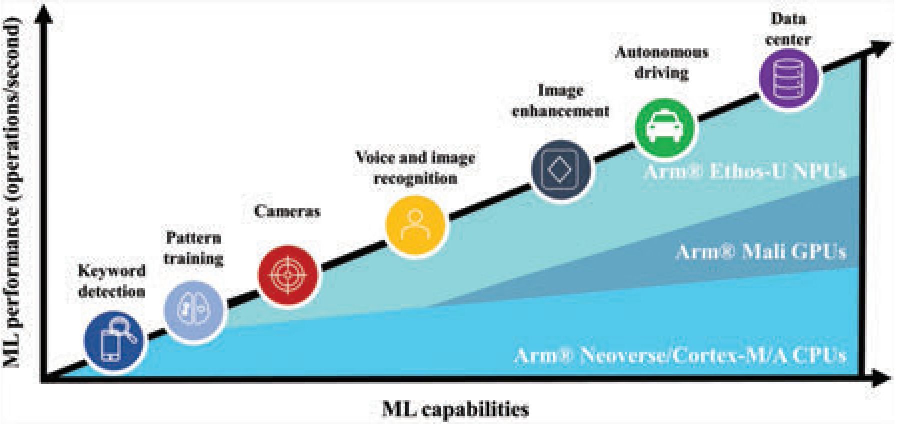


Figure 1.31 ML hardware options for various AI tasks. Adapted from [39].

With the recent advances in hardware, developments have been directed towards integrating AI and DNNs directly into sensor hardware. NNs targeting constrained devices are more efficient in terms of memory footprint and inference time. Techniques such as quantisation are used to reduce computing precision with no significant decrease in algorithm accuracy.

When designing hardware, special attention must be paid to the three main classes of AI-related building blocks, namely memory, storage, and logic. Memory is used for short-term storage during processing and consists of dynamic random-access memory (DRAM). Storage represents the long-term repository of large electronic data sets and consists of NAND flash memory. Logic is used for processing, computing, and optimising the calculation of NN operations or other specific AI functions and consisting of CPUs, GPUs, FPGAs, different custom ASICs, etc.

The edge processing units under development must have several characteristics such as a heterogeneous computing architecture (e.g., CPU, GPU, ASIC, FPGA, neuromorphic, etc.), support for the main AI edge frameworks (e.g., TensorFlow, Caffe, Keras, etc.), multi-modality, end-to-end embedded security, and high energy efficiency.

Accelerators and Neuromorphic Hardware

Accelerators and neuromorphic hardware are both represented as sub layers of the hardware layer, which is at the foundation of the technology stack. Employing both generic and hardware-specific optimisations can lead to a significant decrease in the memory footprint of NNs and accelerate inference latency.

Hardware accelerators are specialised hardware components within the system that enable greater efficiency when running certain computing tasks than is possible with software running on a general-purpose CPU alone. A wide variety of dedicated hardware acceleration systems exist, and the most common hardware used for acceleration include GPU, ASIC, FPGA.

Neuromorphic computing is a new computing technology that reproduces human brain activity with models of selective spiking ensembles of neurons in models that reproduce biological reactions.

Neuromorphic computers, as opposed to Von Neumann computers, which are composed of separate CPUs and memory units, are inspired by the human brain and are composed of neurons and synapses governing both processing and memory. Programs in neuromorphic processing units are determined by the structure of the neural network and its parameters instead of explicit instructions, as in a von Neumann computer. Neuromorphic computers receive spikes as input that can be used to encode numerical information continuously, as opposed to Von Neumann computers that encode numerical values represented by binary values [34]. This is intuitively illustrated in Figure 1.32.

Consequently, neuromorphic computers present some essential operational differences: they are highly parallel, meaning that, in principle, all neurons and synapses can operate simultaneously. Both neurons and synapses perform processing and store values, resulting in no separation between processing and memory. In addition, increasing the number of neurons and synapses can be done easily; thus, neuromorphic computers are highly scalable. Neurons and synapses ‘spike’ only when there are spikes to process, making them “event-driven”.

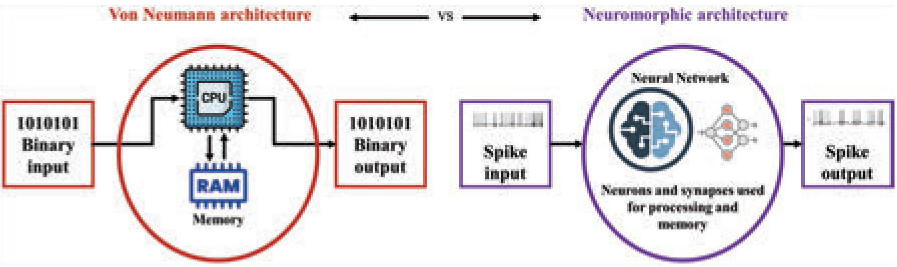


Figure 1.32 Comparison of the von Neumann architecture with the neuromorphic architecture.

Most of the work in neuromorphic computing has focused on hardware development. A neuromorphic chip can contain thousands of neurons, with their synapses, dendrites and axons reproducing human brain activity. However, neuromorphic computing requires both hardware and software, and to be widely adopted by industry in the future, neuromorphic algorithms and applications must catch up with technological advances in hardware. Spiking Neural Networks (SNN), which mimics the energy-efficient signal system in the brain, has drawn much recent attention. The main difference between SNNs and traditional networks is that neurons in SNNs accumulate charge from the environment or from other neurons over time; thus, time is a new element in their operation. Algorithms that have been successful for DL applications will need to be adapted to work on SNNs [34].

1.8.6 Embedded ML Development Flow in Industrial Setting

It is important to emphasise that the embedded edge ML flow and its associated processes are different from most typical ML flows. Many applications deal with static ML flows. A ML flow is static when there are no time variables in the equation. Hence, the static model is trained offline exactly once, and then the trained model is used for inference for some time, at least until an update is required. Moreover, many pre-built data sets are available for various domains and applications that ML practitioners can use as a start.

By contrast, most industrial applications must cope with time series problems and thus deal with data continuously entering the system over time. Pre-built data sets are not configured for use with smaller ML applications such as those intended for microcontrollers. In edge embedded systems, data are not extracted from data stores such as files or databases but rather are acquired directly from sensors. Thus, inference occurs in real time, and in many cases, so does training. The timeline can be short (seconds or minutes) or long (days or months). Owing to the dynamic aspect, re-training is necessary.

Figure 1.33 illustrates a typical embedded ML development flow. In short, the flow starts with the collection of signals. Continuous raw data are sliced into smaller windows and processed into extract features. The trained model is then deployed on the IIoT device and used to run inferences, whose result, depending on the application, can be a prediction, a class detected, or an anomaly detected. Pre-processing steps such as cleaning or filtering data may be necessary to obtain a representative data set for the application and make it easier to process.

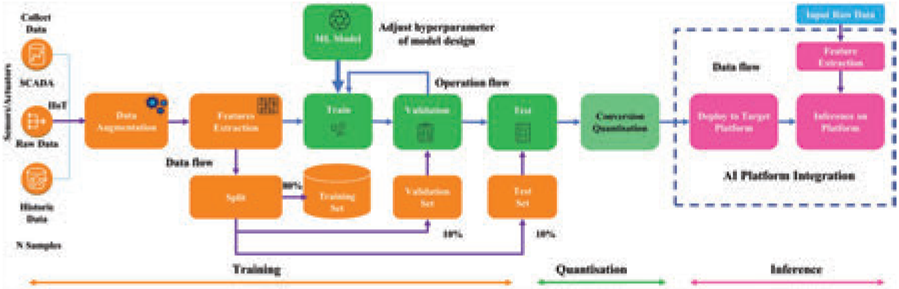


Figure 1.33 The high-level embedded ML development flow.

In the following paragraphs the basic steps of the embedded ML design flow are described, with examples from a generic use case, i.e., classification of the state of a motor based on the vibration measurements using an accelerometer sensor from an IIoT device. The motor is operating at fixed speeds, which are divided into several classes based on various percentages of the maximum speed.

The **data collection** process is essential, as good results are dependent of qualified data for the training and can require considerable effort and expertise to design the correct signal acquisition and sampling methodology suitable for a particular application.

The signals for each of the classes can be acquired straight from the device. The continuous raw data are usually sliced into smaller windows whose size can be configured with parameters. From a three-axis accelerometer sensor and with a buffer size of 256 samples on each axis, a total of 1536 values are produced per signal. With a sampling frequency of 1667 Hz, each buffer represents a snapshot of approximately 300 milliseconds of the accelerometer temporal vibration data. The number of signals and the split between training and validation data can also be configured (usually 80% training, 20%).

The vibration signals collected can be visualised as shown in Figure 1.34, in both temporal and frequency plots for each of the classes.

One common **pre-processing** technique when examining vibration or motion data to identify features is to take the Fourier transform of the data to obtain information about them in the frequency domain and break the signal into its various frequency components. By providing filtering, only the frequencies that represent the characteristics of the motor vibration are kept, and the rest are attenuated.

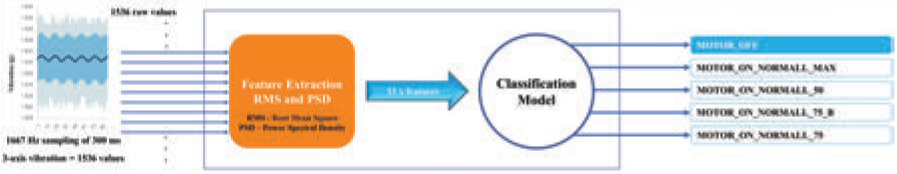


Figure 1.34 Temporal and frequency plots as input to motor classification.

A feature is an individual measurable property or characteristic of a phenomenon being observed and deciding what features to select is an important task. Poor features will have negative impacts. For example, if the feature only takes one snapshot in time, it is a poor feature because it provides no information about how the signal changes in time.

Extracting the features to be fed into the AI model for training and, ultimately, inference from large inputs can be performed automatically by many AI frameworks. In a matter of seconds or minutes, all raw samples are converted into sets of features.

A useful aspect of this automation is the possibility to visualise and explore the features. In the case of a classifier, features that are visually clustered are a good sign that the model can be trained to do the same. On the contrary, if features overlap in various degrees and are intertwined, it is very likely the trained model will have difficulties in differentiating between classes. This problem can be solved in various ways such as increasing the buffer size, that is, prolonging the sampling signal, to better capture signal patterns or even changing some of the features.

The **training** process employs back-propagation algorithms to configure and update the parameters inside the model that can improve the chances of predicting each feature set. Parameters are usually configured automatically.

In contrast to the model parameters, hyper-parameters cannot be tuned by the data and lie outside the model (Figure 1.35). These are values that must be set manually, such as the size and shape of the model, the learning rate, and the number of training steps to take, the features to use, and the methods and calculations to pre-process the data.

The **model validation** data sets and test data sets are not part of the training data sets. The validation data set can be used to analyse how well the model performs against unseen data and to adjust identified problems prior to using the test set (Figure 1.36).

Two common issues in ML are when the model underfits or overfits the input data. The former is when the model performs poorly on training and

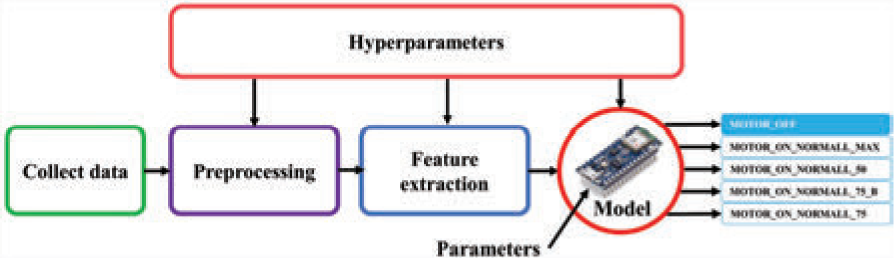


Figure 1.35 Hyperparameters (outside the model) vs parameters (inside the model).

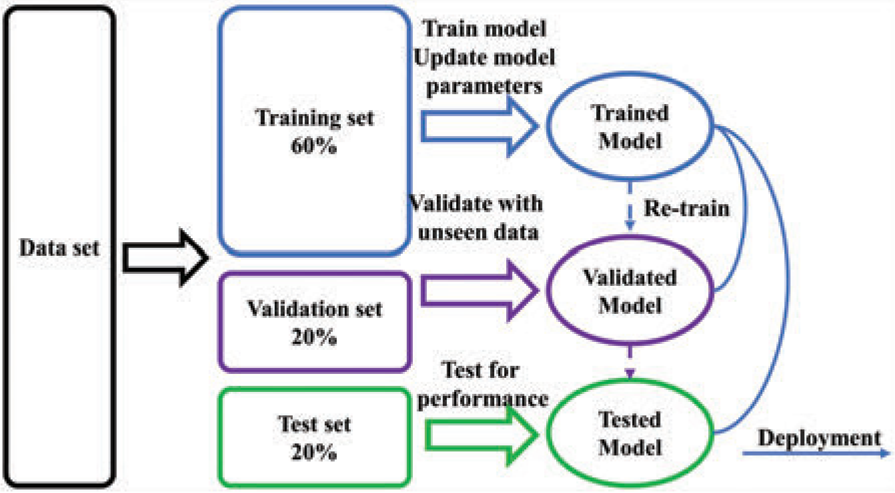


Figure 1.36 Categories of data sets and where they are used.

validation data, whereas the latter is when the model performs better on the training data than it does with the validation or test data.

The solutions to these issues present particularities in the case of embedded edge ML, but, in short, collecting more signals, selecting different features, extending the training time and increasing model complexity will usually work for underfitting, while gathering more data, training for a shorter length of time and reducing model complexity and adding dropout layers will work for overfitting.

Understanding NN architecture is essential to explore how increasing or reducing model complexity affects model accuracy. A neural network architecture can be optimised by several means (adding more layers to deeper the model or increasing the number of hidden units to wider the model, changing the activation, and optimization functions, learning rate, fitting more data, and more), and knowing what and how to optimise it is a matter of experimentation. Fortunately, most platforms can automatically tune hyperparameters.

Much of creating a better model is trial and error: gathering more data or adjusting the hyperparameters and re-training your model to see if it improves the per-class accuracy. Or sometimes, there may not be enough or the right kind of data to train a good model.

One of the most useful evaluation tools is the confusion matrix of the validation data (Figure 1.37). The predicted labels are on the x-axis and the true labels on the y-axis. The diagonal elements are the number of points for which the predicted label is equal to the true label, while off-diagonal elements are those that are mislabelled. The higher the diagonal values of the confusion matrix the better. The matrix is a good way to visually interpret

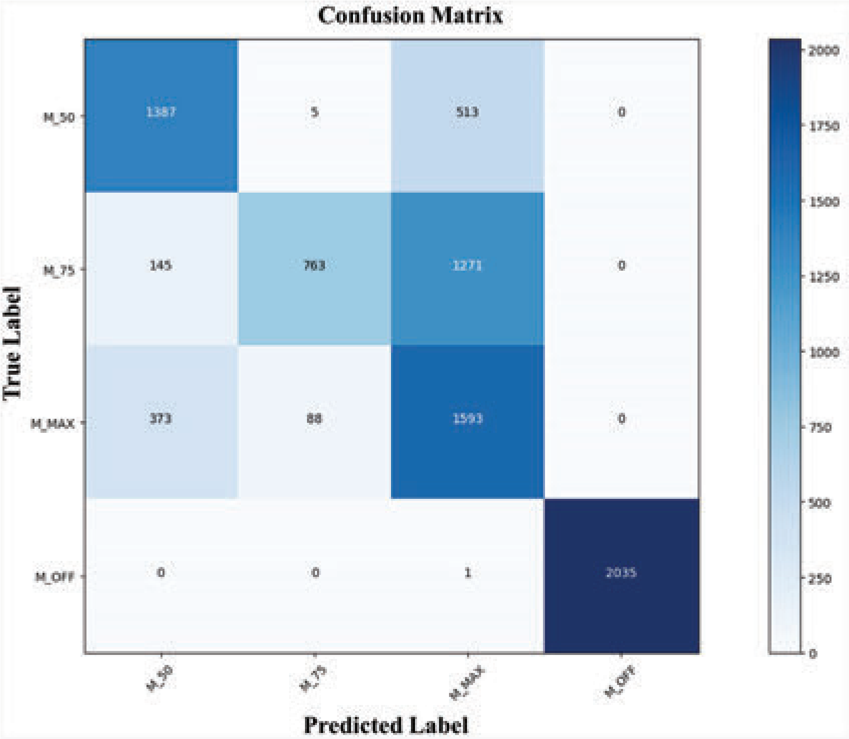


Figure 1.37 Confusion matrix.

how well the model is doing at its predictions and understand where it may need improvement.

In the context of micro-edge embedded systems, the **deployment** is dependent on the hardware/software platform and is more or less automated, and in essence comprises of three steps: the first is a format conversion of the fully trained model, then a weight/model compression to reduce the amount of memory to store the weights in the target hardware platform and to simplify the computation so it can run efficiently on target processors. This step is usually to quantize, i.e., converts all parameters from floating point values to integers. Finally, the last step is compiling the model and generating the code to be integrated with the MCUs firmware. The implementation of these steps must follow the back-end flow specific to the target. The optimisation challenge is to save as much memory as possible in the processor or microcontroller, with as little reduction in the accuracy of the model as possible.

Inference is the process of using live unseen data with a fully trained model to make predictions. The inference and the output will look different depending on the actual target device and production environment, but in essence, it happens in three steps.

First, the input signal is sampled for a period of time sufficient to capture the essence of the signal patterns before sending the raw accelerometer data to the library for inference. With 1667 Hz sampling rate and 300 milliseconds time length, the buffer size will be 256 samples producing in total 1536 values. The library expects these values to be stored in an array containing raw sensor values. Next, features are extracted, and finally, inference is performed, with the inference function returning the predicted probabilities, each corresponding to one of the classes. The highest probability will indicate the correct class, but threshold comparison and other algorithms can be used. A minimum threshold can also be considered. This process loops indefinitely. The state machine usually consists of two states with two functions "init" and "inferencing", respectively, with the former initializing the NN model and the latter being a continuously running function for collecting raw data from the sensors on board and making predictions in real-time. While feature extraction and inference are performed, the buffer fills up with raw sensor data in the background. More about applications that benefit from inference at the edge can be found in [10].

To conclude the discussion on the Hardware/Software technology stack, machine learning and neural networks can now be efficiently deployed on resource-constrained devices, which allow for cost-efficient deployment, widespread availability, and the preservation of sensitive data. However, the trade-offs that optimisation methods, software frameworks and hardware architecture have on key performance metrics, such as inference latency and energy consumption, have yet to be studied in depth.

1.9 Summary

Industrial AI and IoT/IIoT are enablers for building the foundations for digital transformation and business innovation. Full-scale and full-stack industrial AI technology accelerates digital innovation across industries and therefore boosts productivity. The adoption of AI helps industries climb the value chain and drive innovation, thus providing new paths to growth for manufacturing, service, and other industries.

In this context, managing the end-to-end (E2E) AI technologies connected with the IoT/IIoT, SCADA, and edge computing, is crucial for various industrial sectors. Addressing the developments in silicon-born AI that enable and generate AI-born embedded and industrial systems accelerates harnessing the silicon and embedded systems designed specifically for AI, thus supporting E2E solutions and advancing the adoption of AI technologies across industrial sectors.

Contributions to this chapter come from a diverse number of disciplines and communities and cover related technologies across different layers in the AI technology stack.

As result, the chapter provides an overview of the main concepts and terminology related to industrial embedded edge AI technologies.

The shifting of AI methodologies from operating in the cloud to operating at the edge as a fundamental approach for future developments on digitising industries marks the beginning of a widespread transition in the control of industrial processes and the functionality of devices. AI methodologies operating on the edge must drive the major milestones of this transition on any roadmap.

Embedded edge AI platforms, training and learning, and applications form the foundation that supports the development of edge AI applications.

AI-optimised hardware provides the core infrastructure for embedded edge AI applications. It includes AI chips (neuromorphic, CPUs, GPUs, FPGAs, ASICs), large-capacity, low-latency, and all-flash arrays, and solid-state storage devices; high-performance, high-throughput, and highly scalable edge servers and network equipment. Turning data into descriptive, diagnostic, and predictive analytic insights requires visualised modelling and code testing environments, as well as ML and DL edge platforms configured for general AI applications or real-time embedded environments.

Edge AI technologies and applications require advanced industrial enterprise high-level architecture as a reference for implementing embedded edge AI technologies in an environment that can manage the large-scale deployment of AI applications.

The infrastructure layer requires edge computing and modular processing units integrated with on-premises platforms. In the industrial platforms and application layers, the analytics and flexible service capabilities of edge must support the integration of industrial enterprise applications with various industrial AI applications.

As AI matures, AI technological development often intersects other technological areas.

The chapter introduced an overview of AI concepts, including definitions to establish a common vocabulary for the stakeholders involved and for the presentation of E2E industrial embedded edge AI technologies across the technology stack, application, and industrial sectors. The chapter can thus serve as a reference for various partners and stakeholders to help reach the full potential of edge AI for digitising industry by introducing developments in silicon-born AI to enable and generate AI-born embedded and industrial systems and accelerate the adoption of edge AI technologies across various industrial sectors.

Industrial edge AI technologies differ from consumer AI technologies that provide citizens with direct technology exposure, so industrial AI solutions may lack direct consumer scrutiny. Nevertheless, societal perception has an impact on how unions perceive the introduction of edge AI technologies, how management decisions on investment are made and how policymakers decide upon regulations.

Acknowledgements

This work is conducted under the framework of the ECSEL AI4DI "Artificial Intelligence for Digitising Industry" project. The project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826060. The JU receives support from the European Union's Horizon 2020 research and innovation programme and Germany, Austria, Czech Republic, Italy, Latvia, Belgium, Lithuania, France, Greece, Finland, Norway.

References

[1] Research and Markets. “Smart Manufacturing Market by Technology (Robotics, AI, IIoT, Cloud, AR/VR), Application (Machine Inspection; Energy, Quality, and Warehouse Management; Planning, Surveillance, Optimization), End-use Industry, and Geography - Global Forecast to 2029”, June 2022. Available online at: https://www.researchandmarkets.com/

[2] Vantage Market Research. “AI in Manufacturing Market Size, Share & Trends Analysis Report by Offering (Hardware, Software, Services), by Technology (Machine Learning, Natural Language Processing, Context-aware Computing, Computer Vision), by Application (Predictive Maintenance and Machinery Inspection, Material Movement, Production Planning, Field Services), by Industry (Automobile, Energy and Power, Pharmaceuticals, Heavy Metals and Machine Manufacturing), by Region (North America, Europe, Asia Pacific, Latin America and Middle East & Africa) - Global Industry Assessment (2016 - 2021) & Forecast (2022 - 2028)”. Available online at: https://www.globenewswire.com/

[3] AI4DI, Artificial Intelligence for Digitising Industry. Available online at: https://ai4di.eu/

[4] P. H. Winston. Artificial Intelligence. Third Edition, Addison-Wesley Publishing Company, 1992.

[5] D. B. Fogel, “Defining Artificial Intelligence”. In Evolutionary Computation: Toward a New Philosophy of Machine Intelligence. Third Edition, The Institute of Electrical and Electronics Engineers, Inc., IEEE Press, pp. 1-32, 2006.

[6] S. Legg, and M. Hutter, “Universal Intelligence: A Definition of Machine Intelligence. Minds and Machines”, 17(4):391-444, Springer, 2007. Available online at: https://arxiv.org/abs/0712.3329

[7] J. McCarthy, “What Is Artificial Intelligence, Basic Questions”, Stanford Formal Reasoning Group, 2007.

[8] S. J. Russell, and P. Norvig, Artificial Intelligence: A Modern Approach, Fourth Edition. Prentice Hall, 2022.

[9] J. R. Searle, Mind, language and society, New York, NY: Basic Books, ISBN 978-0-465-04521-1, 1999.

[10] What is AI Inference at the Edge? Available online at: https://www.steatite-embedded.co.uk/what-is-ai-inference-at-the-edge/

[11] O. Vermesan, J. Bacquet, (Editors). Next Generation Internet of Things - Distributed Intelligence at the Edge and Human Machine-to-Machine

Cooperation. ISBN: 978-87-7022-008-8 (Hardback), 978-887-7022-007-1 (Ebook). River Publishers, 2018.

[12] R. Schwartz, J. Dodge, N. A. Smith, O. Etzioni, (2019). “Green AI”. Available online at: https://arxiv.org/pdf/1907.10597.pdf

[13] Buchanan, B.G., Shortliffe, E.H. (eds.). Rule-Based Expert Systems - The MYCIN Experiments of the Stanford Heuristic Programming Project. Addison-Wesley Publishing Company (1984)

[14] J. McDermott. R1: an Expert in the Computer Systems Domain. Proceedings of the National Conference on Artificial Intelligence (AAAI), pp. 269-271 1980.

[15] R. Reiter. A logic for default reasoning. Artificial Intelligence 13(1–2), 1980.

[16] X. Yang, Z. Song, I. King and Z. Xu. “A Survey on Deep Semi-supervised Learning”. https://doi.org/10.48550/arXiv.2103.00550

[17] R. Reiter, (1987). A theory of diagnosis from first principles. Artificial Intelligence, 32(1), 57–95.

[18] J. De Kleer, A. K. Mackworth, and R. Reiter, (1992). Characterizing diagnosis and systems. Artificial Intelligence, 56.

[19] T. Eiter, G. Ianni, T. Krennwallner, “Answer Set Programming: A Primer”, pp. 40–110. Springer Berlin Heidelberg, Berlin, Heidelberg (2009). https://doi.org/10.1007/978-3-642-03754-2_2

[20] F. Wotawa, “Reasoning from first principles for self-adaptive and autonomous systems”. Springer (2019). https://doi.org/10.1007/978-3-030-05645-2

[21] A. Choi, R. Wang, A. Darwiche, “On the relative expressiveness of Bayesian and neural networks”. *Int. J. Approx*. Reason. 113: 303-323 (2019)

[22] W. Shi, A. Shih, A. Darwiche, A. Choi, “On Tractable Representations of Binary Neural Networks”. CoRR abs/2004.02082 (2020)

[23] D. Foster. Generative Deep Learning. (Kindle Locations 242-243). O’Reilly Media. Kindle Edition.

[24] S. Akcay, A. Atapour-Abarghouei, and T. P. Breckon, “Ganomaly: Semi-supervised anomaly detection via adversarial training,” in *Asian Conference on Computer Vision*, pp. 622–637. Springer, 2018

[25] Biomimicry 3.8. https://biomimicry.net/resource-handbook/

[26] A. M. Turing, (October 1950). “Computing machinery and intelligence”. Mind. LIX (238): 433–460. https://academic.oup.com/mind/article/LIX/236/433/986238?login=false

[27] E. B. Baum, D. Boneh and C. Garrett, “On genetic algorithms.” COLT ’95 (1995).

[28] Y. Ran, X. Zhou, P. Lin, Y. Wen and R. Deng, “A Survey of Predictive Maintenance: Systems, Purposes and Approaches”, *IEEE Communications Surveys and Tutorials*, Nov. 2019.

[29] M. Ghallab, D. Nau, and P. Traverso. Automated planning and acting. Cambridge University Press, 2016.

[30] R. K. Lindsay, B. G. Buchanan, E. A. Feigenbaum, and J. Lederberg. Applications of Artificial Intelligence for Organic Chemistry: The DENDRAL Project. New York: McGraw-Hill

[31] C. Sabo, K. Cohen, “Fuzzy logic unmanned air vehicle motion planning”, Advances in Fuzzy Systems Volume January 2012 Article No.: 13, pp 13. https://doi.org/10.1155/2012/989051

[32] M. Xu, W. C. Ng, W. Yang, B. Lim, J. Kang, Z. Xiong, D. Niyato, Q. Yang, X. S. Shen, C. Miao. (2022). A Full Dive into Realizing the Edge-enabled Metaverse: Visions, Enabling Technologies, and Challenges. https://doi.org/10.48550/arXiv.2203.05471

[33] M. Bojarski, B. Firner, B. Flepp, L. Jackel, U. Muller, K. Zieba and D. Del Testa, “End-to-End Deep Learning for Self-Driving Cars”. https://developer.nvidia.com/blog/deep-learning-self-driving-cars/

[34] C. D. Schuman, S. R. Kulkarni, M. Parsa, et al. “Opportunities for neuromorphic computing algorithms and applications”. *Nat Comput Sci 2*, 10–19 (2022). https://doi.org/10.1038/s43588-021-00184-y

[35] J. Popper, J. Hermann, K. Cui, et al. (2018). Artificial intelligence across industries - IEC Whitepaper.

[36] O. Vermesan, J. Reiner, C. De Luca, M. Coppola (Eds). Artificial Intelligence for Digitising Industry Applications. ISBN: 9788770226646, River Publishers, 2022.

[37] The German Artificial Intelligence (AI) Standardization Roadmap, 2020, https://www.din.de/resource/blob/772610/e96c34dd6b12900ea75b460538805349/normungsroadmap-en-data.pdf

[38] J. Redmon, S. Divvala, R. Girshick, A. Farhadi (2015). You Only Look Once: Unified, Real-Time Object Detection. https://doi.org/10.48550/arxiv.1506.02640

[39] S. Roddy (2019). Arm NN: the Easy Way to Deploy Edge ML. https://community.arm.com/arm-community-blogs/b/tools-software-ides-blog/posts/arm-nn-the-easy-way-to-deploy-edge-ml

[40] R. Martino, R. Oshana, N. Ekambaram, A. Osman Örs, and L. Pilati (2022). Edge Computing Intelligence. EE Times Asia. https://www.eetasia.com/edge-computing-intelligence/

2

Technology and Hardware for Neuromorphic Computing

Björn Debaillie, Ilja Ocket, and Peter Debacker

imec, Belgium

Abstract

Edge artificial intelligence and machine-learning algorithms increasingly enter our day-to-day products and applications. This massive adoption of data in all aspects of human activity will lead to unprecedented growth in computational needs to process this data into useful information and actions. The current approach to process this data in high-end cloud server parks is no longer sustainable as it costs energy, latency, and poses privacy threats. Realizing intelligent energy-efficient local processing is however extremely challenging. Neuromorphic computing, modelled according to the human's brain nerve network, is often suggested to realize such processing. Building such neuromorphic processing hardware however requires major advancements at different levels. New technology platforms for emerging semiconductor devices must be developed, levering emerging memory technologies which show characteristics related to neuromorphic computation. Magnetoresistive Random Access Memory (MRAM) could mimic the stochastic behavior of synapses, Ferroelectric Random Access Memory (FeRAM) could be tuned to emulate synaptic weight, and the temporal and analog qualities of biological neurons and synapses could be mimicked Resistive Random Access Memory's (RRAM's) memristors. We also present a 3D interconnection roadmap suitable to integrate neural accelerators. Related to neuromorphic hardware design and architectures, we optimize conventional neural network algorithms like Deep Learning (DL) and Spiking Neural Networks (SNNs) by focussing on their most critical parts in terms of power, performance, and area. All this will be leveraged in use case demonstrators for different

applications that need complex machine-learning algorithms in their mobile devices. All these activities are executed in the TEMPO project aiming to broaden the applicability of integrated neuromorphic hardware by means of technological innovation.

Keywords: Neuromorphic computing, edge processing, spiking neural networks, deep learning, hardware, silicon technologies

2.1 Mobile Devices Call for Efficient Neuromorphic Computing

Increasingly, edge artificial intelligence and machine-learning algorithms enter our day-to-day products and applications such as smart home assistants with natural-language processing, face-recognition-based security systems or autonomous vehicles. In the coming years, the demand for these increasingly complex computational algorithms will only grow further. At this moment, high-end server parks process the data in the cloud.

However, sending data to the cloud costs energy, latency, and is often not preferred for privacy reasons. As such, the ultimate edge artificial intelligence applications require intelligent energy-efficient local processing.

Realizing such intelligent energy-efficient local processing is however extremely challenging. Neuromorphic computing which is modelled according to the sophisticated nerve network of our human brain is often suggested as key technology to realize such processing. The project ECSEL JU TEMPO (Technology and hardware for neuromorphic computing) [1] aims to progress towards such processing. TEMPO collaboratively develops technology and hardware platforms leveraging emerging memory technologies for neuromorphic computing. The goal is to develop a new way to support a diversity of applications in mobile devices that need complex machine- learning algorithms.

2.2 Neuromorphic Hardware Enables Next Generation AI

Neuromorphic engineering is a ground-breaking approach to the design of computing technology that draws inspiration from the powerful and efficient biological neural processing systems. Neuromorphic devices can carry out sensing, processing, and motor control strategies with ultra-low power performance. Today's neuromorphic community in Europe is leading the State-of-the-Art in this domain. The community counts an increasing number of labs that work on theory, modelling, and implementation of neuromorphic computing systems using both conventional very large-scale integration (VLSI) technologies, emerging memristive devices, photonics, spin-based, and other nano-technological solutions. To enable the uptake of this technology and to match the needs of real-world applications in future products that solve real-world tasks in industry, healthcare, assistive systems, and consumer devices, extensive work is needed in terms of neuromorphic algorithms, emerging technologies, hardware design and neuromorphic applications respectively.

In the TEMPO project, we consider "neuromorphic" as brain-inspired algorithms, and we focus specifically on conventional DL and SNNs. That way, it is ensured that both established paradigms are covered in the greater domain of brain-inspired computation. Given the slowdown of silicon-only scaling, it is important to extend the roadmap of neuromorphic implementations by leveraging fitting technology innovations. Along these lines, TEMPO sweep technology options, covering emerging memories and 3D integration, and attempt to pair them with contemporary DL and exploratory (SNN) neuromorphic computing paradigms.

Terms like Artificial Intelligence (AI) and Machine Learning (ML) enjoy a popularity trend that is fuelled by a wide variety of applications. They come in a wide variety of underlying algorithms. Regardless of the algorithm, the goal of TEMPO is to implement accurate classifiers and/or predictors of raw data that is either available in a pre-stored location or entering as a stream (images, audio, video, etc). The local deployment of these algorithms, exactly near the generation of raw data, is identified as one of the main progress directions of the overall AI/ML trend [2], which assists the already growing ecosystem that develops and applies neuromorphic algorithms on an increasing number of end-user applications [3]. This observation is echoed additionally by the increasing percentage of custom chips that are designed, which follow the growing AI/ML trend and execute a wide variety of neuromorphic algorithms [4].

To address this, TEMPO aims to **broaden the applicability of integrated neuromorphic hardware** by improving energy efficiency with emerging memory technologies in novel neuromorphic hardware implementations, and to **develop technology platforms** for emerging semiconductor devices and **demonstrate** them for the **energy efficient hardware implementation of neuromorphic workloads**. To achieve this, TEMPO spreads over three action areas as illustrated in Figure 2.1. These action areas cover (1) the definition and the enablement to develop the emerging technologies, (2) the architectural definition and the related neuromorphic hardware design,

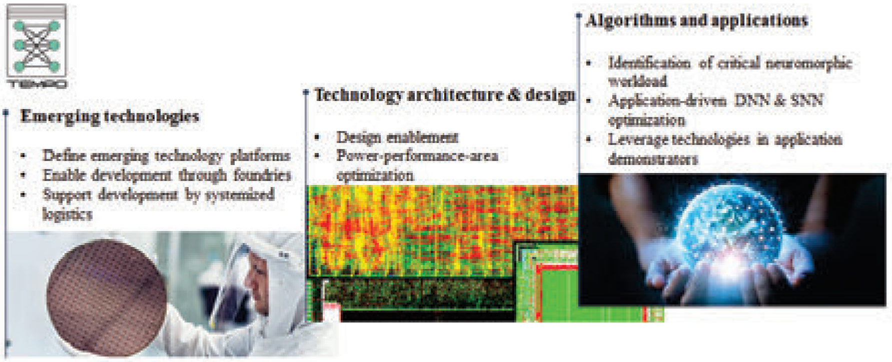

TEMPO
Emerging technologies
• Define emerging technology platforms
• Enable development through foundries
• Support development by systemized logistics
Technology architecture & design
• Design enablement
• Power-performance-area optimization
Algorithms and applications
• Identification of critical neuromorphic workload
• Application-driven DNN & SNN optimization
• Leverage technologies in application demonstrators

Figure 2.1 TEMPO spreads over three action areas.

and (3) the neuromorphic algorithm design and leverage the neuromorphic technologies for future applications in mobile devices that need complex machine-learning algorithms.

2.3 Building Neuromorphic Hardware

Neuromorphic hardware is the key to sustain the ability of mobile devices to deal with complex machine-learning algorithms. Building such neuromorphic solutions, however, comes with many diverse challenges. These challenges can only be tackled through synergetic collaborations across the entire neuromorphic technology value chain covering major foundries, chip design, system houses, application companies and research partners. TEMPO acts as the umbrella to enable such synergetic activities to address the following objectives:

- **Enable the joint development** of participating European Research and Technology Organisations (RTOs), foundries and leading (application) companies towards the identification of emerging semiconductor technologies that fit best to neuromorphic hardware and address relevant applications indicated by participating end-user partner companies.
- **Evaluate current concepts** for the implementation of neuromorphic hardware according to Key Performance Indicators (KPIs) at the device, architecture and application level, like power consumption, silicon area/cost, latency, throughput, energy for a given application task, memory bottlenecks, manufacturing challenges, operating frameworks.
- **Extend the technology roadmap** that is driven by Integrated Circuits (ICs) designed specifically for AI and ML applications by evaluating

and demonstrating the applicability of emerging technologies that can provide scalable power, performance, and area benefits.

- **Broaden the applicability of neuromorphic hardware**, by designing energy efficient integrated neuromorphic implementations, by fabricating them in collaboration with European foundries and in European cleanrooms, and by benchmarking them in terms of power, performance, and area in the context of pervasive applications that are provided by the end-user partners of the TEMPO project.
- **Exchange wafers** (where applicable) between foundries and the participating RTOs to facilitate the demonstration of functional neuromorphic chips, combining concepts from different RTOs and technologies from industrial companies. This will enable the use of the extensive know-how of European RTOs for future products while maintaining contamination free high-volume manufacturing.
- **Quantify the capability of the most prevalent neuromorphic hardware implementations** by targeting a broad algorithmic spectrum and isolating the critical sections of each algorithm. This includes DL inference such as Convolutional Neural Network (CNNs) and SNNs. This wide coverage will result into a CNN - technology-, design-, and system-aware scorecard containing the most sought-after neuromorphic implementations and their coupling with emerging technologies and applications.
- **Complement existing research** and provide guidance for future directions in the domain of neuromorphic algorithms, design, and systems by assessing the suitability of emerging technologies. The comparative evaluation between implementations of different neuromorphic algorithms can provide guidance to European neuro- morphic research, placing each approach in the context of emerging technologies and relevant applications.
- **Enable the European industry** to remain at the **leading edge** of neuromorphic chip development.

More detailed approaches ang the three action areas defined in TEMPO and illustrated in Figure 1.3.1 are described in the next sections.

2.3.1 Approach to Realise the Emerging Technologies

The core technology component of the TEMPO project is the development of emerging technologies that can provide measurable efficiency benefits to neuromorphic hardware implementations. The objectives with respect to technology are to:

- **Align** the process practices of involved partners, so that base wafers can be optimally exchanged for the development of novel neuromorphic hardware. This includes both the transfer of wafers from the foundries to the involved RTOs and, where/when applicable, the transfer of wafers between the cleanrooms of the RTOs.
- **Match** emerging memory technologies with the proper neuromorphic algorithms, so that hardware integration of the former brings about power, performance, area, and cost benefits.
- **Adjust** process practices so that the integrated emerging memory modules are compatible with traditional semiconductor manufacturing practices.

2.3.2 Approach to Derive the Hardware Architectures and Designs

The core hardware component of the TEMPO project is the development of processing hardware technologies which are efficient to support future AI-intensive mobile applications. The objectives with respect to neuromorphic hardware are to:

- **Develop** novel **architectures** and sub-system designs that help to reduce the memory bottleneck and power consumption, allow for a minimization of required memory space, and minimize the occupied silicon area (i.e., chip cost) while maintaining target accuracy, latency, and throughput.
- **Extend** basic architectures of CNN or SNN arrays with a scalable global communication network to enable high throughput and high complexity applications.
- **Design** modules that use emerging memory technologies to implement the core workloads of the major neuromorphic algorithms.
- **Ensure** component- and system-level compatibility with traditional electronic design flows.
- **Estimate** the power, performance, area, and cost of emerging memory integration for neuromorphic algorithms at the system-on-chip level and compare against contemporary implementations.

2.3.3 Approach Related to Neuromorphic Algorithms and Applications

To put the TEMPO project into the general perspective of accelerated ML, it is fundamental to identify the exact workloads that will be targeted for efficient and low power hardware integration with advanced technologies. This is a major precondition, as it is of vital importance to optimally concentrate the effort of the project to the fundamental computational bottlenecks identified in the target neuromorphic algorithms. The algorithmic objectives of the TEMPO project are as follows:

- **Profile** target neuromorphic algorithms for computational/memory bottlenecks
- **Identify** the algorithm regions that warrant hardware support
- **Specify** the complexity of the integrated neuromorphic implementations

TEMPO aims to allow applications to make easy use of the new neuromorphic technologies. The objectives to enable this are:

- **Extend the range of applications** to domains requiring (ultra-)high throughput and high complexity such as high throughput imaging, autonomous vehicles, vision enabled robots.
- **Create a demonstration design flow and a tool flow** that connects the target neuromorphic algorithms with the target applications.
- **Prototype the design and tool flows** to illustrate real time characteristics of the target neuromorphic applications, before the emerging technology samples become available.
- **Demonstrate the feasibility** and efficacy of integrated neuromorphic kernels on state-of-the-art benchmarks with functional demonstrators that use or emulate the proposed neuromorphic building blocks.

2.4 Positioning Within the Neuromorphic Computing Landscape

Neuromorphic computing comes in many flavours and forms of maturity. Figure 2.2 gives a simplified but illustrative view of the greater landscape of neuromorphic computing. In terms of implementation, neuromorphic computing can rely in analog, digital or hybrid hardware technologies. In terms of algorithms, the spectrum can range between the compute-intensive deep learning algorithms towards event-based processing like spiking neural network algorithms. The production level maturity is indicatively illustrated in Figure 2.2. Digital processing units like CPU's and GPU's and readily available on the market and are used for compute-intensive tasks in server racks and in the cloud. Commercial solutions are, however, scarcer when considering more analog implementations and/or more transient-based processing. TEMPO covers the complete brain-inspired computation domain,

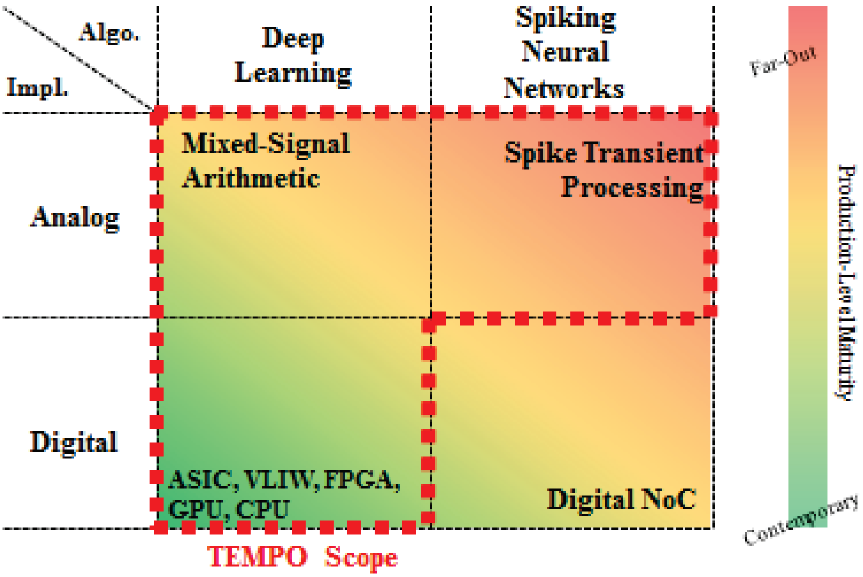


Figure 2.2 TEMPO positioned in the greater landscape of neuromorphic computing.

algorithmically ranging from DL inference engines to exploratory SNNs, and implementation-wise from standard digital to mixed-signal or analog implementations. The quadrant uncovered by TEMPO aims at massively parallel computer architectures. These architectures aim to mimic the implementation of human brains, which are composed of billions of simple computing elements, communicating using unreliable spikes.

The TEMPO project will existing evaluate memory technologies at device, architecture, and application level, and build and expand the technology roadmap for European AI hardware platforms. The project will leverage MRAM, FeRAM and RRAM memory to implement both SNN and Deep Neural Network (DNN) accelerators for 8 different use cases, ranging from consumer electronics to automotive, digital industry and medical applications.

MRAM is a type of memory that stores data magnetically but uses electrons to read and write it. The magnetic character provides non-volatility, which the electronics provides speed. A storage element is comprised of two ferromagnetic layers, consisting of a free layer and a pinned layer, sandwiching a non-magnetic oxide layer. It works by overcoming the resistance required to switch the magnetization from one direction to the other. Multiple resistance states can be achieved by incorporating domain walls in the free layer. The stochastic nature of switching states in these devices can be employed to mimic the stochastic behavior of synapses.

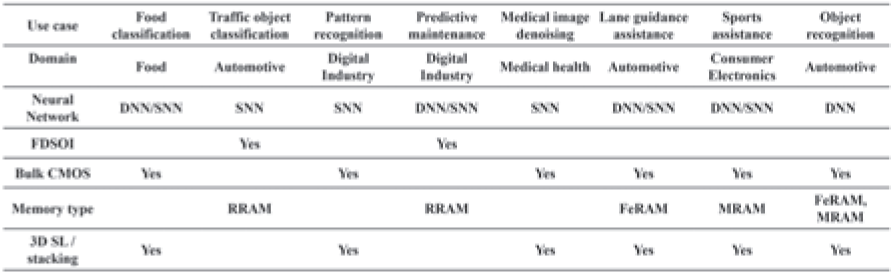
FeRAM memory uses ferroelectric materials that can switch rapidly between two polarized states. This type of memory offers high performance at low power, along with the added advantage of non- volatility. Ferroelectric Field-Effect Transistor (FeFET) can be tuned to emulate synaptic weight, an important element of neuromorphic computation. One big advantage of FeFET is that some ferroelectric compounds are also Complementary Metal-Oxide Semiconductor (CMOS)-compatible, making it easier to integrate into standard computing platforms. The downside is that the technology also suffers some of the limitations as Dynamic Random-Access Memory (DRAM), including scaling, leakage, and reliability.

RRAM is a form of nonvolatile storage that operates by changing the resistance of a specially formulated solid dielectric material. An RRAM device contains a whose resistance varies when different voltages are imposed across it. RRAM acts as an electronic switch that exhibits non- volatility, i.e., will retain its resistance state even after the voltage is turned off. The main advantages of this memory type are its scalability, CMOS compatibility, low power consumption, and analog conductance modulation. Its suitability for neuromorphic computing is related to the memristor's ability to change its state based on the history of voltages applied to it. As a result of this behaviour, it has the temporal and analog qualities of biological neurons and synapses. However, making these memristors more uniform so they will operate reliably is challenging.

2.5 Targeted Use Cases and Application Domains

The TEMPO project leverages its developed technologies over 8 different use cases over 5 application domains (automotive, food, digital industry, consumer electronics, and medical health). Table 2.1 gives an overview of the different use cases and the related neural network approach and technological choices. The different use cases are driven by the key industry partners within the consortium.

Table 2.1 Edge AI use cases addresses in TEMPO covers five application domains



Use case	Food classification	Traffic object classification	Pattern recognition	Predictive maintenance	Medical image denoising	Lane guidance assistance	Sports assistance	Object recognition
Domain	Food	Automotive	Digital Industry	Digital Industry	Medical health	Automotive	Consumer Electronics	Automotive
Neural Network	DNN/SNN	SNN	SNN	DNN/SNN	SNN	DNN/SNN	DNN/SNN	DNN
FDSOI		Yes		Yes				
Bulk CMOS	Yes		Yes		Yes	Yes	Yes	Yes
Memory type		RRAM		RRAM		FeRAM	MRAM	FeRAM, MRAM
3D SL / stacking	Yes		Yes		Yes	Yes	Yes	Yes

The following sections elaborate some of the envisioned use cases.

2.5.1 Food – Food Classification

This use case focusses on building a network and data pipeline for the classification of western food as illustrated in Figure 2.3. This activity builds a state-of-the-art DNN classifier based on the publicly available dataset Food-101 [5]. The classifier is embedded onto the Edge Tensor Processing Unit (TPU) of Coral [6], which is a low-power DNN accelerator. This will enable to benchmark the developed technology against commercially available hardware solutions.

2.5.2 Automotive – Object Recognition and Sound Localization

This use case focusses on localization and recognition of objects/sound generators. A sound event localization, detection, and tracking network has been developed and could be intended to be on an Field-Programmable Gate Array (FPGA) which emulates the analogue parts of the circuit. A simar demonstrator based on the same principle might be developed by replacing the sound measurements by object visualization through a video camera. Additionally, radar-based object detection might be developed based on hardware developed in the project. Radar has the advantage over video as its network size is considerably smaller.

2.5.3 Digital Industry – Pattern Recognition (Keyword Spotting)

Speech processing enables natural communication with smart phones or smart home assistants. However, continuously performing speech recognition is not energy-efficient and would drain batteries of smart devices. Instead, speech recognition systems passively listen for utterances of certain wake

Figure 2.3 Possible inputs for the western food classification DNN [5].

words to trigger the continuous speech recognition system on demand [8]. In the project, “speech command datasets” have been analysed and features were extracted, and processing pipelines were implemented. The pipelines were used to explore different SNN algorithm approaches. Hybrid variants will be specified and simulated. After the hybrid variants are evaluated, the algorithms will be integrated into full SNNs.

2.5.4 Consumer – Coaching Biomechanical Assistance (Running)

This use case focusses on real-time running coaching. From an optimized database infrastructure of runners' user data and an improved classification neural networks will be trained. New software that will facilitate broader data and image assimilation from users and classification will be developed of additional input parameters.

2.5.5 Medical Health – Medical Image Denoising

Efficient medical image denoising is essential on mobile X-ray systems. To facilitate this, dataset specification and analysis of the noise characteristics are being made. This shows to be essential and challenging as part of the noise is signal-dependent. Metrics are being proposed to measure and quantify image quality comparisons, and specifications are set for the test cases to be performed on the SNN implementations.

2.6 Neuromorphic Hardware Technologies Being Developed

The developments in TEMPO are still ongoing; it is planned leverage the developed hardware and application results into the envisioned use case applications and related demonstrators by the end of 2022.

The project started with the process technology pathfinding work to enable neuromorphic and AI applications to leverage embedded non- volatile memories (eNVMs). This pathfinding work included the design of process technology test vehicles and process flows. At the same time, core building blocks and accelerator architectures have been designed to leverage the memory technologies in the application demonstrators. Basic neuromorphic building blocks were investigated with a focus on the development of neuromorphic–ready NVM blocks, the modelling and simulation of eNVM, the quantification of the technology features and neuromorphic implementation of eNVM. 3D specifications suited for DNN accelerators have been defined and a design flow to be able to quantify performance and energy impact of 3D interconnect has been set-up. Design and architecture exploration, specification, and design of critical building blocks to enable full accelerator IP blocks has been done.

Later in the project, the first hardware and algorithms were leveraged towards the applications via the different use cases. In the domain of **emerging technologies**, basic neuromorphic building blocks (MRAM, Oxide Random Access Memory - OxRAM and FeFET) were investigated, with a focus the development of neuromorphic–ready NVM blocks, modelling and simulation of eNVM, and the quantification of the technology features and neuromorphic implementation of eNVM. Also features of embedded memory for Neuromorphic Accelerators have been investigated, such as multi-level memory and the synapticity/plasticity of the memories. In the domain of **technology integration**, compact models were created based on the data from first OxRAM, Phase-Change Random Access Memory (PCRAM) and FeFET implementations. Also, 3D specifications suited for DNN accelerators have been defined and the 3D place and route (PnR) design flow has been created to quantify performance and energy impact of the 3D interconnects. An illustration of an envisioned 3D interconnect roadmap suitable for typical neural accelerators is illustrated in Figure 2.4. In the domain of **neuromorphic hardware design and architectures**, potential design, and architectures of the most critical neuromorphic DNN and SNN building blocks in terms of power, performance and area have been explored. Finally, in the domain of **application specification and demonstration**, the use cases and related data sets have been defined and the reference platform has been chosen and benchmarked. These uses cases have been elaborated in section 2.5. Theses use cases are being implemented towards demonstration.

TEMPO will continue to combine both the developed hardware and application results to enable demonstration of energy efficient accelerators for the different use cases defined in the project.

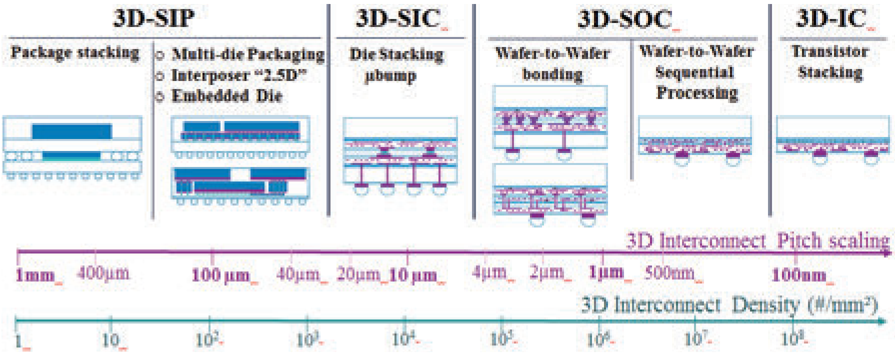


Figure 2.4 3D landscape, ordering of 3D technologies according to the system-level wiring hierarchy [11][12].

2.7 Conclusion

In most application domains, the amount of data produced in sensors and devices is exploding. Sending this data to the cloud costs energy, latency, and is often not preferred for privacy reasons. Applications relying on artificial intelligence in the edge require intelligent energy-efficient local processing. The TEMPO project develops such energy efficiency neuromorphic hardware with emerging memory technologies like MRAM, FeRAM and RRAM, and develops technology platforms for emerging semiconductor devices. In the domain of emerging technologies, the project investigated the different memory types to confirm their suitability and limitations towards offering the needed neuromorphic features and implementation. Compact models were created based on the first memory implementations and a 3D interconnect roadmap suitable for typical neural accelerators has been designed and presented. To enable neuromorphic hardware design, the architecture of the most critical neuromorphic DNN and SNN building blocks have been explored in terms of power, performance, and area. This paves the way to demonstrate these technologies for the neuromorphic workloads required in the envisioned use cases. These use cases and their dataset requirements have been specified as discussed in this article. These use cases cover a broad range of application fields within automotive, consumer electronics, digital industry, food, and medical health. As such, the TEMPO project is successfully pursuing its goal to broaden the applicability of integrated neuromorphic hardware.

Acknowledgements

TEMPO (Technology & hardware for nEuromorphic coMPuting) is a European innovation project. This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826655. The JU receives support from the European Union's Horizon 2020 research and innovation programme, and from Belgium, France, Germany, The Netherlands, and Switzerland. TEMPO was kicked off on the 1st of April 2019 and has a duration of three years. The consortium of this ambitious project consists of eighteen members from five different European countries. Imec takes the lead as the sole Belgian consortium partner. The other consortium members are, for France: CEA-Leti, ST-Microelectronics Crolles, ST-Microelectronics Grenoble and Thales Alenia Space. For Germany: Bosch, Fraunhofer (EMFT, IIS, IPMS), Infineon, Valeo, Innosent, TU Dresden and Videantis. For the Netherlands: imec the Netherlands, Philips Electronics, Philips Medical Systems and Ato-gear. For Switzerland: aiCTX and the

University of Zürich. For more information on the project and the consortium partners: https://tempo-ecsel.eu/

References

[1] TEMPO project. Technology and Hardware for Neuromorphic Computing. Available online at: https://tempo-ecsel.eu/

[2] Deloitte, “Technology, Media and Telecommunications Predictions (TMT) 2021. Available online at: www2.deloitte.com/be/en/pages/technology-media-and-telecommunications/articles/tmt-predictions.html.

[3] NVIDIA, “Annual investor day”, 12 April 2021. Available online at: https://investor.nvidia.com/events-and-presentations/events-and-presentations/event-details/2021/NVIDIA-Annual-Investor-Day/default.aspx.

[4] Deloitte, “Hitting the accelerator: the next generation of machine- learning chips”, 2017. Available online at: www2.deloitte.com/content/dam/Deloitte/global/Images/infographics/technologymediatelecommunications/gx-deloitte-tmt-2018-nextgen-machine-learning-report.pdf.

[5] TensorFlow, “TensorFlow datasets: a collection of ready-to-use datasets – dataset Food-101”, 2021. Available online at: www.tensorflow.org/datasets/catalog/food101.

[6] Coral, “Products”, 2021. Available online at: https://coral.ai/products/.

[7] Robert Bosch, “Embedded siren detection”, 2021. Available online at: www.bosch.com/stories/embedded-siren-detection.

[8] S. Mittermaier, L. Kürzinger, B. Waschneck, and G. Rigoll, “Small-Footprint Keyword Spotting on Raw Audio Data with Sinc- Convolutions”, arXiv, 1911.02086, 2020. Available online at: https://arxiv.org/abs/1911.02086.

[9] Ato-Gear, “Arion”, 2021. Available online at: www.arion.run.

[10] Philips, “Mobile digital radiography system”, 2021. Available online at: www.philips.nl/healthcare/product/HC712001/mobilediagnost-wdr-mobile-digital-radiography-system.

[11] imec, “A 3D technology toolbox in support of system-technology co-optimization”, 2019. Available online at: www.imec-int.com/en/imec-magazine/imec-magazine-july-2019/a-3d-technology-toolbox-in-support-of-system-technology-co-optimization.

[12] S. M. Samavedam, et al., “Future Logic Scaling: Towards Atomic Channels and Deconstructed Chips,” IEEE International Electron Devices Meeting (IEDM), 2020. https://doi.org/10.1109/IEDM13553.2020.9372023

3

Tools and Methodologies for Training, Profiling, and Mapping a Neural Network on a Hardware Target

Alexandre Valentian[1], Simon Narduzzi[2], Muhammad Arsalan[5], Kay Bierzynski[5], Stefano Traferro[4], Preetha Vijayan[4], Amirreza Yousefzadeh[4], Manolis Sifalakis[4], Rene Van Leuken[8], Dylan Muir[7], Rashid Ali[3] Maen Mallah[3], Bijoy Kundu[3], Loreto Mateu[3], and Mario Diaz Nava[6]

[1]CEA, France
[2]CSEM, Switzerland
[3]Fraunhofer IIS, Germany
[4]imec, The Netherlands
[5]Infineon Technologies, Germany
[6]STMicroelectronics, France
[7]SynSense, Switzerland
[8]TU Delft, The Netherlands

Abstract

The European project ANDANTE [1] aims at providing neuro-inspired and/or energy-efficient hardware accelerators for running AI applications at the edge. Given the wealth of applications targeted, with various processing needs and sensors involved, several implementations are pursued in parallel: (1) fully digital or analog-mixed signal; (2) with classical coding or spike coding; (3) leveraging different embedded Non-Volatile Memory (NVM) technologies. However, what do all have all in common? it's the need for adequate tools and methodologies for training and deploying neural network models, considering hardware constraints. This Chapter provides details on what has been developed and used in the frame of the ECSEL JU ANDANTE project. Firstly, a neural network must be learnt, considering limited hardware resources, thus exploiting quantization and sparsity for instance. When dealing with Spiking Neural Networks (SNNs), the training phase is even more critical, depending on the neuron model and making use of various strategies (direct training versus conversion). Then, the network must be mapped on the target accelerator, be it a spatially folded or spatially expanded architecture. In the latter case, graph transformation might be needed using a compiler. Finally, Key Performance Indicators (KPIs) must be extracted, underlying the need for a simulator/profiler.

Keywords: Machine learning, deep learning, artificial neural networks, spiking neural networks, optimisation, quantization, pruning, distillation, NN model transformation, sparsity, ANN-to-SNN conversion, accuracy, NN compiler, mapping strategy, simulation, profiling

3.1 Introduction

3.1.1 Edge Computing Benefices and Challenges

Edge computing is creating new opportunities for Internet of Things (IoT) applications. Through machine learning, objects become intelligent and can process a large amount of information. However, most of this processing today still takes place in the cloud, and it comes at several costs: infrastructure, reliability, security, speed, and energy. Firstly, the infrastructure to process data from heterogeneous devices needs an extensive infrastructure to gather, transform, and store the data and the devices themselves need connectivity and the corresponding energy to send the data. Therefore, having the data stored and analysed at the edge can reduce the infrastructural costs, save energy, and increase globally data processing efficiency. Secondly, for applications that are critical and need high availability (such as pipeline monitoring), a reliable and secure connection is necessary.

Having devices that can decide at the edge can mitigate the risk associated with the loss of connection and prevent data from being accessed by a third party to ensure security and privacy. Moreover, intelligent edge devices are also necessary for applications where decision speed (low latency) is critical, such as autonomous driving, as having data transferred to the cloud is inconceivable: the latency associated with the connection might result in the life or death of people. Finally, the energy associated with the transfer and data processing in the cloud is still enormous: 40% of the energy used in mobile streaming comes from the mobile cellular network. Therefore, in the context of the climate crisis, edge machine learning can substantially reduce the carbon footprint associated with data processing in the cloud. The reduction of infrastructure costs, reduction of communication bandwidth, improvement of security and privacy, and availability of services are by-products of deploying efficient, intelligent, and cost-effective devices at the edge.

With the wide acceptance of Deep Learning (DL) in the last decade, it has become evident that classic deep learning cannot scale, as performant networks use an enormous amount of energy and memory capacity, and the models are becoming larger as their computational power needs increase. Moreover, Moore's law is ceasing to apply, and we need new computational paradigms to increase computational performance with a reduced energy budget. With the evolution of IoT devices, deep learning models are now deployed at the edge, allowing local real-time decision-making, efficient pre-processing, and privacy-preserving applications. Optimizations have been developed in the past few years to allow the deployment of these networks within restricted resource environments; quantization, pruning, distillation, are some of them, which are either applied during training or post-training of the neural network. While these techniques offer a partial solution for the deployment on edge devices, a lot of engineering is still required to design models that fit within the constraint of the hardware. An alternative emerging machine learning technology to reduce energy relies on SNNs, which are structures imitating the neurons in the brain. Their computational efficiency is thought to be due to the coding style of the biological neurons, which communicate using electrical discharges, called spikes, that travel from one neuron to the other using synaptic connections.

While industry leaders Intel, ARM, Google, and NVIDIA are developing systems targeting large-scale computation based on Graphics Processing Units (GPUs) or specialised AI processors for generic AI applications in the cloud, a parallel branch targets low-power applications at the edge, with algorithmic solutions that will only be efficient if they can run on suitable hardware solutions. Currently, much effort is put into developing low-power accelerators for artificial neural networks, and to some extent, spiking neural network. Academia is also putting effort into the development of technologies targeting edge processing: the recent development of memristors and Ferroelectric Field-Effect Transistor (FeFET) technologies herald a new era of ultra-low-power hardware to accelerate neural networks [21][22].

While most accelerators target generic applications, there are still many limitations on the hardware that make them suboptimal for specific tasks:

limitation in speed, memory size, supported operations (spiking or digital), or energy consumption. This wide choice of embedded systems makes it challenging to identify the relevant hardware suitable for a particular application. This major challenge restricts the adoption and dissemination of ultra-low-power applications, as many efforts are put into studying and researching the most suitable device.

3.1.2 Artificial Neural Networks (ANNs) and Spiking Neural Networks (SNNs)

In biology, neurons communicate through current inputs called action potentials, or spikes. When a neuron receives input stimuli (spikes) from other neurons, they depolarize the neuron cell membrane by changing the concentration of ions inside and outside of the cell membrane, creating a potential. The strength of the depolarization depends on the strength of the synaptic connection between the pre- and post-synaptic neurons. The succession of depolarizations events leads to an increase of the membrane potential. If the cell membrane potential increases to a precise threshold voltage, it triggers a cascade effect leading to the emission of a spike.

In 1958, Frank Rosenblatt created the first model of a neuron generating binary decisions, simulating the emission of a spike, or not. The perceptron was a single neuron model performing computation using multiple weighted input values, simulating the strength of the synaptic connections, using a weight matrix. When the weighted sum of input reached a certain value, the neuron output switched. Current deep learning relies on variants of this algorithm, by creating stacked structures (layers) of neurons that combine and transform the information in a non-linear manner, resulting in impressive performance in a wide variety of tasks.

Deep learning algorithms can be accelerated on dedicated hardware to provide low-power solutions for edge applications. ASICs for deep learning inference accelerators offer better area and energy efficiency than GPUs, FPGAs or CPUs but at the cost of less flexibility [1]. Since ANNs perform multiply and accumulate (MAC) operations, the hardware pursues to acceleration such operations by parallelizing them. To overcome the von Neumann bottleneck, ASIC architectures based on analog in-memory computing with crossbar arrays to perform the MAC operation are pointed out as a relevant solution when it comes to low latency and high energy efficiency. Such inference accelerators have on-chip memory buffers as well as processing elements where the weights are stored individually to avoid data movement during inference.

Although inspired by the biological nervous systems, ANNs are yet unable to capture the sophisticated neurocomputational features of biological neurons. To bridge this gap, DL community has come up with a third generation of ANNs known as SNNs. SNNs are more closely mimicking biological neural networks than artificial neural networks that are rate-based. This type of neuron is represented by a membrane state and therefore incorporates the concept of time. Spiking neural networks, in contrast to artificial ones, only send "spikes" and not digital values. However, they can represent values using spike trains, which rate can be as equivalent to values processed by artificial deep neural networks.

Research on spiking neural networks is still on-going. The recent development of neuromorphic hardware platforms has allowed simulation of large-scale brain models. However, how to perform deep learning using these types of neurons is still unclear. In particular, a lot of different types of spiking neurons exist, and no standard has been agreed on yet. This wide variety of neurons must also be considered by designers of neuromorphic hardware, so that researchers can assess the suitability of models.

The neurons in SNN are described on different abstraction levels starting from the most realistic and complex model, Hodgkin-Huxley (HH) model, to the leaky integrate-and-fire (LIF) model which is the simplest and most computationally efficient model bearing the neurocomputational properties [23]. LIF introduces a leaky term to the integrate-and-fire (IF) model that causes neuron potential decay over time making it more biologically plausible.

With the advancement of research on spiking neural networks, academia and industry have developed accelerators and processors specialized in supporting this type of algorithm. A few research institutes and companies develop large-scale hardware solutions to simulate spiking neural networks, like SpiNNaker, IBM TrueNorth, and Intel Loihi and Loihi 2. While reasonably accurate at simulating large-scale brain dynamics, these processors do not target ultra-low-power edge applications and still use a considerable amount of energy. As their primary purpose was to simulate SNN, the processing happening in these accelerators is unsuitable for common industrial applications, as developed nowadays using deep learning. Research is still actively investigating suitable event-based device for industrial applications, and now we observe the emergence of new hardware accelerator relying on binary events computed in a synchronous manner, meeting halfway between the pure asynchronous SNNs and the synchronous processing of ANNs.

In the ANDANTE project, this type of efficient hardware neuromorphic accelerators is being addressed based on new embedded memory technologies such as PCM, OxRAM, MRAM and FeFET, novel ANN/SNN architectures combining analog digital mixed-signal designs, which call for dedicated tools and methodologies.

The rest of this Chapter is organized as follows: Section 3.2 provides the state-of-the-art in neural network training, exploiting quantization, sparsity and showing different strategies for training spiking neural networks. Section 3.3 presents further refinements to sparsity, to exploit temporal sparsity (in addition to the weight and activation ones) by adding a new layer called Temporal Delta Layer. Section 3.4 describes how to map a neural network onto a spatially expanded, in memory computing-based architecture: in such a case, the neural network weights must be adequately clustered or duplicated on the various NVM arrays. Finally, Section 3.5 shows the mapping of spiking neural networks on a hardware target implementing LIF neurons with recurrent connections. Finally, Section 3.6 gives clues on why it is important to profile a neural network topology.

3.2 State-of-the-art of key aspects of Neural Networks

3.2.1 ANN and SNN Hardware Aware Design

Hardware-aware design of artificial and spiking neural networks is still a multistep process. Since no generic design and simulation tools are available for custom neuromorphic hardware platforms, it is still required to deploy the model on the physical devices to obtain the key performance indicators of the application. As shown in Figure 3.1, the optimization of a neural network for a specific edge device is an iterative process involving hardware/software co-design. The first iterative cycle is the development of an accurate model solving the task to which it is designed, and the second cycle consists in the embedding and evaluation of the model. The model can then be further optimized towards the optimization of edge KPIs, necessitating a new training iteration phase followed by deployment. The process can be automatized using automated search procedures like Network Architecture Search (NAS), which have demonstrated to be suitable solutions for the design of a model respecting the constraint of their end-deployment platform [77]. However, this framework still contains major challenges. Indeed, the deployment on a device is often complicated and requires a manual adaptation of the model to allow the neural network to run on the device. Moreover, some platforms have a specific instruction set or a variable data representation (float or integer),

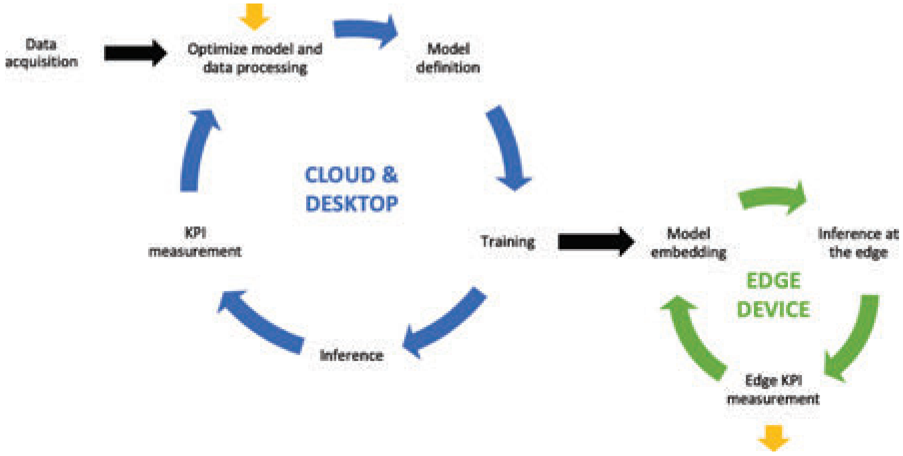


Figure 3.1 Networks to hardware workflow.

requiring a quantization step either during training (defined at the creation of the model) or after training, usually impacting the overall performance of the network. Finally, platforms are often behind the latest software developments in the creation of layers in neural networks, which makes some architectures impossible to deploy due to the existence of unsupported layers.

Regarding the automatic search of architectures for a certain platform, the computational cost is still very intensive and usually must be replicated for each new platform, despite recent improvements in this direction [69]. Therefore, flows and techniques have been developed to design efficient neural networks for neuromorphic hardware platforms. Some of them are described in the next paragraphs.

3.2.2 Sparsity

Reducing energy consumption is a critical point for neural network models running on edge devices. In this regard, reducing the number of MAC operations of DNNs running on edge hardware accelerators will reduce the energy consumption during inference. Optimizations have been developed in the past few years to allow the deployment of these networks within restricted resource environments; quantization [2], pruning [3], distillation [4], are some of them, which are applied either during training or post-training of the neural network. Great emphasis is also put on the development of efficient accelerators, that reach competitive performance compared to CPUs and GPUs. Recent hardware accelerators include optimization techniques such as computational reduction by zero-skipping [5][6][7], that skip zero weight computation in and are therefore optimized for very sparse neural networks.

Efforts have been made toward the sparsification of deep neural networks to reduce the memory footprint of the models deployed at the edge. Pruning is a method used to achieve weight [12] and feature map 13[14][15] sparsification to remove redundant information and subsequently reduce network computations. In SNNs, spikes and synaptic computation reduction are mostly exploited through temporal and spatial sparsity. Temporal sparsity of SNNs have inspired training techniques in deep learning [16][17], targeting time-series applications. Recently, regularization techniques have been applied to SNN training [18][19] to increase spatial sparsity, and during BP-trained DNNs training prior to SNN conversion [11][20][79].

3.2.3 ANN-to-SNN Conversion

Spiking neural networks can potentially save much more energy than continuous-valued Artificial Neural Networks due to their sparse nature and event-driven computations. While SNNs may provide a large panel of advantages, their training is still complicated, as the current hardware and training algorithms are not suitable to train SNN in an asynchronous manner. Therefore, one common technique to create performant SNNs is to convert them from a previously trained ANN.

Early attempts to convert ANN to SSN comprise the work of [70] where neurons of a Convolutional Neural Network were transformed to leaky-integrate and fire (LIF) neurons with refractory periods. A similar technique [71] used a weight normalization scheme in an ANN to regulate the firing rate of the converted SNN. Another work [72] developed a conversion method using spiking neurons that adapt their firing threshold to reduce the number of spikes needed to encode information.

One largely used technique of conversion of ANN to SNN has been developed by Rueckauer [10]. It is based on scaling of the weights of the pretrained SNN such that the firing rate of the neurons match the activation values of the ANN. While this technique supports a wide range of layers, it requires a long simulation time for the model to reach competitive accuracy. Recent methods [73][74][75] adjust the threshold values of the neurons to reduce the inference latency.

3.2.4 Surrogate Gradient Descent

Spiking neuron models commonly incorporate highly non-linear transfer functions, such as the Heaviside function, to map from internal state variables to binary output events.

$$S\left(V_{mem}\right) = H\left(V_{mem} - V_{th}\right) \tag{3.1}$$

These functions often have poorly behaved or undefined derivatives. In the example here *dS/dV_{mem} = 0* everywhere. When used in conjunction with gradient-based optimisation methods such as error backpropagation [9], these poorly behaved derivatives propagate to cause the gradients of parameters to be not informative. Standard gradient-based training techniques cannot therefore be directly applied to SNNs.

One method to work around this limitation is to define a *surrogate gradient* for the SNN transfer function. In this approach the derivative of the transfer function is defined using an auxiliary "surrogate" function, ideally with similar behaviour to the true transfer function, but with better-behaved derivatives. For example, instead of the non-linear Heaviside function, a ReLU function can be used as an approximation for computing the gradient in the backwards pass.

$$\hat{S}\left(V_{mem}\right) = \max\left(V_{mem} - V_{th}, 0\right) \tag{3.2}$$

$$\frac{dS}{dV_{mem}} \equiv \frac{d\hat{S}}{dV_{mem}} = V_{mem} > V_{th} \tag{3.3}$$

This method permits SNNs to be trained using gradient-based optimisation algorithms such as SGD [78][79][80] and Adam [81]. Recently this approach has been used to integrate SNNs with industry-standard automatic differentiation libraries such as PyTorch and Jax, to permit training of deep SNNs [82]. In this way not only the weights of a network can be optimised, but in addition all the auxiliary parameters of an SNN such as time constants and thresholds [82].

3.2.5 Neural Engineering Object (Nengo) Simulator

The Neural Engineering Object (Nengo) is a neural network simulation tool for large-scale neural systems with applications in cognitive science, psychology, AI, and neuroscience [25]. Nengo offers NengoDL, a deep learning simulator, which enables for easy integration of the TensorFlow library and access to advanced features such as convolution connections. Using a neural engineering framework NEF with Nengo designs neural network models for application in machine learning and deep learning such as inductive reasoning [27], gesture sensing [26], action selection [27], speech production [84] and image classification [85], etc.

NengoDL uses NEF for building neuron models for building biologically plausible neural networks. NEF provides the principles of representation, transformation, and dynamics to construct a neural model. The NEF encodes the incoming time varying input data of real numbers and based on the input data; a specific amount of current is injected into a single neuron model. This current causes the neuron to spike and the spiking behaviour is controlled by the tuning the curve of the neuron models. The tuning curve is determined by the bias, gain of the neuron and the encoding weights. In the decoding stage, an exponentially decaying filter is applied to the spike train resulting in a spike generating postsynaptic current [25].

The strength of the postsynaptic current is defined by its amplitude which is affected by various factors. The NEF summarizes these factors in the form of a connection weight matrix representing the strength of the connection between two neural populations. These matrices can be factorized into smaller matrices allowing to efficiently run large-scale neural models on low commodity hardware [25].

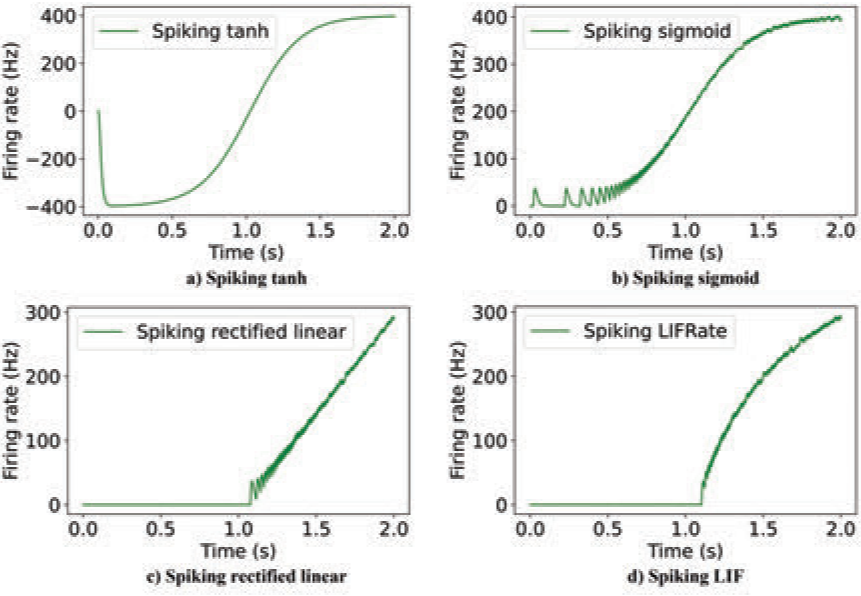
An information is represented by a Nengo ensemble, and a connection defines how the information is transformed. Nengo uses an object model to translate the ensembles and their connections into a network of interconnected neurons. In this way, it acts as a neural compiler, converting high-level functional models to low-level models. Nengo defines six core objects as an object model: 1) ensemble, 2) node for non-neural information such as sensory inputs, 3) connection, 4) probe for data collection during simulation, 5) network for interconnected nodes and ensembles, and 6) model. Because of the separation of model construction and simulation, Nengo models can be used on a variety of simulators [25].

In addition to the biological plausible neurons, nengoDL allows to use rate-based neurons such as LIFRate, Rectified Linear, Sigmoid and Tanh by converting them to their spiking version using wrappers that take some function and return an instantaneous firing rate. These wrappers are [26]:

1) **Regular Spiking**: takes the instantaneous firing rate and integrates it multiplied by a timestep.
2) **Poisson Spiking**: Given an instantaneous rate, this wrapper draws a sample from a Poisson distribution. The value of the distribution is this instantaneous firing rate.

3) **Stochastic Spiking**: is kind of a mix between the two, and the difference mostly shows up when neurons can spike more than once per timestep.

In Figure 3.2, some conversion examples are illustrated. These neurons are created by employing Regular Spiking to convert rate-based neurons to their spiking counterparts. For example, Figure 3.2(a) is created $\tau_{ref} = 0.0025$ indicating that the firing will saturate at 400 Hz. The neuron begins in a blank state (i.e., no input current, no membrane current, etc.), implying that the neurons are doing nothing when the simulations begin, and it takes a few time steps for the neuron to get going. The curve becomes a little noisy around the middle because the neuron has modest firing rates and so few spikes in that area. Moreover, it can be seen that the neuron is showing two kinds of spikes, positive and negative. Because this type of spiking behaviour isn't biologically reasonable, it won't operate on most neuromorphic technology. Similarly, Figures 3.2(b), (c), and (d) represent the spiking version of Sigmoid, Rectified Linear and LIFRate based neuron. It should be noted that the curve's slope is determined by the neuron's gain. The gain of the neurons has been modified in these cases to produce less noisy curves.



a) Spiking tanh

b) Spiking sigmoid

c) Spiking rectified linear

d) Spiking LIF

Figure 3.2 Spiking neuron models [26].

3.3 NN Transformation: Temporal Delta Layer

This Section focuses on a transformation applicable to DNNs which generates temporal activation sparsity during training and exploit it during inference.

The energy consumed by running DNNs on hardware accelerators is dominated by the number of memory read/writes and multiply-accumulate (MAC) operations. As a potential solution, the role of activation sparsity in efficient DNN inference is proposed. i.e., as the predominant operation in DNNs is matrix-vector multiplication of weights with activations, skipping operations and memory fetches where (at least) one of them is zero can make inference more energy efficient.

In this Section, a new DNN layer (called temporal delta layer) whose primary objective is to induce temporal activation sparsity during training is presented. The temporal delta layer promotes activation sparsity by performing delta operation through activation quantization and l_1 norm-based penalty to the cost function. During inference, the resulting model acts as a conventional quantized DNN with high temporal activation sparsity.

3.3.1 Temporal Delta Layer: Training Towards Brain Inspired Temporal Sparsity for Energy Efficient Deep Neural Networks

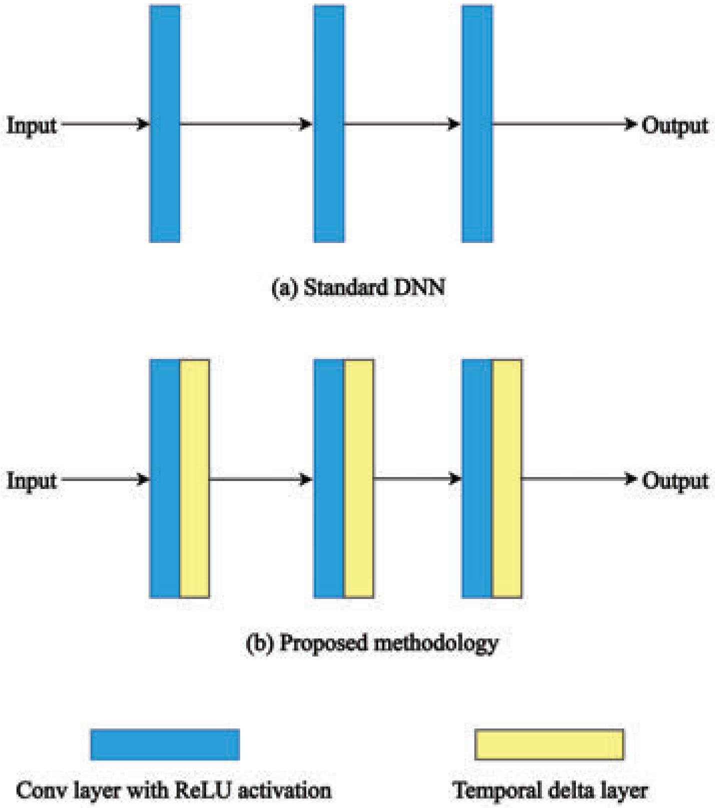
DNNs have lately managed to successfully analyses video data to perform action recognition [27], object tracking [28], object detection [29], etc., with human-like accuracy and robustness. Unfortunately, the high accuracy of DNNs comes with high compute and memory costs, resulting in high energy consumption. This makes them infeasible for always-on edge devices.

Over the years, techniques like network pruning, quantization, regularization, and knowledge distillation [30][31][32] have helped in reducing the model size footprint resulting in overall lesser computation and memory consumption. Noticeably, sparsity is an underlying feature in all the solutions. This is notable, as sparse tensors provide the potential to skip computations that involve multiplication with zeroes. Also, they are easier to store and access in memory. Structural sparsity (of weights) and spatial sparsity (of activations) are well-researched topics in DNN literature [33]. However, temporal activation sparsity is comparatively less explored in the context of DNN, although it is a popular concept in neuromorphic computing.

The concept of change or delta-based processing is taken from the human retina to the training and inference phases of deep neural networks [34]. DNN inference which processes each frame separately with no regard to the temporal correlation is dense and obscenely wasteful. Whereas processing only the changes in the network can lead to zero-skipping in sparse tensor operations reducing redundant operations and memory accesses.

Therefore, the proposed methodology in this work induces temporal sparsity to potentially any DNN, by means of a new layer (called Temporal Delta Layer), which can be introduced in a DNN at any phase (training, refinement, or inference only). This new layer can be integrated into an existing architecture by placing it after all or some of the ReLU activation layers as deemed computationally beneficial (see Figure 3.3).

The inclusion of this layer does not require any change to the preceding and following layers. Moreover, during the training phase, the new layer adds a novel sparsity penalty to the overall cost function of the DNN. This l_1 norm-based penalty minimizes the activation density of the delta



(a) Standard DNN

(b) Proposed methodology

Figure 3.3 (a) Standard DNN and (b) DNN with temporal delta layer.

maps (i.e., temporal difference between two consecutive feature maps). Apart from that, two activation quantization methods, namely fixed-point quantization (FXP) and learned step-size quantization (LSQ), are also compared in conjunction with the new layer.

The inclusion of this layer does not require any change to the preceding and following layers. Moreover, during the training phase, the new layer adds a novel sparsity penalty to the overall cost function of the DNN. This l_1 norm-based penalty minimizes the activation density of the delta maps (i.e., temporal difference between two consecutive feature maps). Apart from that, two activation quantization methods, namely fixed-point quantization (FXP) and learned step-size quantization (LSQ), are also compared in conjunction with the new layer.

3.3.2 Related Works

Although DNNs are in essence bio-inspired, they have not been able to find the balance between power consumption and accuracy yet, especially while dealing with computationally heavy streaming signals. On the other hand, the brain's neocortex handles complex tasks like sensory perception, planning, attention, and motor control while consuming less than 20 W [35]. Scalable architecture, in-memory computation, parallel processing, communication using spikes, low precision computation, sparse distributed representation, asynchronous execution, and fault tolerance are some of the characteristics of the biological neural networks that can be leveraged to bridge the energy consumption gap between the brain and DNNs [36]. Among these, the proposed methodology focuses on the viability of using sparsity within DNNs to achieve energy efficiency. During a matrix-vector multiplication between a weight matrix and an activation vector, zero elements in the tensor can be skipped leading to computational as well as memory access reduction (see Figure 3.4).

There are broadly two types of sparsity available in DNNs: weight sparsity (related to the interconnect between neurons) and activation sparsity (related to the number of neurons). Furthermore, activation sparsity can be categorized into spatial and temporal sparsity, which exploits the spatial and temporal correlation within the activations, respectively, [38]. Unlike weight and spatial sparsity [39][40][41][42][43][44], exploiting the temporal redundancy of DNNs while processing streaming data to reduce energy consumption is a relatively less explored idea. Exploiting temporal sparsity

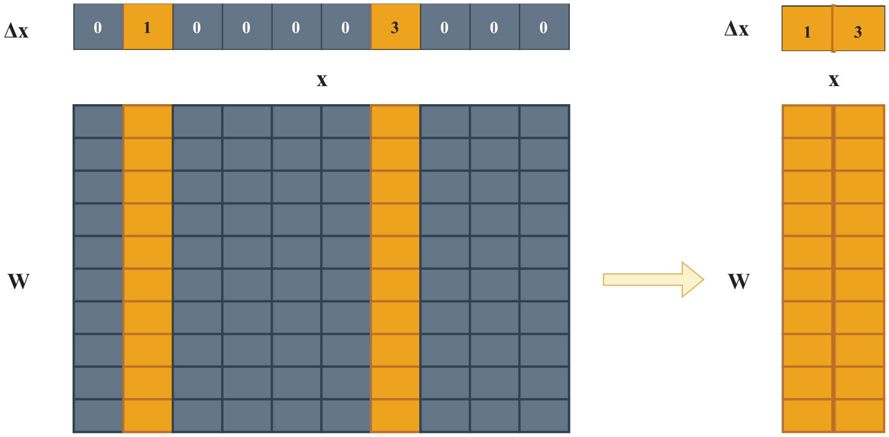


Figure 3.4 Sparsity in Δx can save multiplications between Δx and columns of W that correspond to zero [37].

translates to skipping re-calculation of a function when its input remains unchanged since the last update.

One of the methods to exploit temporal sparsity is to use the compressed representation (like H.264, MPEG-4, etc.) of videos at the input stage itself. These compression techniques only retain a few key-frames completely and reconstruct others using motion vectors and residual error, thus using temporal redundancy [45][46]. Another path includes finding a neuron model which is somewhere in between "frame-based DNN" and "event-based spiking neural networks". This Section describes an attempt in the direction. A similar work, CBInfer [7] proposes replacing all spatial convolution layers in a network with change-based temporal convolution layers (or CBconv layers). In this, a signal change is propagated forward only when a certain threshold is exceeded. Likewise, [48] tapped into temporal sparsity by introducing Sigma-Delta Networks, where neurons in one layer communicated with neurons in the next layer through discretized delta activations. An issue when it comes to CBInfer is the potential error accumulation over time as the method is threshold-based. If the neuron states are not reset periodically, this threshold can cause drift in the approximation of the activation signal and degrade the accuracy. Whereas sigma-delta scheme experiments on smaller datasets like temporal MNIST, which might not be a reliable confirmation of the method's effectiveness.

3.3.3 Methodology

In video-based applications, traditional deep neural networks rely on frame-based processing. That is, each frame is processed entirely through all the layers of the model. However, there is very little change in going from one frame to the next through time, which is called temporal locality. Therefore, it is wasteful to perform computations to extract the features of the non-changing parts of the individual frame. Taking that concept deeper into the network, if feature maps of two consecutive frames are inspected after every activation layer throughout the model, this temporal overlap can be observed. Therefore, we postulate that temporal sparsity can be significantly increased by focusing the inference of the model only on the changing pixels of the feature maps (or deltas).

3.3.3.1 Delta inference

A new layer is introduced that calculates the delta (or difference) between two temporally consecutive feature maps and quantifies the degree of these changes at only relevant locations in the frame. Since zero changes are not propagated through the layer, the role of this layer may be perceived as an ”analog event propagation”. It is considered an ”analog event” as it is not the presence of change, but the magnitude of change that is propagated through. To better understand it mathematically, in a standard DNN layer, the output activation is related to its weights and input vector through Equations (3.4) and (3.5).

$$Y_t = WX_t + B \tag{3.4}$$

$$Z_t = \sigma(Y_t) \tag{3.5}$$

where W and B represent the weights and bias parameters, Xt represents the input vector, and Yt represents the transitional state. Then, Zt is the output vector which is the result of s(.) - a non-linear activation function. t indicates that the tensor has a temporal dimension. However, in the temporal delta layer, weight-input multiplication transforms into,

$$\Delta Y_t = W\Delta X_t = W(X_t - X_{t-1}) \tag{3.6}$$

$$\begin{aligned} Yt &= \Delta Yt + Yt - 1 \\ &= W(X_t - X_{t-1}) + W(X_{t-1} - X_{t-2}) + \cdots + Y_0, \ \ where \ \ Y_0 = B \\ &= WX_t + B, \end{aligned} \tag{3.7}$$

$$\Delta Z_t = Z_t - Z_{t-1} = \sigma(Y_t) - \sigma(Y_{t-1}), \ \ where \ \ \sigma(Y_0) = 0 \tag{3.8}$$

In Equation (3.4), instead of using Xt directly, only changes or ΔXt are multiplied with W. Using the resulting ΔYt, the corresponding Yt can be recursively calculated with Equation (3.5), where Yt−1 is the transitional state obtained from the previous calculation. Equation (3.8) is the final delta activation output that is passed onto the next layer.

Another notable difference between the standard DNN layer and the proposed layer is the role of bias. In delta-based inference, bias is only used as an initialization for the transitional state, Y0 in Equation (1.4). However, since bias tensors do not change over time, their temporal difference is zero and is removed from Equation (3.6).

Now, as the input video is considered temporally correlated, the expectation is that ΔXt and by association ΔZt are also temporally sparse. In essence, the temporal sparsity between consecutive feature maps is cast on the spatial sparsity of the delta map that is propagated. Additionally, Yt in Equations (3.4) and (3.7) are always equal. This indicates that if the input is the same, both standard DNN and temporal delta layer based DNN provide the same result at any time step.

3.3.3.2 Activation quantization to induce sparsity

There is temporal redundancy evident in feature maps of two consecutive frames. However, if looked closely, it can be observed that these feature maps are similar but not identical as shown in Figure 3.5(a) and (b). Therefore, if two such consecutive feature maps are subtracted, the resulting delta map has many near zero values, thus restricting the potential increase in temporal sparsity, Figure 3.5(c). This is mainly due to the higher precision available in the floating-point representation (FP32) of the activations. For example, in IEEE 754 representation, a single precision 32-bit floating point number has 1 bit for sign, 8 bits for the exponent and 23 bits for the significant. It not only leads to a very high dynamic range, but also increases the resolution or precision for numbers close to 0. The number nearest to 0 is about ±1.4 x 10−45. Therefore, due to high resolution, two similar floating-point values have difficulty going to absolute zero when subtracted. A plausible solution to decrease the precision of the activations is to use quantization.

A post-training quantization method (fixed point quantization [49]) and a quantization aware training method (learnable step size quantization [50]) are considered for comparison as a temporal sparsity facilitator for the new layer.

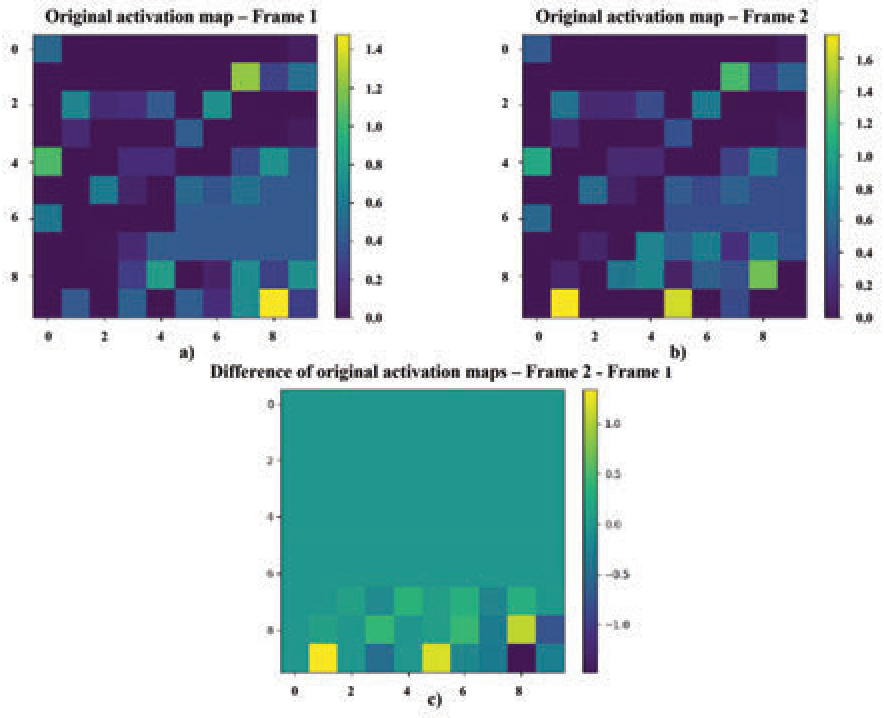


Figure 3.5 Demonstration of two consecutive activation maps leading to near zero deltas.

3.3.3.3 Fixed point quantization

In this method, the floating-point numbers are quantized to integer or fixed-point representation [49]. Unlike floating point, in fixed point representation, the integer and the fractional part have fixed length. This limits both range and precision. That is, if more bits are used to represent the integer part, it subsequently decreases the precision and vice versa.

Method: firstly, a bit-width is defined to which the 32-bit floating parameter is to be quantized, BW. Then, the number of bits required to represent the unsigned integer part of the parameter (x) is calculated as shown in Equation (3.9).

$$I = 1 + \lfloor log_2(max|x|) \rfloor \quad 1 < i < N \qquad (3.9)$$

A positive value of I means that I bits are required to represent the absolute value of the integer part, while a negative value of I means that the fractional part has I leading unused bits. Now, it is known that 1 bit is for sign, so the number of fractional bits, F, is given by Equation (3.10).

$$\mathrm{F} = \mathrm{BW} - \mathrm{I} - 1 \tag{3.10}$$

Considering the parameters, BW - bit-width, F - fractional bits, I - integer bits, and S - sign bit, Equation (3.11) maps the floating-point parameter x to the fixed point by,

$$Qx = \frac{C(R(x \cdot 2^F), -t, t)}{2^F} \tag{3.11}$$

where *R*(.) is the round function, *C*(*x,a,b*) is the clipping function, and t is defined as,

$$t = \begin{cases} 2^{BW-S}, & BW > 1 \\ 0 & BW \leq 1 \end{cases}$$

The fixed-point quantization, as shown above, is a straightforward mapping scheme and is easy to be included in the model training process during the forward pass before the actual delta calculation. However, it poses a limitation to the extent of quantization possible without sacrificing accuracy. Typically, an 8-bit quantization can sustain floating point accuracy with this method, but if the bit-width goes below 8 bits, the accuracy starts to deteriorate significantly. This is because, unlike weights, activations are dynamic and activation patterns change from input to input making them more sensitive to harsh quantization [51]. Also, quantizing the layers of a network to the same bit-width can mean that the inter-channel behaviour of the feature maps is not captured properly. Since the number of fractional bits is usually selected depending on the maximum activation value in a layer, this type of quantization tends to cause excessive information loss in channels with a smaller range.

3.3.3.4 Learned step-size quantization

Quantization aware training is the most logical solution to the drawback as it can potentially recover the accuracy in low bit tasks given enough time to train. Therefore, a symmetric uniform quantization scheme is considered called Learned Step size Quantization (LSQ). This method considers the quantizer itself as a trainable parameter which is trying to minimize the task loss using backpropagation and stochastic gradient descent. This serves two purposes: (a) step size, which is the width of quantization bins, gets to be adaptive through the training according to the activation distribution. It is vital to find an optimum step size because, as shown in Figure 3.6, if the step size is too small or too large, it can lead to the quantized data being a poor

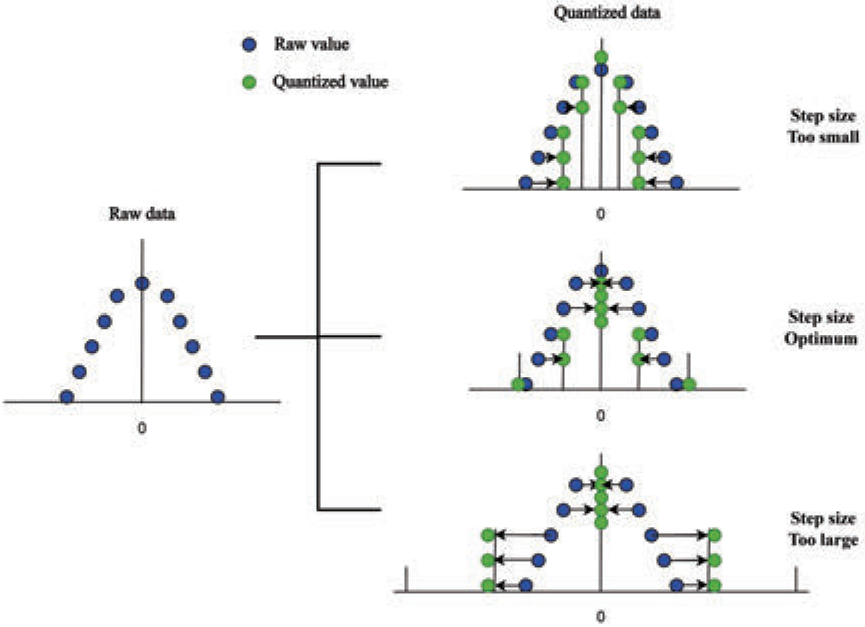


Figure 3.6 Importance of step size in quantization: on the right side, in all three cases, the data is quantized to five bins with different uniform step sizes, but without optimum step size value, the quantization can alter the range and resolution of the original.

representation of the raw data. (b) as the step size is a model parameter, it is also directly seeking to improve the metric of interest, i.e., accuracy.

Method: given: x - the parameter to be quantized, s - step size, QN and QP - number of negative and positive quantization levels respectively, and q(x;s) is the quantized representation with the same scale as x,

$$q(x;s) = \begin{cases} \left\lfloor \frac{x}{s} \right\rceil \cdot s & if - Q_N \leq \frac{x}{s} \leq Q_p \\ -Q_{N,s} & \frac{x}{s} \leq -Q_N \\ -Q_{P,s} & \frac{x}{s} \geq -Q_P \end{cases} \tag{3.12}$$

where $\lfloor \mathrm{a} \rceil$ rounds the value to the nearest integer. Considering the number of bits, *b*, to which the data is to be quantized, Q_N = *0* for unsigned and Q_N = 2^{b-1} for signed data. Similarly, Q_P = 2^{b-1} for unsigned and $2^{b-1} - 1$ for signed data.

The original LSQ method is slightly modified to remove the clipping function from the equations as (a) the bit-width, b, required to calculate

QN and QP is not known. This is because the bit-width is not pre-defined and is determined using the activation statistics of each layer while training which leads to a mixed precision model, which is more advantageous, and (b) clipping leads to accuracy drop as it alters the range of the activation. That is, if activations are clipped during training, there could be a significant difference between the real-valued activation value and the quantized activation value, which in turn affects the gradient calculations and, therefore, the SGD optimization.

Thus, in temporal delta layer, the forward pass of the quantization includes only scaling, rounding and de-scaling and can be mathematically expressed as,

$$q(x;s) = \left[\frac{x}{s}\right] \cdot s \tag{3.13}$$

The gradient of the Equation (1.10) for backpropagation is given by Equation (3.14.)

$$\nabla_s q(x,s) = \left[\frac{x}{s}\right] - \frac{x}{s} \tag{3.14}$$

3.3.3.5 Sparsity penalty

The quantized delta map, created using the above-mentioned methods, has a fair number of absolute zeroes (or sparsity) available. However, like the biological brain, learning can help in increasing this sparsity further. The inspiration for this came from an elegant set of experiments performed by Y. Yu et al. [52]. The experiment showed a particular 30 second video to rodent specimens and tracked their activation density during each presentation. It was found that activation density decreased as the number of trials increased, i.e., as the learning increased, the active neurons required for inference decreased. Adapting the said concept to this work, a l_1 norm-based constraint is introduced to the loss function. This is termed as the sparsity penalty. Therefore, the new cost function can be mathematically expressed as cost function = task loss + sparsity penalty, i.e,

$$\begin{aligned} &Cost\ function \\ &\quad = Task\ loss + \lambda \left(\frac{l_1\ norm\ of\ active\ neurons\ in\ delta\ map}{total\ number\ of\ neurons\ in\ delta\ map} \right) \end{aligned} \tag{3.15}$$

where task loss minimizes the error between the true value and the predicted value and, sparsity penalty minimizes the overall temporal activation density. The λ mentioned in Equation (1.12) refers to the penalty co-efficient of the cost function. If λ is too small, the sparsity penalty takes little effect and model accuracy is given more priority and if λ is too large, sparsity becomes the priority leading to very sparse models but with unacceptable accuracy. The key is to find the balance between task loss and sparsity penalty.

3.3.3.6 Proposed algorithms

Putting it all together, two algorithms are presented. One uses delta calculations and sparsity penalty concepts with fixed point quantization, and the other uses modified learned step size quantization. The flow charts of the methodology are given in Figures 3.7 and 3.8.

3.3.4 Experiments and Results

The proposed methodology is analysed to study how it helps to achieve the desired temporal sparsity and accuracy.

3.3.4.1 Baseline

For baseline, the two-stream architecture [53] was used with ResNet50 as the feature extractor on both spatial and temporal streams. The dataset used was UCF101, which is a widely used human action recognition dataset of 'in-the-wild' action videos, having 101 action categories [54]. The spatial stream used single-frame RGB images of size (224, 224, 3) as the input, while the temporal stream used stacks of 10 RGB difference frames of size (224, 224, 10 × 3) as the input. Also, both these inputs were time distributed to apply the same layer to multiple frames simultaneously and produce output that has time as the fourth dimension. Both the streams were initialized with pre-trained ImageNet weights and fine-tuned with an SGD optimizer.

Under the above-mentioned setup, spatial and temporal streams achieved an accuracy of 75% and 70%, respectively. Then, both streams were average fused to achieve a final classification accuracy of 82%. Also, in this scenario, both streams were found to have an activation sparsity of about 47%.

3.3.4.2 Experiments

Scenario 1: The setup consecutively places the fixed-point based quantization layer and temporal delta layer after every activation layer in the network. The temporal delta layer here also includes a l_1 norm-based penalty. Fixed point quantization, in this setup, is used to decrease the precision of input activation maps. Both techniques promote temporal sparsity

The baseline weights were used as a starting point, and all the layers including the temporal delta layer is fine-tuned until acceptable convergence.

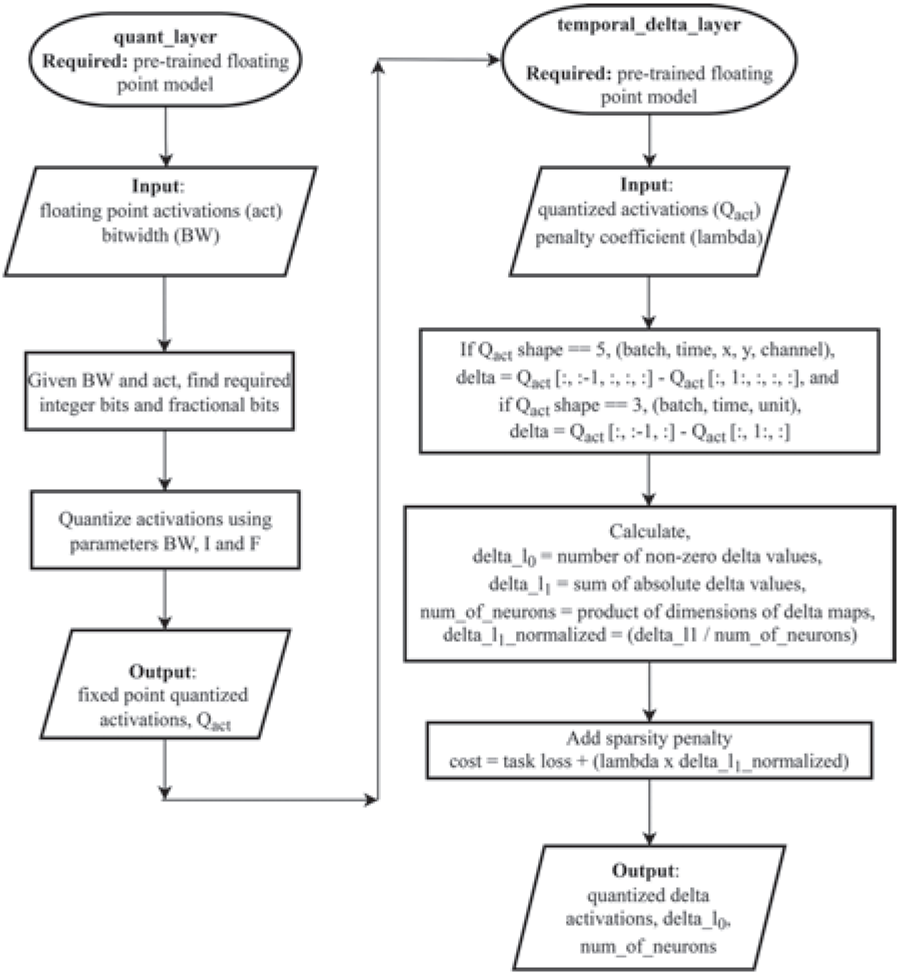

quant_layer
Required: pre-trained floating point model
Input: floating point activations (act) bitwidth (BW)
Given BW and act, find required integer bits and fractional bits
Quantize activations using parameters BW, I and F
Output: fixed point quantized activations, Q_{act}
temporal_delta_layer
Required: pre-trained floating point model
Input: quantized activations (Q_{act}) penalty coefficient (lambda)
If Q_{act} shape == 5, (batch, time, x, y, channel), delta = Q_{act} [:, :-1, :, :, :] - Q_{act} [:, 1:, :, :, :], and if Q_{act} shape == 3, (batch, time, unit), delta = Q_{act} [:, :-1, :] - Q_{act} [:, 1:, :]
Calculate, delta_l_0 = number of non-zero delta values, delta_l_1 = sum of absolute delta values, num_of_neurons = product of dimensions of delta maps, delta_l_1_normalized = (delta_l1 / num_of_neurons)
Add sparsity penalty cost = task loss + (lambda x delta_l_1_normalized)
Output: quantized delta activations, delta_l_0, num_of_neurons

Figure 3.7 Methodology flow of temporal delta layer with fixed point quantization.

The hyper-parameters specifically required for this setup were bit width (to which the activations were to be quantized) and penalty co-efficient to balance the tussle between task loss and sparsity penalty

Scenario 2: The setup is like the previous scenario except for the activation quantization method used. The previous experiment used fixed precision quantization where all the activation layers in the network were quantized to the same bit width. However, this experiment uses learnable step-size quantization (LSQ), which performs channel-wise quantization depending

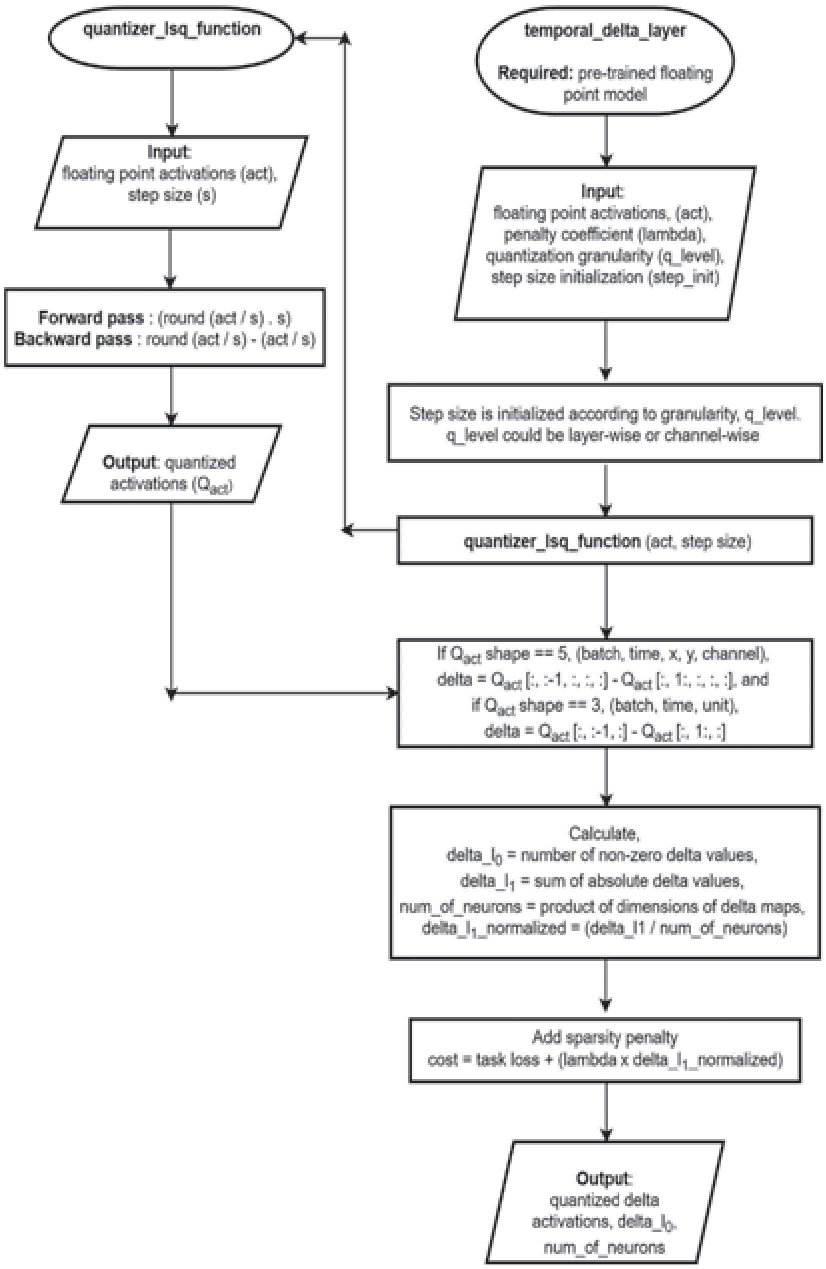

quantizer_lsq_function
Input:
floating point activations (act),
step size (s)
Forward pass : (round (act / s) . s)
Backward pass : round (act / s) - (act / s)
Output: quantized activations (Q_{act})
temporal_delta_layer
Required: pre-trained floating point model
Input:
floating point activations, (act),
penalty coefficient (lambda),
quantization granularity (q_level),
step size initialization (step_init)
Step size is initialized according to granularity, q_level.
q_level could be layer-wise or channel-wise
quantizer_lsq_function (act, step size)
If Q_{act} shape == 5, (batch, time, x, y, channel),
delta = Q_{act} [:, :-1, :, :, :] - Q_{act} [:, 1:, :, :, :], and
if Q_{act} shape == 3, (batch, time, unit),
delta = Q_{act} [:, :-1, :] - Q_{act} [:, 1:, :]
Calculate,
delta_l_0 = number of non-zero delta values,
delta_l_1 = sum of absolute delta values,
num_of_neurons = product of dimensions of delta maps,
delta_l_1_normalized = (delta_l1 / num_of_neurons)
Add sparsity penalty
cost = task loss + (lambda x delta_l_1_normalized)
Output:
quantized delta activations, delta_l_0,
num_of_neurons

Figure 3.8 Methodology flow of temporal delta layer with learned step size quantization.

on the activation distribution resulting in mixed-precision quantization of the activation maps. The layer also introduces a hyperparameter during training (apart from the penalty coefficient mentioned earlier) for the step size initialization. Then, during training, the step size increases or decreases depending on the activation distribution in each channel.

3.3.4.3 Accuracy v/s Activation sparsity

Tables 3.1 and 3.2 show the baseline accuracy and activation sparsity compared against the two scenarios mentioned.

Firstly, when the temporal delta layers with fixed point quantized activations are included in the baseline model, it can be observed that the activation sparsity increases considerably with a slight loss in accuracy. One interesting observation is that, although the sparsity penalty in the temporal delta layer does a good job of decreasing the activation density, quantizing the activations from floating to fixed point representation pushes the activation sparsity of the model even higher. This is because lowering the precision from 32 bits to 8 bits (or less) leads to temporal differences of activations going to absolute zero.

Additionally, the reason for close-to baseline accuracy in the method involving fixed point quantization can be attributed to fractional bit allocation flexibility. That is, as the bit width is fixed, the number of integer bits required is decided depending on the activation distribution within the layer, and the rest of the bits are assigned as fractional bits. This makes sure that the precision of the activation is compromised for range.

Also, another contributing factor for accuracy sustenance is that the first and the last layers of the model are not quantized, similar to works like [55][56][57]. This is because the first and last layer has a lot of information density. Those are the layers where input pixels turn into features and features turn into output probabilities, respectively, which makes them more sensitive to quantization.

Table 3.1 Spatial stream - comparison of accuracy and activation sparsity obtained through the proposed scenarios against the benchmark. In the case of fixed-point quantization, the reported results are for a bit width of 6 bits.

Model setup (Spatial stream)	Accuracy	Activation sparsity
Baseline	75%	48%
Temporal delta layer with fixed point quantization	73%	74%
Temporal delta layer with learned step-size quantization	**69%**	**86%**

Table 3.2 Temporal stream - comparison of accuracy and activation sparsity obtained through the proposed scenarios against the benchmark. In the case of fixed-point quantization, the reported results are for a bit-width of 7 bits.

Model setup (Temporal stream)	**Accuracy**	**Activation sparsity**
Baseline	70%	47%
Temporal delta layer with fixed point quantization	68%	67%
Temporal delta layer with learned step-size quantization	**65%**	**89%**

Although the activation sparsity gain in the case of the temporal delta layer with fixed point quantization is better than the baseline, it is still not sufficiently high as required. In this effort, the bit-width of the activations are decreased in the expectation of increasing sparsity. However, as the bit-width goes below a certain value (6 bits for spatial and 7 bits for temporal stream), sparsity increases, but accuracy starts to deteriorate beyond recovery, as shown in Table 3.3. This is because quantizing all layers of a network to the same bit-width can mean that the inter-channel variations of the feature maps are not fully accounted for. Since the number of fractional bits is usually selected to cover the maximum activation value in a layer, the fixed bit-width quantization tends to cause excessive information loss in channels with a smaller dynamic range. Therefore, it can be inferred that mixed-precision quantization of activations is a better approach to obtain good sparsity without compromising accuracy.

Finally, using the temporal delta layer where incoming activations are quantized using learnable step-size quantization (LSQ) gives the best results for both spatial and temporal streams. As the step size is a learnable parameter, it gives the model enough flexibility to result in a mixed precision

Table 3.3 Result of decreasing activation bit-width to increase activation sparsity while maintaining accuracy. For spatial stream, decreasing below 6 bits caused the accuracy to drop considerably. For temporal stream, the same happened below 7 bits.

	Spatial stream		**Temporal stream**	
Activation bit-width	Accuracy (%)	Activation sparsity (%)	Accuracy (%)	Activation sparsity (%)
32	75	50	70	47
8	75	68	70	65
7	75	71	**68**	**70**
6	**73**	**75**	61	73
5	65	80	-	-

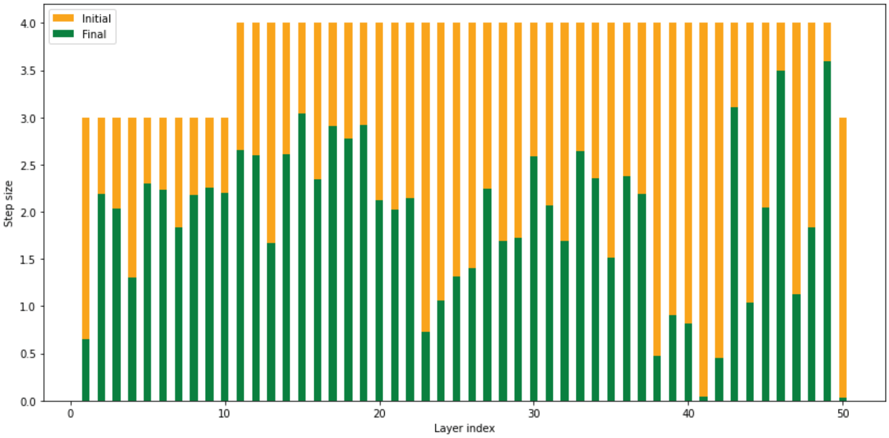


Figure 3.9 Evolution of step size from initialization to convergence. As step-size is a learnable parameter, it gets re-adjusted during training to cause minimum information loss in each layer.

model, where each channel in a layer has a bit-width that suits its activation distribution. This kind of channel-wise quantization minimizes the impact of low-precision rounding.

It is also evident in Figure 3.9 that as the training nears convergence, the values of the step size differ according to the activation distribution and bit width required to represent each layer. Moreover, consistent with the literature [58], the first and last layers during training opts for smaller step sizes implying they need more bandwidth for their representation.

The weights generated using this method was then average fused to find the final two-stream network accuracy and activation sparsity (Table 3.3). Finally, the proposed method can achieve 88% activation sparsity with a 5% accuracy loss.

3.4 NN Compiler for Dedicated Inference Accelerator Hardware with Analog In-Memory Computing

This section explains the role of NN compilers in inference hardware accelerator as well as the methodology to follow to implement and evaluate one. To map the tasks of an NN algorithm to a dedicated hardware with analog in-memory computing for inference, a compiler is needed to automatically generate the instruction set that would provide better performance on the dedicated hardware. The input of such compiler is the trained NN algorithm as well as the hardware architecture of the inference accelerator while the output is the executable set of operations. Therefore, the NN algorithm can only have the type of layers supported on the hardware.

NNs consist of large amounts of Multiplication-and-Accumulation (MAC) operations. Therefore, analog in-memory computation accelerators are ideal to perform such operations. However, different sizes, shapes and bit resolutions make challenging to map NNs on analog crossbar arrays. Nowadays multi-core analog accelerators are becoming very popular for NN inference [64][65]. Each accelerator core consists of an analog crossbar array surrounded by ADCs, DACs and digital logic (FSMs, etc.). Multiple processing cores are connected via NoC (Network-on-Chip), which is used to transfer data between processing cores. Moreover, analog crossbar arrays are not fixed anymore. State-of-the-art crossbar arrays consist of small crossbar elements which can be horizontally or vertically concatenated using programmable switches. The programmability of crossbar makes them very flexible. The crossbar performs multiplications, and the digital logic configures the crossbar switches and controls data flow (weights and activations, etc.) to the crossbar. However, the flexible architecture and constraints of modern analog accelerators pose a challenge in terms of NN workload, mapping, and scheduling. The traditional mapping techniques such as loop tiling, and loop interchange are not efficient anymore [66]. Moreover, the digital FSMs controlling the crossbar and NoC also require a complex instruction set. To the best of our knowledge, there are no commercial compilers available which can be used to map NN workloads and generate the instruction set for flexible and multi-core analog in-memory accelerators.

3.4.1 Compiler Components

The compiler consists of three main components: hardware architecture, parser, and mapper, see Figure 3.10. The unique architecture and constrains of the dedicated inference accelerators require the compiler to consider hardware specifications and constraints while mapping NN workloads. Therefore, the compiler generates a hardware representation of the accelerator using the specifications. The compiler also contains a parser that parses the information of each NN node and converts it into a specific data structure. By using the hardware representation and the parsed NN the mapper generates a mapping in the form of instructions for the FSMs.

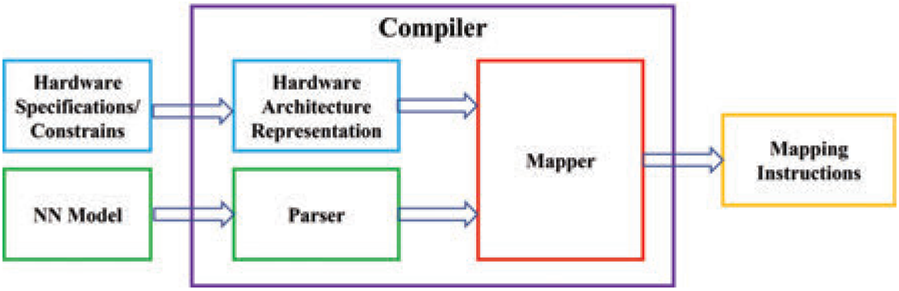

Compiler
Hardware Specifications/ Constrains
NN Model
Hardware Architecture Representation
Parser
Mapper
Mapping Instructions

Figure 3.10 Overview of Compiler Tool.

3.4.2 ONNX Parser

The trained networks are stored using ONNX with a custom export to store additional information such as quantization. The custom ONNX file is parsed by a custom ONNX Parser to extract the information needed by the mapper. This parser parses every graph node of the ONNX model and creates a Python data structure to ease access to information and attributes. The parsed model is a list of nodes and each one stores relevant information. The list of nodes includes:

- Input contains information about the input layer
- Conv contains information regarding convolution layers
- MatMul contains the parameters of fully connected layers
- Add is pointwise addition
- Mul is pointwise multiplication
- Div is a pointwise division
- Act contains activation functions
- Squeeze and Unsqueeze to remove or add singleton dimension when needed.

Moreover, the parser allows the user to fuse the information of consecutive nodes if these nodes follow a certain pattern, e.g., Conv layer followed by one or more of Add, Mul, Div or Act layers are fused together, as shown in Figure 3.11. The fused information is stored in the parsed fused model. The layers are combined when their computations can/should be carried out by same processing core in one computing cycle to minimize data exchange. The parsed fused model is used by the compiler to pre-process and generate instructions to run the NN on the hardware.

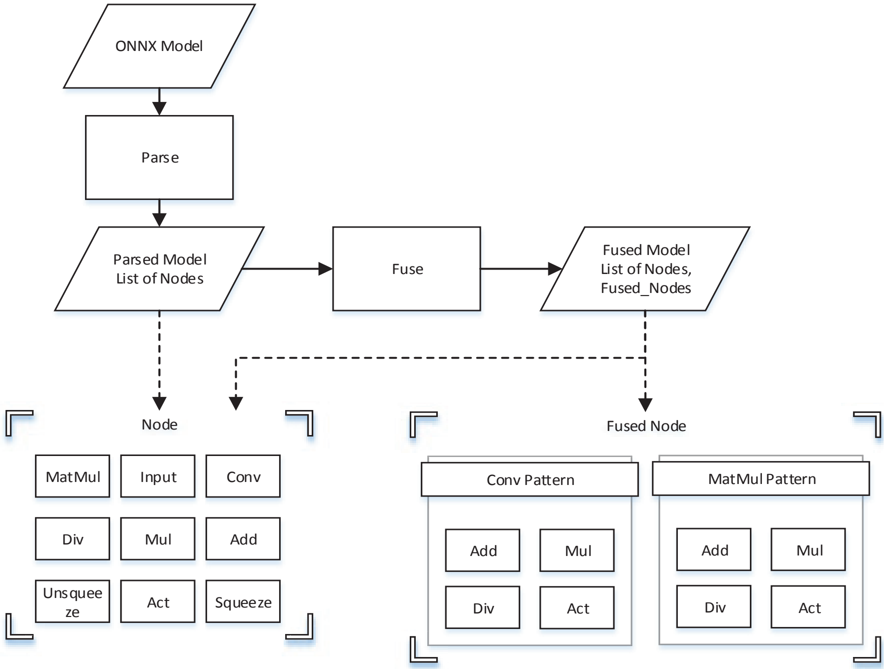


Figure 3.11 ONNX Parser diagram of parsing and fusing the input ONNX model into a list of Nodes and Fused Nodes.

3.4.3 Hardware Architecture Representation

The hardware architecture representation component allows the compiler to support arbitrary number of processing cores and crossbar array configurations. It generates blueprint of available computation resources for each core according to the configurations and constraints.

The input parameters include the hardware specifications and constrains like crossbar specifications, number of processing cores, memory sizes, etc. When the architectural parser is invoked, it generates a blueprint of the FSMs, the processing cores and the NoC controller. The NoC controller is responsible for data transfer between processing cores and the FSMs for the storage of the weights on the crossbar array, providing the input data to the crossbar array according to the input specifications of each layer and handling the output results of the crossbar array.

3.4.4 Mapper

The mapper analyses the parsed NN and available computation resources on the hardware to map the NN workload into the processing cores. Figure 3.12 shows the flow of the mapper.

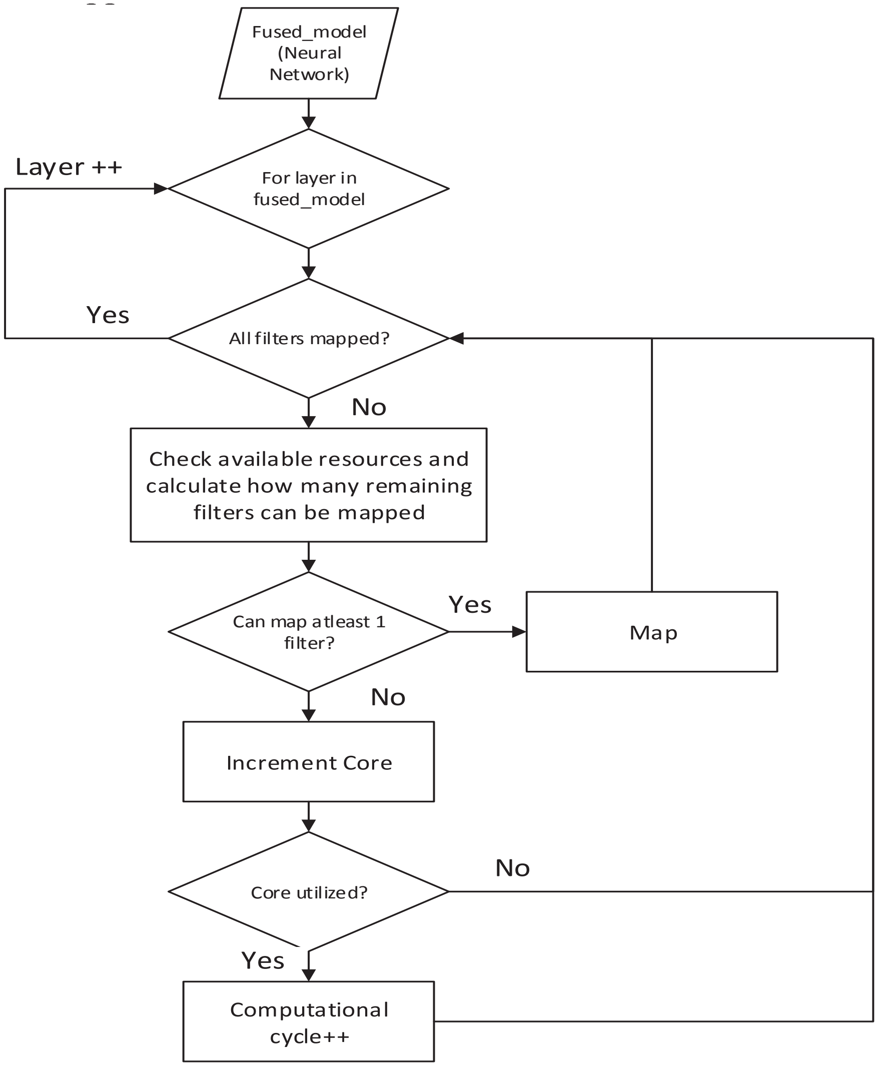

Fused_model (Neural Network)
For layer in fused_model
Layer ++
All filters mapped?
Yes
No
Check available resources and calculate how many remaining filters can be mapped
Can map atleast 1 filter?
Yes
Map
No
Increment Core
Core utilized?
No
Yes
Computational cycle++

Figure 3.12 Mapping flow of the Compiler.

The mapper maps a NN workload layer by layer. Analog crossbars perform computations using vector-matrix multiplications. Therefore, the compiler converts different NN workloads into vector-matrix multiplications. Moreover, the compiler also schedules the data transfers between processing cores. The weights are converted into a matrix representation and stored in crossbar memory columns and inputs are converted into vectors and fed to crossbar rows using DACs. The mapper determines the required hardware resources for every layer and then it checks how many filters of a layer can be mapped on the current processing core. If there is enough space to map all filters of a layer, then all filters are mapped to the current processing core. Otherwise, a layer is partially mapped to a processing core and remaining filters are mapped to the next core. Similarly, when all processing cores are utilized, the mapper starts mapping again from the first processing core for the next computation. The mapper keeps increasing the computation cycle until the whole NN workload is mapped. When mapper maps the NN workload to a part of processing core, that part is marked as utilized and the parameter values in all FSMs are set according to the mapping. After mapping a workload to each processing core, the compiler generates instructions sets that can be decoded on the dedicated hardware to generate sets of parameters, which are then used to configure the processing cores.

3.4.5 Mapping Strategy

The crossbar array of a processing core is composed of multiple rows and columns of analog synaptic weights performing MAC operations. The crossbar allows to map either fully connected layers or convolutional layers with different kernel sizes.

Figure 3.13 shows the mapping strategy of how a small CNN with two convolution layers has been mapped to a crossbar array of a processing core. Each black rectangle represents 16x4 synaptic weights. The input size is 13x64, the kernel sizes of Conv1 and Conv2 are 4x4x1, with 16 filters, and 3x3x16, with 8 filters, respectively, and the output size is 3x29x8. According to the compiler flow, the compiler checks first for available resources in the processing core and then determines how many synaptic weights are required to map a layer. At the beginning, since the whole processing core is available and the Conv1 filters are small, Conv1 layer can be fully mapped. The filters of Conv1 are converted into vectors and mapped to crossbar like a matrix. The Conv1 filters’ vectors are small, and each filter requires 4x4x16 synaptic weights. Similarly, filters of Conv2 are also converted to a vector and mapped.

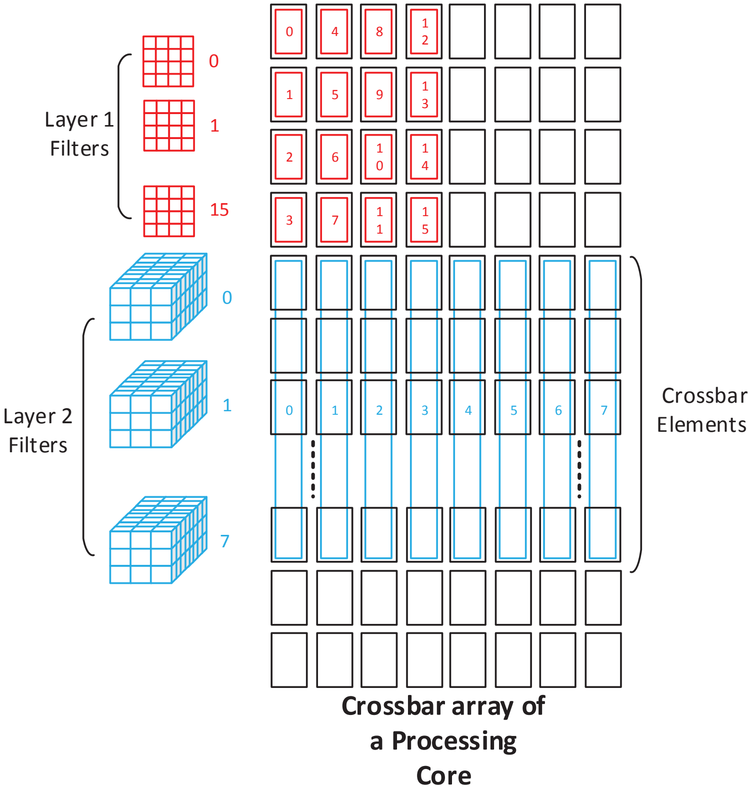


Figure 3.13 Mapping of layers 1 and 2 on processing core 1.

Since Conv2 layer consists of large filters, it requires 3x3x16x8 synaptic weights. The mapper will evaluate if enough synaptic weights are available in the same processing core at which Conv1 was mapped. If it is not the case, a new processing core will be used for mapping Conv2 layer.

3.4.6 Mapping of Deep Spiking NN Architectures to Digital SNN Inference Devices

This section presents an approach for mapping arbitrary deep SNN network architectures onto fixed-architecture inference devices. As example, a device is considered with a single population of hidden neurons, supporting a fixed number of synaptic inputs per neuron, and with a limited number of input and output neurons (see Figure 3.14).

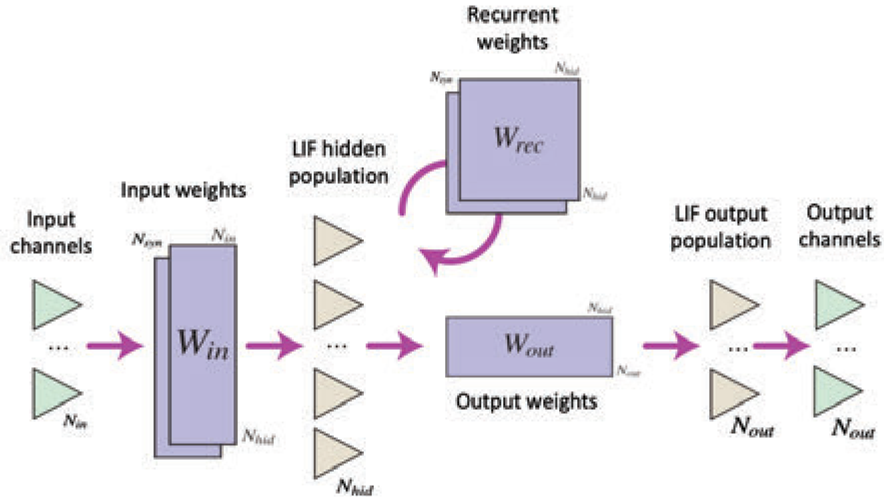


Figure 3.14 A HW architecture for SNN inference.

This architecture supports a fixed maximum number of input channels N_{in}; a fixed maximum number of hidden neurons N_{hid}; and a fixed maximum number of output neurons N_{out}. The neurons are LIF spiking neurons, supporting several synaptic inputs N_{syn}. This architecture implements a single hidden population with the possibility of recurrent weights W_{rec}. Multi-layer networks must be mapped into this single hidden population using the recurrent weights. Other logical weight blocks are supported for input weights W_{in} and output weights W_{out}. These weight matrices are assumed to be sparse, with a limited maximum fan-in N_F per neuron. Only on-zero weights are stored, with weights linearised into memory blocks of fixed maximum size $N_F * N$. Figure redrawn from [82].

A mapping system must be flexibly to accommodate a wide range of SNN network architectures, including recurrent spiking populations (e.g., reservoir networks and other recurrent architectures; deep feed-forward architectures; and residual network architectures. An example of a deep spiking network making use of all these architectural elements is shown in Figure 3.14. Several LIF spiking neuron layers ("LIF"; orange) are connected via weight blocks ("W"; blue). The first LIF layer "LIF$_1$" is recurrently connected with weights "W$_{rec}$". Residual blocks (dashed) include additional connections bypassing the blocks inside.

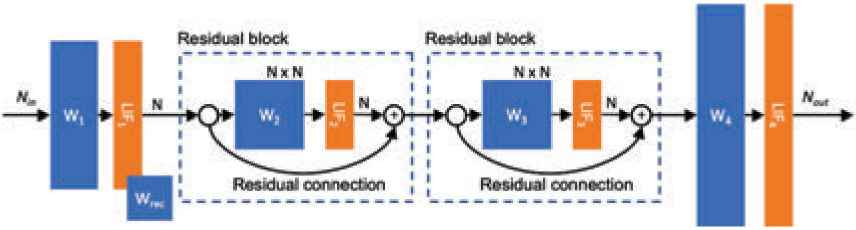


Figure 3.15 An example of a deep spiking network that will be mapped to a HW architecture.

To map the network onto the HW architecture shown in Figure 3.15, several steps are taken.

1. *Graph extraction:* Convert the simulation modules from a high-level representation to a standardised set of graph modules. These graph modules individually represent the weights and neuron populations in the network, as well as embodying a traversable graph that accurately corresponds to the computations and information flow in the network.
2. *DRC check:* Perform a design-rule-check to ensure that the network is compatible with the hardware architecture.
3. *HW mapping:* Assign the network logical resources to available hardware resources.
4. *Parameter configuration:* Assign network parameters to appropriate HW memory blocks and serialise to a bitstream to configure the HW.

3.5 Simulator/Profiler

When implementing a neural network topology on an embedded hardware target, it is critical to do it being able to profile that network and to simulate it, by considering the hardware architecture for extracting the right power/performance/latency figures. Profiling a network means extracting key parameters of interest when fed with representative data. This obviously relates to the number of parameters and number of operations, which is readily available for a given topology, but not only. To choose the right hardware target and optimize the mapping of the network or its graph transformation, the data volume and data bandwidth per layer are also needed. Counter intuitively, the highest data bandwidth does not occur in the layers with the higher number of parameters. This is illustrated in the figures below, considering a popular network topology for embedded applications, i.e., a MobileNet V1.

Figure 3.16 depicts the number of parameters per layer: the deeper the layer, the higher the number of parameters. The depth wise convolution layers use less parameters than the pointwise convolution layers since they use 2D filters instead of 3D ones.

But, as can be seen in Figure 3.17, the first layers are the ones having the biggest data volume. This is because there are more parameters reused on those first layers.

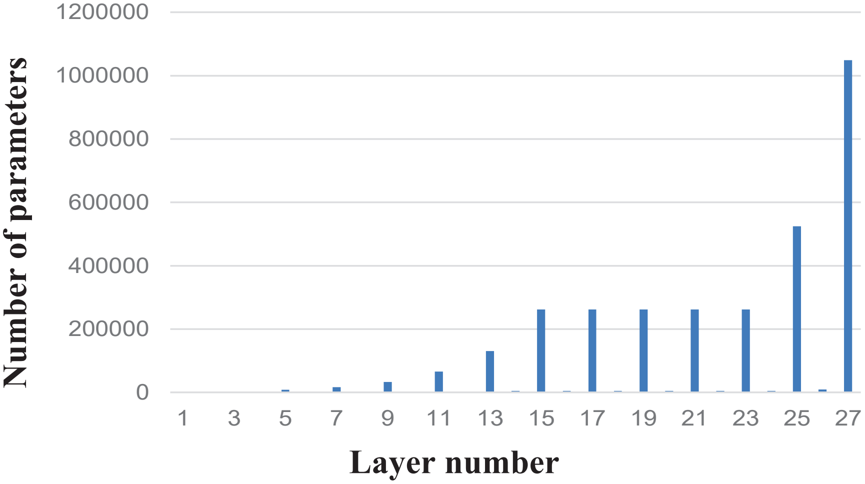


Figure 3.16 MobileNet V1 parameters per layer.

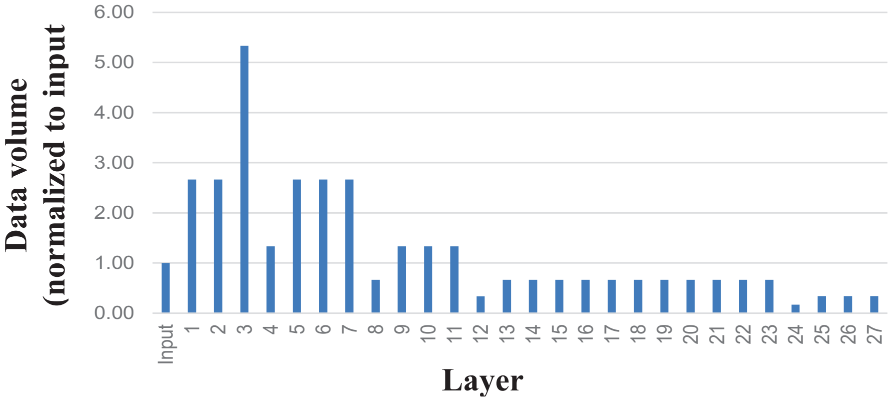


Figure 3.17 MobileNet V1 data volume per layer, normalized to input data volume.

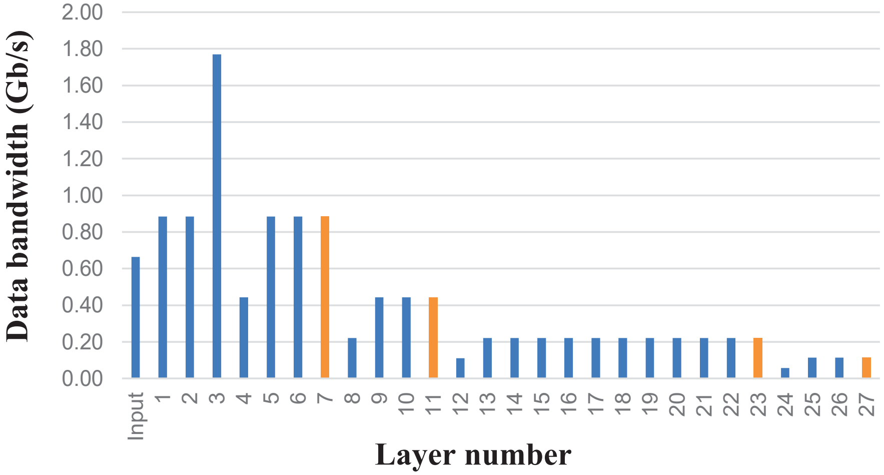


Figure 3.18 MobileNet V1 bandwidth (Gb/s) at each layer.

This profiling shows that it is more important to keep the data locally for layers 1 to 12 to minimize data movement: thus, a data-flow architecture for those layers might be a good fit.

Figure 3.18 also shows the impact of stride on data volume reduction. For instance, layers 1, 4, 8 and 24 have a stride of 2.

The data bandwidth is obviously proportional to the size of the input image and number of images per second. Figure 3.20 illustrates the bandwidth for a 30FPS, 1280x720p, 8b RGB input. The intermediate layers use 4b activations in this case.

Depending on the application, i.e., segmentation, detection, classification, different layers can be exploited. Those layers are highlighted in orange. For object detection, the most important layers are number 27 and 23. For segmentation, layers number 7 and 11 can be exploited, with layer 7 having a higher bandwidth and thus a higher definition.

To obtain those figures, the N2D2 [67][68] (Neural Network Design & Deployment) dedicated framework is used in ANDANTE for deploying neural networks on digital hardware targets, see Figure 3.19.

For optimizing a network, N2D2 considers applicative performance metrics to be achieved and the hardware target memory capacity. It exploits sparsity of weights by implementing state-of-the-art quantization-aware training methods, such as SAT and LSQ. Finally, N2D2 can address several hardware targets, generating bit streams or configuration files, but it can also be used

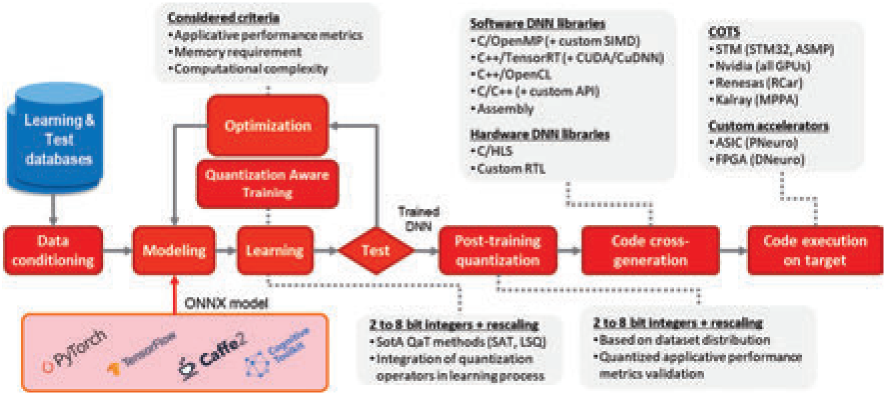

Considered criteria
•Applicative performance metrics
•Memory requirement
•Computational complexity
Software DNN libraries
•C/OpenMP (+ custom SIMD)
•C++/TensorRT (+ CUDA/CuDNN)
•C++/OpenCL
•C/C++ (+ custom API)
•Assembly
Hardware DNN libraries
•C/HLS
•Custom RTL
COTS
•STM (STM32, ASMP)
•Nvidia (all GPUs)
•Renesas (RCar)
•Kalray (MPPA)
Custom accelerators
•ASIC (PNeuro)
•FPGA (DNeuro)
Learning & Test databases
Optimization
Quantization Aware Training
Trained DNN
Data conditioning
Modeling
Learning
Test
Post-training quantization
Code cross-generation
Code execution on target
ONNX model
PyTorch
TensorFlow
Caffe2
Cognitive Toolkit
2 to 8 bit integers + rescaling
•SotA QaT methods (SAT, LSQ)
•Integration of quantization operators in learning process
2 to 8 bit integers + rescaling
•Based on dataset distribution
•Quantized applicative performance metrics validation

Figure 3.19 N2D2: Neural Network Design & Deployment.

for driving architecture exploration. In such a case, a graph transformation is necessary for accurately performing the simulation and obtaining the Key Parameters of Interest.

Figure 3.20 shows the three steps needed for converting a topology to target a pipelined DNN architecture:

1. The selection of the pre-templated architecture to be used for implementing each layer type (sub-steps a and b in the figure below).

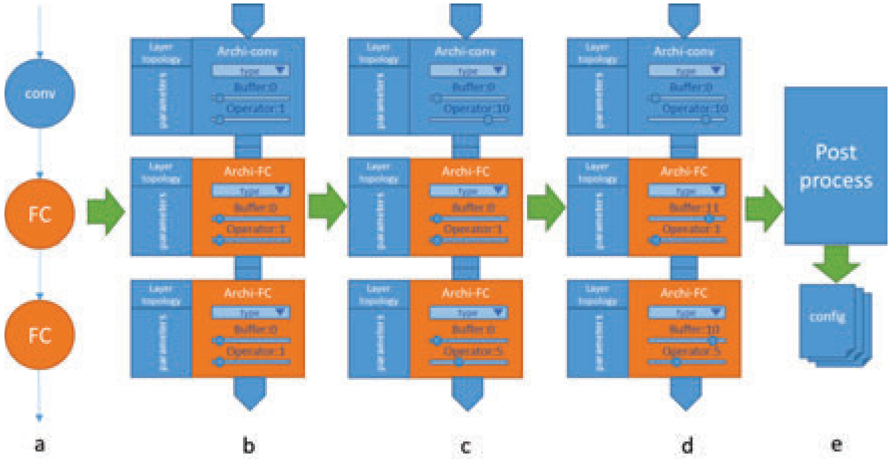


Figure 3.20 Process flow: (a,b) conversion of the neural network to the hardware representation, (c) tuning of the layer parallelism at architectural level, (d) tuning of the buffer, (e) post-processing.

2. The configuration of the resources used, in each architecture, in terms of parallelism and buffer (sub-steps c and d).
3. The network parameters post-process (sub-step e).

This enables to tune various architecture parameters, such as the parallelism of each layer, the buffer, etc. and assess their impact.

The first two steps, architecture selection and hardware resource configuration, are performed using a ROI algorithm. Each layer of the network is associated with a hardware architecture corresponding to the type of operation carried out in the layer. Several architectural models can be considered for a given layer type; in which case the best suited architecture is selected for that layer.

At the end of these steps, a graph of configured architectures is obtained, that ensures the smooth running of the calculation throughout the pipeline. Those "hardware-aware" simulations prove to be much more accurate in terms of energy efficiency and latency, while being bit-equivalent to high level simulations.

3.6 Conclusions

3.6.1 On NN Model Transformation

Intuitively, the new temporal delta layer [63] casts the temporal activation sparsity between two consecutive feature maps into spatial activation sparsity of their delta map. This spatial sparsity is then exploited to reduce computations and memory access when performing sparse tensor multiplications in hardware. As shown in 3.4 proposed method resulted in 88% activation sparsity with an accuracy drop of 5% on UCF-101 dataset for human action recognition.

Table 3.4 Final results on 2 stream networks after average fusing the spatial and temporal stream weights. With 5% accuracy loss, the proposed method almost doubles the activation sparsity available in comparison to the baseline

	Baseline		**Proposed method**	
Model type	Accuracy (%)	Activation sparsity (%)	Accuracy (%)	Activation sparsity (%)
Spatial stream	75	50	69	86
Temporal stream	70	46	65	89
Two-stream (Average fused)	82	47	**77**	**88**

The collateral advantage of temporal sparsity is that the computations does not increase linearly with the increase in frame rate. In standard DNN, doubling the frame rate naturally would require double the computations. However, in the case of temporal delta layer-based model, increasing the frame rate would not only increase the temporal precision of the network but also increase the temporal sparsity limiting the computations required [59].

The drawback of using temporal delta layer derives from its requirement to keep track of the previous activations to perform delta operations. This increases the overall memory footprint which in turn increases the reliance on off-chip memory. For instance, external DRAM memory consumes two orders of magnitude more energy than SRAM [60]. However, the increasing popularity of new memory technologies (like resistive RAM [61], embedded Flash memory [62], etc.) may improve the cost calculations in the near future.

3.6.2 On NN Compiler for Dedicated Inference Accelerator Hardware with Analog In-Memory Computing Conclusion

The main objectives for a compiler tool are maximize hardware utilization, maximize throughput, and minimize latency. However, there is always a trade-off between these objectives since not all of them can be achieved at the same time. Therefore, in the particular methodology described in Section 4 maximum utilization of hardware resources is the focus. The mapping algorithm checks the resources needed for allocating each layer of the NN. Afterwards checks for the available resources on the hardware and tries to find the optimum mapping to fully utilize each processing core.

3.6.3 Simulator/Profiler

Profiling a neural network is essential when considering deploying it on an embedded hardware target. Indeed, for choosing the right target and correctly mapping the network on it, one obviously needs to know the number of parameters (for the memory footprint), the number of operations (for the number of processing elements in a latency-constrained implementation) but also the data bandwidth and data volume. Such a simulation/profiling environment is developed and used in the ANDANTE project. It allows to assess the impact of quantization (even mixed quantization depending on the layers), to identify the most adequate layers for e.g., object detection or image segmentation (in terms of data bandwidth). It can also guide architecture exploration, with a mix of spatially expanded and spatially folded architecture, and the needed graph transformations.

Acknowledgements

This work is supported through the project ANDANTE. ANDANTE has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No. 876925. The JU receives support from the European Union's Horizon 2020 research and innovation programme and France, Belgium, Germany, Netherlands, Portugal, Spain, Switzerland. ANDANTE has also received funding from the German Federal Ministry of Education and Research (BMBF) under Grant No. 16MEE0116 and 16MEE0111K. The authors are responsible for the content of this publication.

References

[1] ANDANTE. AI for New Devices and Technologies at the Edge. Available online at: https://www.andante-ai.eu/

[2] M. Capra, et al. "Hardware and software optimizations for accelerating deep neural networks: Survey of current trends, challenges, and the road ahead." *IEEE Access* 8 (2020): 225134-225180.

[3] T. Liang, J. Glossner, L.Wang, and S. Shi, "Pruning and quantization for deep neural network acceleration: A survey," arXiv preprint arXiv:2101.09671, 2021.

[4] T. Hoefler, D. Alistarh, T. Ben-Nun, N. Dryden, and A. Peste, "Sparsity in deep learning: Pruning and growth for efficient inference and training in neural networks," arXiv preprint arXiv:2102.00554, 2021.

[5] J. Gou, B. Yu, S. J. Maybank, and D. Tao, "Knowledge distillation: A survey," International Journal of Computer Vision, vol. 129, no. 6, pp. 1789–1819, 2021.

[6] S. Han, X. Liu, H. Mao, J. Pu, A. Pedram, M. A. Horowitz, and W. J Dally, "Eie: Efficient inference engine on compressed deep neural network," ACM SIGARCH Computer Architecture News, vol. 44, no. 3, pp. 243–254, 2016.

[7] A. Parashar, M. Rhu, A. Mukkara, A. Puglielli, R. Venkatesan, B. Khailany, J. Emer, S. W. Keckler, and W. J. Dally, "Scnn: An accelerator for compressed-sparse convolutional neural networks," ACM SIGARCH Computer Architecture News, vol. 45, no. 2, pp. 27–40, 2017.

[8] D. Kim, J. Ahn, and S. Yoo, "Zena: Zero-aware neural network accelerator," IEEE Design & Test, vol. 35, no. 1, pp. 39–46, 2017.

[9] E. O. Neftci, H. Mostafa, and F. Zenke, "Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks," IEEE Signal Processing Magazine, vol. 36, no. 6, pp. 51–63, 2019.

[10] M. Mirsadeghi, M. Shalchian, S. R. Kheradpisheh, and T. Masquelier, "Stidi-bp: Spike time displacement based error backpropagation in multilayer spiking neural networks," Neurocomputing, vol. 427, pp. 131–140, 2021.

[11] B. Rueckauer, I.-A. Lungu, Y. Hu, M. Pfeiffer, and S.-C. Liu, "Conversion of continuous-valued deep networks to efficient event driven networks for image classification," Frontiers in neuroscience, vol. 11, pp. 682, 2017.

[12] M. Sorbaro, Q. Liu, M. Bortone, and S. Sheik, "Optimizing the energy consumption of spiking neural networks for neuromorphic applications," Frontiers in neuroscience, vol. 14, pp. 662, 2020.

[13] C. Louizos, M. Welling, and D. P. Kingma, "Learning sparse neural networks through l 0 regularization," arXiv preprint arXiv:1712.01312, 2017.

[14] Z. Liu, J. Li, Z. Shen, G. Huang, S.g Yan, and C. Zhang, "Learning efficient convolutional networks through network slimming," in Proceedings of the IEEE international conference on computer vision, 2017, pp. 2736–2744.

[15] M. Kurtz, J. Kopinsky, R. Gelashvili, A. Matveev, J. Carr, M. Goin, W. Leiserson, S. Moore, N. Shavit, and D. Alistarh, "Inducing and exploiting activation sparsity for fast inference on deep neural networks," in International Conference on Machine Learning. PMLR, 2020, pp. 5533–5543.

[16] G. Georgiadis, "Accelerating convolutional neural networks via activation map compression," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 7085–7095.

[17] A. Yousefzadeh and M. Sifalakis, "Training for temporal sparsity in deep neural networks, application in video processing," arXiv preprint arXiv:2107.07305, 2021.

[18] A. Yousefzadeh, M. A. Khoei, S.r Hosseini, P. Holanda, S. Leroux, O. Moreira, J. Tapson, B. Dhoedt, P. Simoens, T. Serrano-Gotarredona, and B. Linares-Barranco, "Asynchronous spiking neurons, the natural key to exploit temporal sparsity," IEEE Journal on Emerging and Selected Topics in Circuits and Systems, vol. 9, no. 4, pp. 668–678, 2019.

[19] J. Zhao, J. Yang, J. Wang, and W. Wu, "Spiking neural network regularization with fixed and adaptive drop-keep probabilities," IEEE Transactions on Neural Networks and Learning Systems, 2021.

[20] T. Pellegrini, R. Zimmer, and T. Masquelier, "Low-activity supervised convolutional spiking neural networks applied to speech commands recognition," in 2021 IEEE Spoken Language Technology Workshop (SLT). IEEE, 2021, pp. 97–103.

[21] D. Neil, M. Pfeiffer, and S.-C. Liu, "Learning to be efficient: Algorithms for training low latency, low-compute deep spiking neural networks," in Proceedings of the 31^{st} annual ACM symposium on applied computing, 2016, pp. 293–298.

[22] M. Jerry et al., "Ferroelectric FET analog synapse for acceleration of deep neural network training," 2017 IEEE International Electron Devices Meeting (IEDM), 2017, pp. 6.2.1-6.2.4, doi: 10.1109/IEDM.2017.8268338

[23] S., Dutta, C., Schafer, J., Gomez, K., Ni, S., Joshi, and S. Datta, (2020). Supervised learning in all FeFET-based spiking neural network: Opportunities and challenges. Frontiers in Neuroscience, 14, 634.

[24] S., Dutta, V., Kumar, A. Shukla, et al. "Leaky Integrate and Fire Neuron by Charge-Discharge Dynamics in Floating-Body MOSFET". Sci Rep 7, 8257 (2017).

[25] B. Trevor, et al. Nengo: a Python tool for building large-scale functional brain models, Frontiers in Neuroinformatics , Volume 7, 2014.

[26] M. Arsalan, A. Santra and V. Issakov, "RadarSNN: A Resource Efficient Gesture Sensing System Based on mm-Wave Radar," in IEEE Transactions on Microwave Theory and Techniques, doi: 10.1109/TMTT.2022.3148403.

[27] V. Senft, T. C. Stewart, T. Bekolay, C. Eliasmith, B. J. Kröger, "Reduction of dopamine in basal ganglia and its effects on syllable sequencing in speech": A computer simulation study, Basal Ganglia, Volume 6, Issue 1, 2016.

[28] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool, "Temporal segment networks: Towards good practices for deep action recognition," in European conference on computer vision, pp. 20–36, Springer, 2016.

[29] K. Chen and W. Tao, "Once for all: a two-flow convolutional neural network for visual tracking," IEEE Transactions on Circuits and Systems for Video Technology, vol. 28, no. 12, pp. 3377–3386, 2017.

[30] K. Kang, H. Li, J. Yan, X. Zeng, B. Yang, T. Xiao, C. Zhang, Z. Wang, R. Wang, X. Wang, et al., "T-cnn: Tubelets with convolutional neural networks for object detection from videos," IEEE Transactions on Circuits and Systems for Video Technology, vol. 28, no. 10, pp. 2896–2907, 2017.

[31] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding," arXiv preprint arXiv:1510.00149, 2015.

[32] T. Gale, E. Elsen, and S. Hooker, "The state of sparsity in deep neural networks," arXiv preprint arXiv:1902.09574, 2019.

[33] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," arXiv preprint arXiv:1503.02531, 2015.

[34] W. Wen, C. Wu, Y. Wang, Y. Chen, and H. Li, "Learning structured sparsity in deep neural networks," arXiv preprint arXiv:1608.03665, 2016.

[35] M. Mahowald, "The silicon retina," in An Analog VLSI System for Stereoscopic Vision, pp. 4– 65, Springer, 1994.

[36] J. W. Mink, R. J. Blumenschine, and D. B. Adams, "Ratio of central nervous system to body metabolism in vertebrates: its constancy and functional basis," American Journal of PhysiologyRegulatory, Integrative and Comparative Physiology, vol. 241, no. 3, pp. R203–R212, 1981.

[37] A. Yousefzadeh, M. A. Khoei, S. Hosseini, P. Holanda, S. Leroux, O. Moreira, J. Tapson, B. Dhoedt, P. Simoens, T. Serrano-Gotarredona, et al., "Asynchronous spiking neurons, the natural key to exploit temporal sparsity," IEEE Journal on Emerging and Selected Topics in Circuits and Systems, vol. 9, no. 4, pp. 668–678, 2019.

[38] C. Gao, D. Neil, E. Ceolini, S.-C. Liu, and T. Delbruck, "Deltarnn: A power-efficient recurrent neural network accelerator," in Proceedings of the 2018 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays, pp. 21–30, 2018.

[39] O. Moreira, A. Yousefzadeh, F. Chersi, G. Cinserin, R.-J. Zwartenkot, A. Kapoor, P. Qiao, P. Kievits, M. Khoei, L. Rouillard, et al., "Neuronflow: a neuromorphic processor architecture for live ai applications," in 2020 Design, Automation & Test in Europe Conference & Exhibition (DATE), pp. 840–845, IEEE, 2020.

[40] J. Frankle and M. Carbin, "The lottery ticket hypothesis: Finding sparse, trainable neural networks," arXiv preprint arXiv:1803.03635, 2018.

[41] H. Yang, W. Wen, and H. Li, “Deephoyer: Learning sparser neural network with differentiable scale-invariant sparsity measures,” arXiv preprint arXiv:1908.09979, 2019.

[42] S. Seto, M. T. Wells, and W. Zhang, “Halo: Learning to prune neural networks with shrinkage,” in Proceedings of the 2021 SIAM International Conference on Data Mining (SDM), pp. 558–566, SIAM, 2021.

[43] G. Georgiadis, “Accelerating convolutional neural networks via activation map compression,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7085–7095, 2019.

[44] M. Kurtz, J. Kopinsky, R. Gelashvili, A. Matveev, J. Carr, M. Goin, W. Leiserson, S. Moore, B. Nell, N. Shavit, et al., “Inducing and exploiting activation sparsity for fast neural network inference,” in 37th International Conference on Machine Learning, ICML 2020, vol. 119, 2020.

[45] M. Mahmoud, K. Siu, and A. Moshovos, “Diffy: A dt’ej‘a vu-free differential deep neural network accelerator,” in 2018 51st Annual IEEE/ACM International Symposium on Microarchitecture (MICRO), pp. 134–147, IEEE, 2018.

[46] C.-Y. Wu, M. Zaheer, H. Hu, R. Manmatha, A. J. Smola, and P. Krĺahenbuĺhl, “Compressed video action recognition,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6026–6035, 2018.

[47] M. Buckler, P. Bedoukian, S. Jayasuriya, and A. Sampson, “Eva2: Exploiting temporal redundancy in live computer vision,” in 2018 ACM/IEEE 45th Annual International Symposium on Computer Architecture (ISCA), pp. 533–546, IEEE, 2018.

[48] L. Cavigelli, P. Degen, and L. Benini, “Cbinfer: Change-based inference for convolutional neural networks on video data,” in Proceedings of the 11th International Conference on Distributed Smart Cameras, pp. 1–8, 2017.

[49] P. O’Connor and M. Welling, “Sigma delta quantized networks,” arXiv preprint arXiv:1611.02024, 2016.

[50] P.-E. Novac, G. B. Hacene, A. Pegatoquet, B. Miramond, and V. Gripon, “Quantization and deployment of deep neural networks on microcontrollers,” Sensors, vol. 21, no. 9, p. 2984, 2021.

[51] S. K. Esser, J. L. McKinstry, D. Bablani, R. Appuswamy, and D. S. Modha, “Learned step size quantization,” arXiv preprint arXiv:1902.08153, 2019.

[52] R. Krishnamoorthi, "Quantizing deep convolutional networks for efficient inference: A whitepaper," arXiv preprint arXiv:1806.08342, 2018.

[53] Y. Yu, R. Hira, J. N. Stirman, W. Yu, I. T. Smith, and S. L. Smith, "Mice use robust and common strategies to discriminate natural scenes," Scientific reports, vol. 8, no. 1, pp. 1–13, 2018.

[54] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," arXiv preprint arXiv:1406.2199, 2014.

[55] K. Soomro, A. R. Zamir, and M. Shah, "Ucf101: A dataset of 101 human actions classes from videos in the wild," arXiv preprint arXiv:1212.0402, 2012.

[56] J. Choi, Z. Wang, S. Venkataramani, P. I.-J. Chuang, V. Srinivasan, and K. Gopalakrishnan, "Pact: Parameterized clipping activation for quantized neural networks," arXiv preprint arXiv:1805.06085, 2018.

[57] S. Zhou, Y. Wu, Z. Ni, X. Zhou, H. Wen, and Y. Zou, "Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients," arXiv preprint arXiv:1606.06160, 2016.

[58] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio, "Quantized neural networks: Training neural networks with low precision weights and activations," The Journal of Machine Learning Research, vol. 18, no. 1, pp. 6869–6898, 2017.

[59] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi, "Xnor-net: Imagenet classification using binary convolutional neural networks," in European conference on computer vision, pp. 525–542, Springer, 2016.

[60] M. A. Khoei, A. Yousefzadeh, A. Pourtaherian, O. Moreira, and J. Tapson, "Sparnet: Sparse asynchronous neural network execution for energy efficient inference," in 2020 2nd IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS), pp. 256–260, IEEE, 2020.

[61] V. Sze, Y.-H. Chen, T.-J. Yang, and J. S. Emer, "Efficient processing of deep neural networks: A tutorial and survey," Proceedings of the IEEE, vol. 105, no. 12, pp. 2295–2329, 2017.

[62] S. Huang, A. Ankit, P. Silveira, R. Antunes, S. R. Chalamalasetti, I. El Hajj, D. E. Kim, G. Aguiar, P. Bruel, S. Serebryakov, et al., "Mixed precision quantization for reram-based dnn inference accelerators," in 2021 26th Asia and South Pacific Design Automation Conference (ASP-DAC), pp. 372–377, IEEE, 2021.

[63] M. Kang, H. Kim, H. Shin, J. Sim, K. Kim, and L.-S. Kim, "S-flash: A nand flash-based deep neural network accelerator exploiting bit-level sparsity," IEEE Transactions on Computers, 2021.

[64] P. Vijayan, "Temporal Delta Layer." Available online at: http://resolver.tudelft.nl/uuid:0806241d-9037-4094-a197-6e65d6482f2b.

[65] C. Yakopcic, T. M. Taha and R. Hasan, "Hybrid crossbar architecture for a memristor based memory," NAECON 2014 - IEEE National Aerospace and Electronics Conference, 2014, pp. 237-242. doi: 10.1109/NAECON.2014.7045809.

[66] X. Wang et al., "TAICHI: A Tiled Architecture for In-Memory Computing and Heterogeneous Integration," in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 69, no. 2, pp. 559-563, Feb. 2022. doi: 10.1109/TCSII.2021.3097035.

[67] A. Parashar et al., "Timeloop: A Systematic Approach to DNN Accelerator Evaluation," 2019 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS), 2019, pp. 304-315. doi: 10.1109/ISPASS.2019.00042.

[68] C. Gamrat "Neural Networks Design and Deployment N2D2 for embedded AI" www.youtube

[69] CEA-LIST/N2D2 -GitHub, github.com > CEA-LIST > N2D2

[70] H., Cai, C., Gan, T., Wang, Z., Zhang, and S. Han, (2019). Once-for-all: Train one network and specialize it for efficient deployment. arXiv preprint arXiv:1908.09791

[71] J. A., Pérez-Carrasco, B., Zhao, C., Serrano, B., Acha, T., Serrano-Gotarredona, S., Chen, and B. Linares-Barranco (2013). Mapping from frame-driven to frame-free event-driven vision systems by low-rate rate coding and coincidence processing–application to feedforward ConvNets. IEEE transactions on pattern analysis and machine intelligence, 35(11), 2706-2719.

[72] P. U., Diehl, D., Neil, J., Binas, M., Cook, S. C., Liu, and M. Pfeiffer (2015, July). Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing. In 2015 International joint conference on neural networks (IJCNN), pp. 1-8.

[73] D., Zambrano, and S. M. Bohte (2016). Fast and efficient asynchronous neural computation with adapting spiking neural networks. arXiv preprint arXiv:1609.02053.

[74] S., Kim, S., Park, B., Na, and S. Yoon (2020, April). Spiking-yolo: spiking neural network for energy-efficient object detection. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 34, No. 07, pp. 11270-11277).

[75] B., Han, G., Srinivasan, and K. Roy (2020). RMP-SNN: Residual membrane potential neuron for enabling deeper high-accuracy and

low-latency spiking neural network. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 13558-13567).

[76] S., Deng, and S. Gu (2021). Optimal conversion of conventional artificial neural networks to spiking neural networks. arXiv preprint arXiv:2103.00476.

[77] S., Narduzzi, S. A., Bigdeli, S. C., Liu, and A. L. Dunbar (2022). Optimizing the consumption of spiking neural networks with activity regularization. arXiv preprint arXiv:2204.00607

[78] A., Howard, M., Sandler, G., Chu, L. C., Chen, B., Chen, M., Tan, and H. Adam (2019). Searching for mobilenetv3. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 1314-1324)

[79] E., Hebert Robbins. ASA, A Stochastic Approximation Method, Annals of Mathematical Statistics, 1951, volume 22, pp. (400-407)

[80] J. Kiefer and J. Wolfowitz}, Stochastic Estimation of the Maximum of a Regression Function, Annals of Mathematical Statistics, 1952, volume 23, pp. (462-466), https://doi.org/10.1214/aoms/11[729392

[81] F. Rosenblatt (1958). The perceptron: A probabilistic model for information storage and organization in the brain. Psychological Review, 65(6), 386–408. https://doi.org/10.1037/h0042519

[82] Kingma and Ba 2014. https://doi.org/10.48550/arXiv.1412.6980

[83] D. Muir, F. Bauer, and P. Weidel (2019). Rockpool Documentaton. Zenodo. 10.5281/zenodo.3773845

[84] B. J. Kröger et al., "Modeling speech production using the neural engineering framework," in Proc. 5th CogInfoCom, Nov. 2014, pp. 203–208

[85] J. A. K. Ranjan et al., "A novel and efficient classifier using spiking neural network," J. Supercomput., vol. 76, pp. 6545–6560, May 2019

4

Using FeFETs as Resistive Synapses in Crossbar-based Analog MAC Accelerating Units

Lei Zhang[1,2], David Borggreve[1], Frank Vanselow[1], Ralf Brederlow[2]

[1] Fraunhofer EMFT, Germany
[2] Technical University of Munich, Germany

Abstract

Emerging non-volatile memories (eNVMs) face problems such as insufficient R_{OFF}/R_{ON}-ratio and limited memory operating window that significantly deteriorate the precision of multiply-accumulate computations (MACs), the core computation of artificial intelligence algorithms, using crossbar-based analogue resistive compute-in-memory (CIM) structures. Properly selecting between single-ended and pseudo-differential structures is the fundamental for the most efficient use of the advantages of a particular eNVM, where, e.g., ferroelectric field-effect-transistors (FeFETs) have a large R_{OFF}/R_{ON}-ratio as a great advantage but a significant variability between devices due to the current technology maturity. By investigating and modelling both structures, the results demonstrate that the pseudo-differential structure requires a larger combined operating window from eNVM cells. The reason relies on a statistically enlarged state variation with an increasing number of input channels in the pseudo-differential structure, while the difference between the means of memory's state distributions remains unchanged. Compared to pseudo-differential structures, single-ended structures require a much higher R_{OFF}/R_{ON}-ratio from resistance-switching memories, while the requirement for process variation can be relaxed. The results indicate that FeFETs can be well suited to single-ended crossbar-based structures. However, the considerable state variation of FeFETs makes the applications of FeFETs as resistive synapses hard suited into practice. After investigating existing methods, a gate-cascaded synapse with a higher R_{OFF}/R_{ON}-ratio and a significantly enlarged operating window is proposed. This article discusses boundary conditions for using eNVMs such as FeFETs in crossbar-based analogue MAC accelerating units from a circuit design perspective.

Keywords: Ferroelectric Field-Effect Transistor, Compute-In-Memory, Resistive Synapses, Multiply-Accumulate Computations, Analog MAC Accelerator, Dot-Product Accelerator, Emerging Non-Volatile Memories, Crossbar

4.1 Introduction and Background

Emerging non-volatile memory-based CIM is attracting widespread interest in the field of integrated circuit (IC) design on account of its great potential for enabling a highly parallel analogue (or multi-bits) computation to accelerate MACs in artificial intelligence algorithms sharply [1]. The FeFET, one of the eNVMs, has been studied and implemented for accelerating MACs using its programmable switching property [2][3][4][5], where its threshold voltage can be programmed by adapting the polarization of the ferroelectric layer on the top of the transistor's gate, as illustrated in Figure 4.1. Like using other resistance-switching eNVMs (e.g., ReRAM, OxRAM) for crossbar-based MAC accelerators, FeFETs must fulfil requirements such as a reasonably large R_{OFF}/R_{ON}-ratio, and a sufficient operating window to allow an analogue (multi-bits) computation [6].

Unfortunately, according to the current technology maturity but also the fact that the techniques with smaller sizing dimensions have often a more significant variation, using eNVMs with a minimal size likes FeFETs in a resistive crossbar-based accelerator must face a considerable process variation as shown in Figure 4.1(c), which causes an insufficient operating window, and consequently, leads to an unpromising inference computation precision. Due to this fact, implementing accelerators with either binary states (On or Off) [7] or few bits [8] become an intermediate step towards to fully analogue computation, and significant power efficiencies of 532 TOP/W and over 10000TOPS/W for binary computations are achieved for particular use cases, respectively. However, further improving the efficiency and accuracy of FeFET-based accelerators requires knowledges of the fundamental design challenges of crossbar-based MAC accelerators.

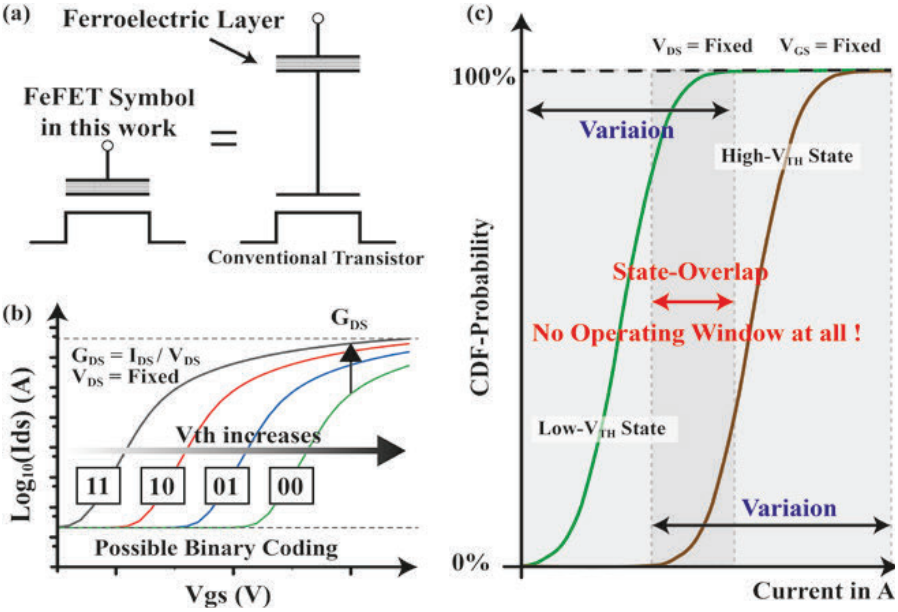


Figure 4.1 (a) shows FeFETs' abstract structure (modified from [9]), where a ferroelectric layer is placed at the top of the transistor's gate. The threshold voltage of FeFETs can be programmed by adapting the polarity of the ferroelectric layer and coded as shown in (b). (c) illustrates possible cumulative distribution functions (CDFs) of real FeFET's current in High-/Low-V_{TH} states, where a state-overlap happens, and the operating window vanishes.

This article shows how to select an optimal structure out of single-ended and pseudo-differential read-out schemes for a particular eNVM. Furthermore, this article discusses the scenarios using FeFETs for two above mentioned structures and how to deal with a limited FeFET's operating window in synapse design.

4.2 Requirements of Crossbar Structure on eNVMs

Figure 4.2 shows single-ended (a) and pseudo-differential (b) structures of analogue crossbar-based MAC accelerating units, where the resistive synapses in the pseudo-differential structure is realized by two resistance-switching devices with oppositely programmed states instead of utilizing a single device in single-ended structures. The computations performed using the single-ended, and pseudo-differential structures can be written as

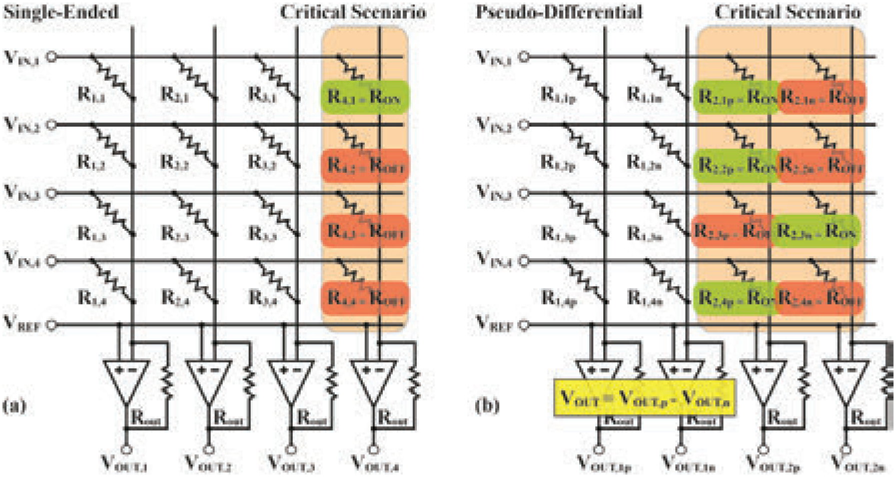


Figure 4.2 Implementations of analogue MAC accelerating units using single-ended (a) and pseudo-differential (b) structures are shown.

Equations (4.1a) and (4.1b), respectively.

$$V_{OUT,\ x} = \sum_{y=0}^{\infty} (V_{REF} - V_{IN,x}) \cdot \frac{R_{out}}{R_{x,\ y}}, \tag{4.1a}$$

$$V_{OUT,\ x} = \sum_{y=0}^{\infty} (V_{REF} - V_{IN,x}) \cdot R_{out} \cdot (\frac{1}{R_{x,\ y_p}} - \frac{1}{R_{x,\ y_n}}), \tag{4.1b}$$

where x and y represent the output channel and input channel as shown in Figure 4.2. The critical scenario happens for single-ended structures if only one resistive synapse is programmed to low-resistance state (LRS) R_{ON} and the others are programmed to high-resistance state (HRS) R_{OFF}. In order to ensure that the output voltage is still can be distinguished, the following condition has to be met:

$$(N-1) \cdot (V_{IN,max} - V_{REF}) \,/\, R_{OFF} \ll (V_{IN,max} - V_{REF}) \,/\, R_{ON}, \tag{4.2}$$

where N is the total number of input channels. The total on- and off-currents in the single-ended structure can be represented by Equations (4.3a) and (4.3b).

$$I_{ON(a)} = (V_{IN,max} - V_{REF}) \,/\, R_{ON} \tag{4.3a}$$

$$I_{OFF(a)} = (N-1) \cdot (V_{IN,max} - V_{REF}) \,/\, R_{OFF} \tag{4.3b}$$

By simplifying Equation (1.2), the final requirement for on-/off-resistance can be given:

$$\frac{R_{\mathrm{OFF}}}{R_{ON}} \gg (N-1)\,. \tag{4.4}$$

The pseudo-differential structure has a different critical scenario, where (N/2 + 1) synapses are positively programmed, and others are negatively programmed. Its total on-current $\mathrm{I}_{\mathrm{ON(b)}}$, and off-current $\mathrm{I}_{\mathrm{OFF(b)}}$ can be represented as:

$$I_{ON(b)} = (V_{IN,max} - \ V_{REF}) \cdot \{(N/2\ +1)/R_{ON} + (N/2-1)/R_{OFF}\} \tag{4.5a}$$

$$I_{OFF(b)} = (V_{IN,max} - \ V_{REF}) \cdot \{(N/2\ +1)/R_{OFF} + (N/2-1)/R_{ON}\} \tag{4.5b}$$

The required on-/off-resistance can be derived from required on-/off-current as

$$I_{OFF(b)} \ll \ I_{ON(b)}, \tag{4.6}$$

so that

$$\frac{R_{\mathrm{OFF}}}{R_{ON}} \gg 1. \tag{4.7}$$

Equation (4.4) and (4.7) indicate that the pseudo-differential structure has a much relaxing requirement on the $\mathrm{R_{OFF}/R_{ON}}$-ratio. However, the process variation can more easily make the computation with the pseudo-differential structure fail.

Considering the resistance variation of eNVMs as

$$X_{ON}{\sim}\mathbb{N}\left(\mu_{ON},\ \sigma_{ON}^2\right) \text{ and } X_{OFF}{\sim}\mathbb{N}(\mu_{OFF}, \sigma_{OFF}^2), \tag{4.8}$$

and assuming that the resistance variation is independent from devices to devices (joint normally distributed), the distributions of the total resistance for on-/off-current in the single-ended structure can be written as

$$Y_{RON(a)}{\sim}\mathbb{N}(\mu_{ON},\ \sigma_{ON}^2) \tag{4.9a}$$

$$Y_{ROFF(a)}{\sim}\ \mathbb{N}(\mu_{OFF}/\,(N-1)\,, \sigma_{OFF}^2/(N-1)^2). \tag{4.9b}$$

Considering the 3σ-variation of eNVMs and assuming no existing state overlap, the relationship between the distribution of total on-/off-resistance can be written as

$$\mu_{ON} + 3 \cdot\ \sigma_{ON} \ll\ \mu_{OFF} - 3 \cdot\ \sigma_{\mathrm{OFF}}. \tag{4.10}$$

For a successful computation, the total resistance for on-/off-current in the single-ended structure must fulfil the relationship as expressed following:

$$\mu_{ON} + 3 \cdot \ \sigma_{ON} \ll \ \mu_{OFF}/\left(N-1\right) - 3 \cdot \sigma_{OFF}/\left(N-1\right). \tag{4.11}$$

It is obvious that the condition of Equation (4.11) will be met if Equations (4.4) and (4.10) can be simultaneously fulfilled. In terms of the process variation, the pseudo-differential structure faces a more serious situation. Deriving a concrete analytical solution for the distribution of total on-/off-resistance in pseudo-differential structures requires lots of efforts, however, still their rough relationship can be checked by making following assumptions:

$$\sigma_{ON}/\mu_{ON} = \ \sigma_{OFF}/\mu_{OFF} \tag{4.12}$$

$$b = \ R_{OFF}/R_{ON} = \ {}_{OFF}/\mu_{ON}. \tag{4.13}$$

Then, the distributions of total on-/off-resistance can be written as

$$Y_{RON(b)} \sim \mathbb{N}\left(\frac{b \cdot \mu_{ON}}{b \cdot (N/2+1) + (N/2-1)}, \left(\frac{b \cdot \sigma_{ON}}{b \cdot (N/2+1) + (N/2-1)}\right)^2\right), \tag{4.14}$$

$$Y_{ROFF(b)} \sim \mathbb{N}\left(\frac{b \cdot \mu_{ON}}{b \cdot (N/2-1) + (N/2+1)}, \left(\frac{b \cdot \sigma_{ON}}{b \cdot (N/2-1) + (N/2+1)}\right)^2\right). \tag{4.15}$$

A similar condition likes Equation(1.11) for the pseudo-differential structure can be written as

$$\begin{aligned}&\left(\frac{b \cdot \mu_{ON}}{b \cdot (N/2-1) + (N/2+1)} - \frac{b \cdot \mu_{ON}}{b \cdot (N/2+1) + (N/2-1)}\right) \\ &\gg 3 \cdot \left(\frac{b \cdot \sigma_{ON}}{b \cdot (N/2+1) + (N/2-1)} + \frac{b \cdot \sigma_{ON}}{b \cdot (N/2-1) + (N/2+1)}\right).\end{aligned} \tag{4.16}$$

By simplifying Equation (4.16), the condition for the pseudo-differential structure can be finally expressed as

$$\sigma_{ON} \ll \frac{2 \cdot (b-1)}{3 \cdot N \cdot (b+1)} \cdot \mu_{ON}. \tag{4.17}$$

Equation (4.17) indicates that increasing the input channels requires reducing the device process variation to keep computation precision unchanged even if the R_{OFF}/R_{ON} is sufficiently large. For easy comparison, Equation (4.10) can be re-written using same assumptions as

$$\sigma_{ON} \ll \frac{(b - N + 1)}{3 \cdot (b + N - 1)} \cdot \mu_{ON}. \tag{4.18}$$

Note that conclusions made from Equations (4.17) and (4.18) are based on some very optimistic assumptions like Equations (4.12) and (4.13) that may vary from the reality. To verify those conclusions, a numerical analysis for the total on-/off-current in both structures is made, and the result is shown in Figure 4.3. This result identifies the above mathematical derivation that increasing R_{OFF}/R_{ON} yields a better computation precision in single-ended structures even if the process variation is significant. For the pseudo-differential structure, ensuring that the device has less variation is the precondition for a good computation precision instead of seeking for a large R_{OFF}/R_{ON}. The requirements given by single-ended and pseudo-differential structures on eNVMs are listed in Table 4.1.

FeFETs have a very high R_{OFF}/R_{ON}-ratio because their switching property is as the same as conventional transistors, but also suffer from the significant process variation due to the current technology maturity. According to those properties, the single-ended structure is a better fit for the design with FeFETs. However, simply using FeFETs in a single-ended structure can still deteriorate computation precision since the state overlap exists, as shown

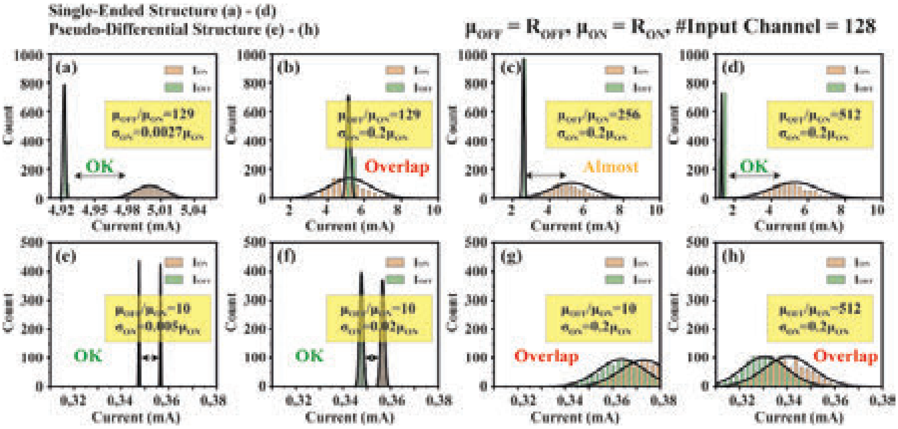


Figure 4.3 The numerical analysis indicates that the R_{OFF}/R_{ON} plays a dominant role for the computation precision in the single-ended structure, where the inherent device process variation is more important for the pseudo-differential structure.

Table 4.1 Comparison between single-ended and pseudo-differential structures

Requirements	Single-Ended	Pseudo-Differential
R_{OFF}/R_{ON}	$\gg (N-1)$	$\gg 1$
Process variation	$\sigma_{ON} \ll \frac{(b-N+1)}{3\cdot(b+N-1)} \cdot \mu_{ON}$	$\sigma_{ON} \ll \frac{2\cdot(b-1)}{3\cdot N\cdot(b+1)} \cdot \mu_{ON}.$
Area	Small	Large
FeFET	suitable	Not suitable

in Figure 4.1 (c). In order to prevent computational precision loss, a proper synapse design should be derived.

4.3 Synapse Design

A good synapse design should have a large R_{OFF}/R_{ON}-ratio and a sufficient operating window without a huge area overhead. This chapter reviews the existing circuit techniques, which could be applied to the synapse design, proposes a gate-cascade technique for improving the synapse's operating window, and shows achieving a better trade-off by combining various circuit techniques.

4.3.1 Conventional Design

Figure 4.4 (a) shows the simplest FeFET synapse that consists of a single FeFET M_{F1} and an access transistor M_a. Its characteristic is the same as conventional transistors' but with an adjustable threshold voltage. However,

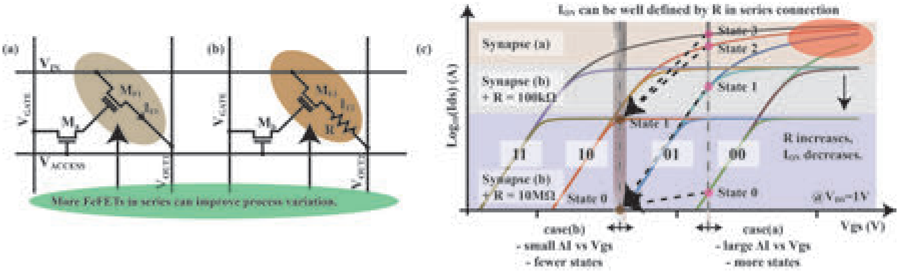


Figure 4.4 Two conventional FeFET synapses are shown, where synapse (b) has an additional current-limiting resistor in the series connection compared to the stand-alone FeFET synapse (a). Both synapses can be activated by connecting a certain gate-voltage using access transistors M_a and M_b, respectively. (c) shows the characteristics of synapses (a) and (b), where a large series resistor enlarges the threshold voltage range of individual states by scarifying the number of available states.

a slight operating-points shift, or the process variation can result in noticeable current changes for different states, as demonstrated by the case (a) in Figure 4 (c). Adding a resistor in series with the FeFET, as demonstrated in Figure 4.4 (b), results in a well-defined on-current, which can be estimated using the linearized transistor equation in the triode-region as following:

$$I_{F2} = \left(\frac{\partial I_{DS}}{\partial V_{DS}} + \frac{1}{R}\right)(V_{OUT2} - V_{IN}) \tag{4.20a}$$

so that

$$I_{F2} = (K_n(V_{GS} - V_{TH} - V_{DS} + 1/R))V_{DS}, \tag{4.20b}$$

where $\mathrm{K_n}$ and $\mathrm{V_{TH}}$ are transistors' transconductance parameter, and threshold voltage. $\mathrm{V_{GS}}$, $\mathrm{V_{DS}}$ are $\mathrm{V_{GATE}}$ and $(\mathrm{V_{OUT2}} - \mathrm{V_{IN}})$ in Figure 4(b), respectively. Considering conditions of:

$$K_n\,(V_{GS} - V_{TH} - V_{DS}) > 0 \tag{4.21a}$$

$$K_n\,(V_{GS} - V_{TH} - V_{DS}) \ll 1/R, \tag{4.21b}$$

The on-current of synapse (b) is well-defined as

$$I_{F2} \approx \frac{V_{OUT2} - V_{IN}}{R}. \tag{4.22}$$

Additionally, the synapse (b)'s characteristic in the saturation region remains the same as conventional transistors. However, FeFET enters earlier into the triode-region depending on the resistor's value because the drain-source voltage is reduced by voltage drop over the resistor, as revealed by the case (b) in Figure 4.4(c).

Compared to the synapse (a), synapse (b) has a higher robustness against the operating-point shifts since it defines the on-current better. After reducing four states of the case (a) to the case (b) with only two states, the impact of the process variation on the on-current is reduced, where the threshold voltage's variation between state 11 and 10 (01 and 00) always results in a well-defined on(off)-current if $\mathrm{V_{GS}}$ is selected between transfer curves of state 10 and 01, as illustrated in Figure 4.4(c). Nevertheless, if the process variation is as significant as shown in Figure 4.1(c), the state overlap cannot be eliminated using synapse (a) and (b) with a single FeFET. A conventional way to yield more stable devices against process variation is connecting multiple FeFETs in series to form a relatively larger FeFET, where a large area overhead may be caused by a large amount of FeFETs in series needed.

4.3.2 Gate-Cascaded FeFETs

Inspired by analysis for single-ended and pseudo-differential structures, the thought was made for enlarging the FeFET's operating window faster than devices variation. Figure 4.5(a) shows a possible implementation, the gate cascaded FeFET. A tiny leakage current I_1 flows through M_1 when gate voltage V_G increases. It enables that the voltage V_X rises with V_G, and correspondingly, FeFET M_2 is turned-on by rising V_X. Due to the diode-connection of M_1 and a very tiny drain-source current I_1, M_1 conducts in the sub-threshold region, and I_1 can be expressed as

$$I_1 = \ I_s \text{exp}(2 - \frac{V_{TH1}}{nU_T}) \text{exp}(\frac{V_{GS1}}{nU_T}), \tag{4.23}$$

where n and I_S are the process-dependent sub-threshold factor and specific current, respectively. U_T represents the thermal voltage, V_{GS1} and V_{TH1} denote the gate-source voltage and threshold voltage of M_1. By solving Equation (4.23), V_{GS1} can be written as

$$V_{GS1} = V_{TH1} + \ln\left(\frac{I_1}{I_s}\right) nU_T - 2nU_T = V_{TH1} + V_C, \tag{4.24}$$

where $V_C$represents the sum of the second and third terms of Equation (4.24). Because only I_1 changes very slightly and any other parameters are process-specific, $V_C$is approximately constant. Therefore, V_{GS2}-V_{TH2} can be written as

$$V_{GS2} - V_{TH2} = V_G - V_C - 2V_{TH0} - \triangle V_{TH1} - \triangle V_{TH2}, \tag{4.25}$$

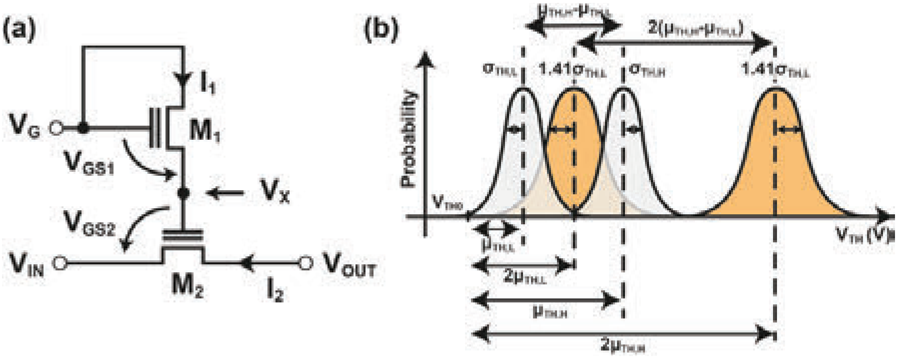


Figure 4.5 The proposed gate-cascaded FeFET synapse, where a diode-connecting FeFET is connected to the gate of another FeFET, is shown in (a). Its statistical distribution is shown in (b), that the distance between threshold voltages doubles and the variation of the state overlap.

where V_{TH0} is the threshold voltage of conventional transistors, ΔV_{TH1} and ΔV_{TH2} represent the threshold voltage changes applied to conventional transistors by the ferroelectric layer. If both FeFETs are simultaneously programmed to the same state (ΔV_{TH1} = ΔV_{TH2}), the operating window V_{Δ}, which is defined as voltage difference $\Delta(V_{GS2}$ - $V_{TH2})$ between high- and low threshold voltage state (HVT and LVT), can be written as

$$V_{\triangle} = \triangle\left(V_{GS2} - V_{TH2}\right) = 2(\triangle V_{TH,HVT} - \triangle V_{TH,LVT}), \tag{4.26}$$

where is twice as conventional synapses.

Considering that the ΔV_{TH}-variation has approximately a normal distribution, the distribution of ΔV_{TH1} and ΔV_{TH2} are referred to X and Y, where

$$X{\sim}\mathbb{N}\left(\mu_{\triangle TH1},\ \sigma^2_{\triangle TH1}\right) \tag{4.27a}$$

$$Y{\sim}\mathbb{N}\left(\mu_{\triangle TH2},\ \sigma^2_{\triangle TH2}\right). \tag{4.27b}$$

Two FeFETs in a circuit should have identical distributions, and they are independent of each other, which means that they are jointly normal. The distribution U of (ΔV_{TH1} + ΔV_{TH2}) with (ΔV_{TH1} = ΔV_{TH2}) can be written as

$$U = X + Y \tag{4.28a}$$

$$U{\sim}\mathbb{N}\left(2\mu_{\triangle TH},\ 2\sigma^2_{\triangle TH}\right) \tag{4.28b}$$

with

$$\mu_{TH} = \mu_{TH1} = \mu_{TH2} \tag{4.29a}$$

$$\sigma_{TH} = \sigma_{TH1} = \sigma_{TH2}. \tag{4.29b}$$

Assuming a 3σ-variation, the operating window for the conventional synapse $V_{\Delta conv}$ and the gate-cascaded FeFET $V_{\Delta prop}$ can be derived as

$$V_{\triangle conv} = (\mu_{HVT} - \mu_{LVT}) - 6(\sigma_{HVT} - \sigma_{HVT}) \tag{4.30a}$$

$$V_{\triangle prop} = (\mu_{HVT} - \mu_{LVT}) - 3\sqrt{2}(\sigma_{HVT} - \sigma_{HVT}). \tag{4.30b}$$

If no overlap between two states is expected, the operating window must be positive (V_{Δ}>0). The conventional synapse (CONV.) and the gate-cascaded synapses (PROP.) operate correctly if the following conditions are fulfilled.

$$CONV.: (\mu_{HVT} - \mu_{LVT}) > 6(\sigma_{HVT} - \sigma_{HVT}) \tag{4.31a}$$

$$PROP.: (\mu_{HVT} - \mu_{LVT}) > 3\sqrt{2}(\sigma_{HVT} - \sigma_{HVT}) \tag{4.31b}$$

According to Equations (4.31a) and (4.31b), synapses with gate-cascaded FeFETs has a 1.4 times relaxed requirement for the process variation compared to the conventional synapses, which is shown in Figure 4.5(b). Extending a single gate-cascaded synapse to N-stage gate-cascaded synapses (N>0), as shown in Figure 4.6(a) and (b), the improvement A can be written as:

$$A = \ (N+1)/\sqrt{N+1}. \tag{4.32}$$

The derivative of the improvement can be written as

$$A' = \frac{1}{\sqrt{(4N+4)}}, \tag{4.33}$$

which indicates that the improvement of the operating window slows down with a continuously increasing number of gate-cascaded stages.

4.3.3 Exploration Results

Figure 4.6 (c) shows the drain-source current curve of FeFETs without gate-cascade, with one- and two-stage gate-cascade. By increasing the number of gate-cascaded stages, the operating window, and the voltage difference between states are enhanced with the same gate voltage V_G. Figure 4.6(d) compares the conventional synapses with 3 FeFETs in series and with 3-stage gate-cascaded FeFETs. The conventional design has a slightly improved operating window compared to a single FeFET that has no operating window at all. The design with gate-cascaded FeFETs has an operating window up to 12.1 times larger than the operating window with 3 FeFETs in series. The I_{ON}/I_{OFF}-ratio, which is exactly equal to R_{OFF}/R_{ON}-ratio, and the operating window are enhanced approximately 2.67 times and 12.1 times compared to the conventional design, respectively.

Table 4.2 Relative Performance Comparison

	Single FeFET	3 FeFETs in series	3-Stage Gate-Cascaded FeFET
# of FeFETs	1	3	3
I_{ON}/I_{OFF}	N/A	α	2.67 α
I_{ON}/I_{OFF} with process variation	N/A	β	26900 β
Operating Window	<0	γ	12.1 γ

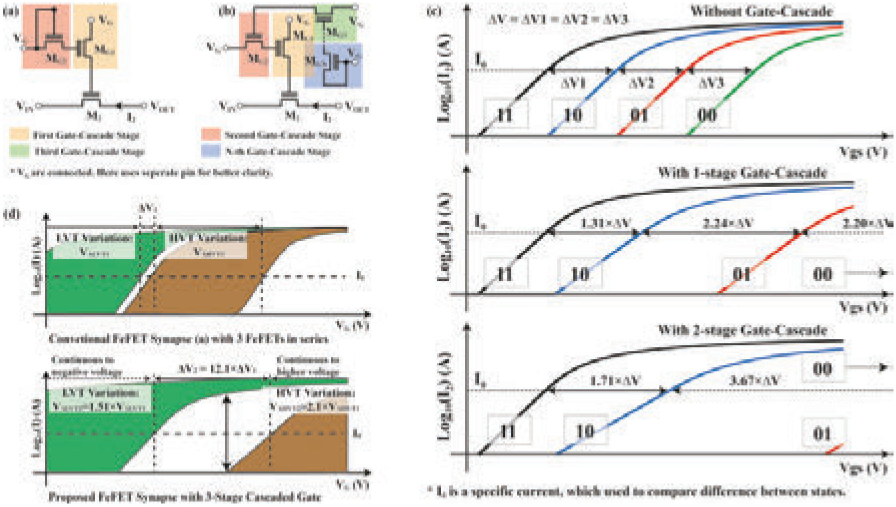

(a)
V_G
V_G
$M_{2,1}$
$M_{2,2}$
V_{IN}
V_{OUT}
M_1
I_1
First Gate-Cascade Stage
Third Gate-Cascade Stage
(b)
V_G
V_G
V_G
V_G
$M_{2,1}$
$M_{2,2}$
$M_{2,3}$
$M_{2,N}$
V_{IN}
V_{OUT}
M_1
I_1
Second Gate-Cascade Stage
N-th Gate-Cascade Stage
* V_G are connected. Here uses seperate pin for better clarity.
(c)
ΔV = ΔV1 = ΔV2 = ΔV3
Without Gate-Cascade
$\mathrm{Log}_{10}(I_1)$ (A)
I_0
ΔV1
ΔV2
ΔV3
11
10
01
00
Vgs (V)
With 1-stage Gate-Cascade
$\mathrm{Log}_{10}(I_1)$ (A)
I_0
1.31×ΔV
2.24×ΔV
2.20×ΔV
11
10
01
00
Vgs (V)
With 2-stage Gate-Cascade
$\mathrm{Log}_{10}(I_1)$ (A)
I_0
1.71×ΔV
3.67×ΔV
00
11
10
01
Vgs (V)
* I_0 is a specific current, which used to compare difference between states.
(d)
ΔV_1
$\mathrm{Log}_{10}(I_1)$ (A)
LVT Variation: $V_{th(1)}$
HVT Variation: $V_{th(1)}$
I_0
V_G (V)
Convational FeFET Synapse (a) with 3 FeFETs in series
Continuous to negative voltage
$\Delta V_2 = 12.1 \times \Delta V_1$
Continuous to higher voltage
LVT Variation: $V_{th(2)} = 1.51 \times V_{th(1)}$
HVT Variation: $V_{th(2)} = 2.1 \times V_{th(1)}$
$\mathrm{Log}_{10}(I_1)$ (A)
I_0
V_G (V)
Proposed FeFET Synapse with 3-Stage Cascaded Gate

Figure 4.6 (a) and (b) show a two-stage and a N-stage gate-cascaded FeFET synapse, respectively. (c) shows the change of their characteristics, where the voltage difference between states is enlarged. (d) demonstrate the characteristic of a conventional synapse with three serially connected FeFET and a three-stage gate cascaded FeFET. The gate-cascaded FeFET achieved 12.1 times larger operating window than conventional design.

Nevertheless, the drawbacks that the gate-cascaded FeFET brings need to be pointed out:

1. Programming FeFETs requires a particularly high voltage applied to FeFETs. The more gate-cascaded stages are used, the more access high-voltage transistors are required, which occupy the most area in the design as the design example shown in Figure 4.7(a) and (b).
2. Shifting threshold voltage to a very high value does not yield much. On the one hand, the improvement slows down with an increasing number of cascaded stages, according to Equation (4.33). On the other hand, the real gate voltage cannot achieve a very high potential. An optimal number of stages is highly technology dependent.

Since drawbacks listed above, combing different methods in a right manner will result into an optimal design point. A design example is shown in Figure 4.7(a), where

- $M_{3,L}$and $M_{3,R}$ play the role of resistors to limit the current,
- M_1 and M_2 are serially connected FeFETs for reducing the process variation slightly,

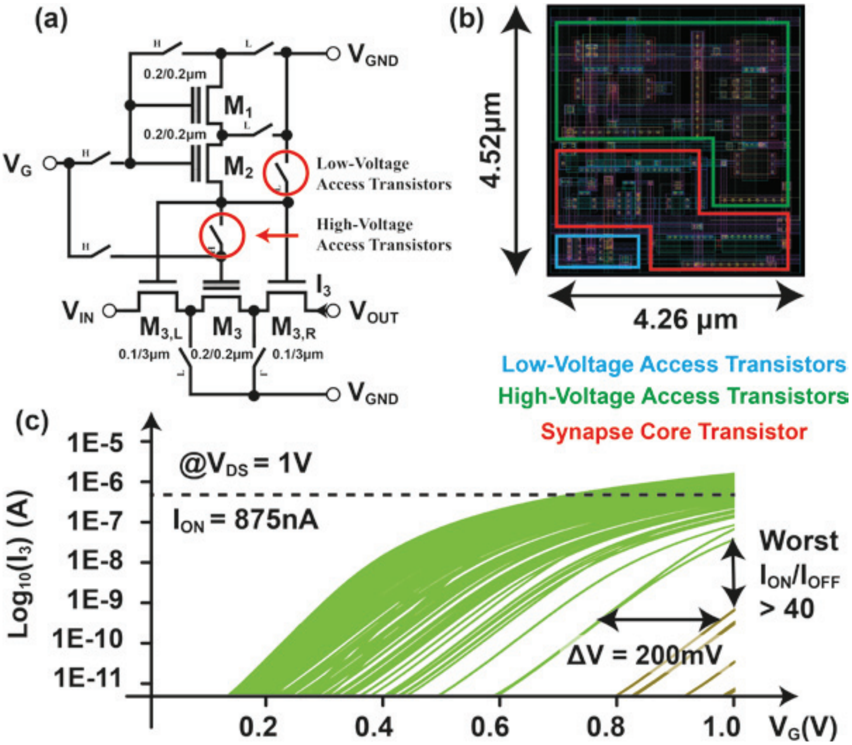


Figure 4.7 design example, which combine the proposed and conventional techniques, is shown in (a). (b) displays the layout of this design example. (c) indicates that a up to 200mV operating window is achieved using 1-stage gate-cascade.

- M_1, M_2 and M_3 build a one-stage gate cascade for generally enhancing operating window.

The performance of the design example is shown in Figure 4.7(c). Depending on the need, the operating window can further be enhanced by either connecting more FeFETs in series or applying more gate-cascaded stages.

4.4 Conclusion

This article mainly reviews circuit aspects, such as select of the best readout structure and the design of resistive synapses, for using FeFETs in a crossbar-based analogue MAC accelerating unit. Both analytical and numerical analyses indicate that FeFETs have a better fit to the single-ended structure, which requires a high R_{OFF}/R_{ON}-ratio but has relaxing requirements for the process variation. Those considerations can be transferred to other types of eNVMs. Furthermore, this article compares three ways of using FeFETs as resistive synapses, and the result indicates that only combining different methods can lead into a high R_{OFF}/R_{ON}-ratio and non-overlapped states without a significant area overhead. For implementing an entire accelerator, other design aspects, such as programming algorithms, parasitic effects, design of efficient data-converter and so on, need to be considered. However, this article gives readers an essential guidance how to start using FeFETs or other eNVMs, for crossbar-based analogue MAC accelerators.

Acknowledgements

This work is supported through the project TEMPO. TEMPO has received funding from the ECSEL Joint Undertaking (JU) under grant No.826655. The JU receives support from the European Union's (EU's) Horizon 2020 research and innovation program and France, Belgium, Germany, Netherlands, Portugal, Spain, Switzerland. TEMPO has also received funding from the German Federal Ministry of Education and Research (BMBF) under Grant No.16ESE0407. This work reflects only the author's view, JU, EU and BMBF are not responsible for any use that may be made of the information it contains.

References

[1] W. Haensch, T. Gokmen, and R. Puri, "The next generation of deep learning hardware: Analog Computing", *Proceeding of the IEEE*, vol. 107, no. 1, pp. 108-122, 2019.

[2] M. Jerry, P.-Y. Chen, J. Zhang, P. Sharma, K. Ni, S. Yu and S. Datta, "Ferroelectric fet analog synapse for acceleration of deep neural network training", in *2017 IEEE International Electron Devices Meeting (IEDM)*, pp. 6.2.1-6.2.4, 2017.

[3] M. Jerry, S. Dutta, A. Kazemi, K. Ni, J. Zhang, P.-Y. Chen, P. Sharma, S. Yu, X. S. Hu, M. Niemier, and S. Datta, "A ferroelectric field effect transistor based synapse weight cell", *Journal of Physics D: Applied Physics*, vol. 51, no. 43, august 2018.

[4] S. Oh, H. Hwang, and I. K. Yoo, "Ferroelectric materials for neuromorphic computing", *APL Materials*, vol. 7, no. 9, p.091-109, 2019.

[5] N. E. Miller, Z. Wang, S. Dash, A. I. Khan, and S. Mukhopadhyay, “Characterization of drain current variations in fefets for pim-based dnn accelerator”, in *2021 IEEE 3rd International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, pp. 1-4, 2021.

[6] P.-Y. Chen, X. Peng, and S. Yu, “NeuroSim+: An integrated device-to-algorithm framework for benchmarking synaptic devices and array architectures,” in *2017 IEEE International Electron Devices Meeting (IEDM)*, pp.6.1.1-6.1.4, 2017.

[7] D. Bankman, L. Yang, B. Moons, M. Verhelst, and B. Murmann, “An always-on 3.8μJ 86% cifar-10 mixed-signal binary cnn processor with all memory on chip in 28-nm process”, *IEEE Journal of Solid-State Circuits*, vol. 54, no. 1, pp. 158-172, 2019.

[8] T. Soliman, F. Müller, T. Kirchner, T. Hoffmann, H. Ganem, E. Karimov, T. Ali, M. Lederer, C. Sudarshan, T. Kämpfe, A. Guntoro, and N. Wehn, “Ultra-low power flexible precision fefet based analog in-memory computing”, in 2020 *IEEE International Electron Devices Meeting (IEDM)*, pp. 29.2.1-29.2.4, 2020.

[9] A. Aziz, E. T. Breyer, A. Chen, X. Chen, S. Datta, S. K. Gupta, M. Hoff-mann, X. S. Hu, A. Ionescu, M. Jerry, T. Mikolajick, H. Mulaosmanovic, K. Ni, M. Niemier, I. O’Connor, A. Saha, S. Slesazeck, S. K. Thirumala, and X. Yin, “Computing with ferroelectric fets: Devices, models, systems,and applications,” in *2018 Design, Automation, Test in Europe Conference Exhibition (DATE)*, pp. 1289–1298, 2018.

5

Emerging In-memory Computing for Neural Networks

Nellie Laleni[1], Taha Soliman[2], Alptekin Vardar[1], and Thomas Kämpfe[1]

[1]Fraunhofer IPMS
[2]Robert Bosch GmbH

Abstract

Recently, deep neural networks (DNNs) have proved their success in performing various tasks at high accuracy. However, these networks come at a high cost of computational and memory requirements and with the continuously growing neural networks sizes, conventional von Neumann accelerators hit the memory wall. Processing-in-memory (PIM) acceleration is heavily investigated to deliver the aforementioned requirements with a great potential to further accelerate these application and meet the possible future needs. In this chapter, we explore the state-of-the-art, challenges and future possibilities of the PIM based DNN accelerators. First, we explore various volatile and non-volatile memory cells that are commonly used for PIM architectures. Second, we discuss the possible approaches to design a PIM accelerator (digital, analog, mixed-signal processing). Third, we investigate the operational accuracy these architectures are offering, the requirements these architectures enforce when it comes to the inferred network quantization. Finally, we conduct an extensive comparison between these architectures.

5.1 Memory Technologies

5.1.1 Volatile Memories

For many years, the main memory cells to provide storage space for any computational process were volatile memories. Volatile Memory is called the memory which retains the data only as long as there is power supplied. The most common volatile memories are the dynamic random access memory (DRAM) and the static random-access memory (SRAM).

An SRAM cell is constructed from two transistors and four more transistor forming two cross-coupled inverters storing one single bit (Figure 5.1). The SRAM cell is preferred among the volatile memories due to its low access time and high performance comparing with the DRAM of which the threshold voltage of the access transistor is very high [1]. However, SRAM is considered as en expensive memory and dominates a high amount of area in a digital chip and also the total chip leakage current [2]. Although, in advanced technologies, the decreased VDD can lower the leakage current, the storage capacitance of a bitcell SRAM is reduced and soft error rate (SER) is introduced [3]. Moreover, with respect to the NVM, it lacks of high power efficiency and exposes higher read delay, for higher temperature [4]. SRAM cell is often used as the main memory cell where the MAC operations are performed [5][6].

At the moment, DRAM is the most popular type of memory when designing an AI accelerator and needing memory storage. Its simple design consists of a transistor and a storage capacitance (Figure 5.1). The need for very large and dense quantities of memory, led to the usage of DRAM to be the main off-chip memory [7]. The cost of the DRAM cell is less than the SRAM, but the DRAM memory needs a circuit to periodically refresh the memory since the capacitance needs to be discharge also, DRAM's capacitance leaks current and the data has to be transferred at the main chip, so higher latency and

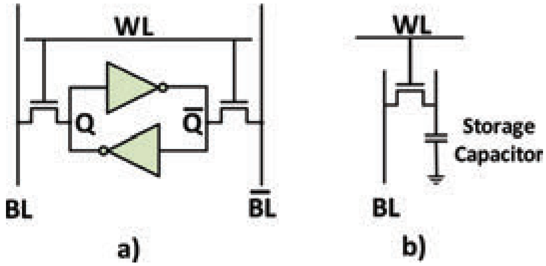


Figure 5.1 Conventional volatile memory cells a) 6T SRAM cell and b) DRAM cell.

power needed [7]. Recent studies has designed new techniques to compensate those effects like the time minimization of the DRAM access [8] or the latency in sense of energy per access [9]. Nevertheless, the biggest concern of the DRAM memory seems to be the scaling limit with the newer technologies and the smaller sizes of the transistor [10].

5.1.2 Non Volatile Memories

The aforementioned disadvantages of the volatile memories lead the researchers to investigate another type of memories, the non volatile memories (NVM). These memories have the ability to retain the data even if the power supply is disconnected. They present high power efficiency with respect to the volatile memories and low latency since the network's operations happen inside the memory. Although the low cost and high density, they may present some reliability issues like data retention and finite endurance, resulting in high bit-error-rates (BER) in the stored weights [11]. Recent studies have showed some solutions for the BER problem like error correction code (ECC) [12][13], but these techniques demand high power during the read operation which can not be compatible with the new edge technologies. Some of the most popular NVM are the flash memory, the resistive random-access memory (ReRAM or RRAM) and the ferroelectric RAM (FeRAM, F-RAM or FRAM).

A Flash memory cell is simply a MOSFET cell, except that a polysilicon floating gate (or a silicon nitride charge trap layer) is sandwiched between a tunnel oxide and an interpolyoxide to form a charge storage layer [10]. The floating gate is used to store the data and it provides programming and erasing process. However, the Flash memory lacks of scalability since a conventional Flash type of memory needs a tunnel oxide layer thickness of 8nm to avoid charge loss and maintain the data (data retention) for 10 years [14]. As a result, a reduction of device dimension could cause threshold voltage shift, retention, endurance and dielectric leakage [10][15].

The ReRAM cell consists of one Memristor and one transistor. Memristor is a device which acts as a programmable resistance, so the voltage level of the transistor can be determined. This voltage level represents the state/value of the weights in a neural network. However, concerning to the power consumption, the ReRAM presents gate leakage and relatively high power consumption for low latency and vise versa [4]. Moreover, a significant effect which should be taken into consideration when designing a NN accelerator with ReRAM, is that for any small variation of the V_{th}, the write delay is increased exponentially [4].

In order to avoid the ReRAMs high write power and long read latency (RC delay), studies focused on FeRAM as one popular upcoming technologies for NVM [16]. It is firstly introduced in [17] where ferroelectricity in silicon doped hafnium oxide ($Si\ :\ HFO_2$) is presented as a high scalable and complementary metal-oxide semiconductor (CMOS) compatible technology (ferroelectric field-effect transistor - FeFET). It consists of a transistor and a capacitor structure which gives the transistor the ability to be programmed and erased in different levels with respect to its V_{th} It has already been integrated into various CMOS new edge technologies [18][19] and it presents low device-to-device variation. One disadvantage of the FeFET is that during the read operation, a leakage current can be detected which is involved to a small writing pulse to each cell [10]. It is a fast memory (higher read speed than Flash and SRAM memory) with high endurance and low hold power making FeFET a competitive technology of NVM.

5.2 In-memory Architecture

In this section, we review the different design trends in the field of in-memory computing architectures based on different memory technologies and targeting various neural networks. The different architectures are explored according to their computational domain, flexibility and programmability, used technology, target networks and their representation and finally the reliability and accuracy of the computations.

5.2.1 Computational Domain

As In-memory architectures main idea is to perform the target operation in memory by leveraging the memory cell properties in the analog domain or in more digital approach. However, pure analog domain usually targets neuromorphic computations which is not the main scope of this survey. In this section, we will be exploring two main trends in In-memory architecture; the mixed signal based architectures and digital based architectures. For each, we will investigate the possible advantages and disadvantages each is offering as well as the potential each hold for future applications.

5.2.1.1 Mixed signal approach

In this approach, the main computation is realized by using the analog properties of the memory cell within the memory crossbar or sub-array. As shown in Figure 5.2, the min idea here depends on storing the weight value

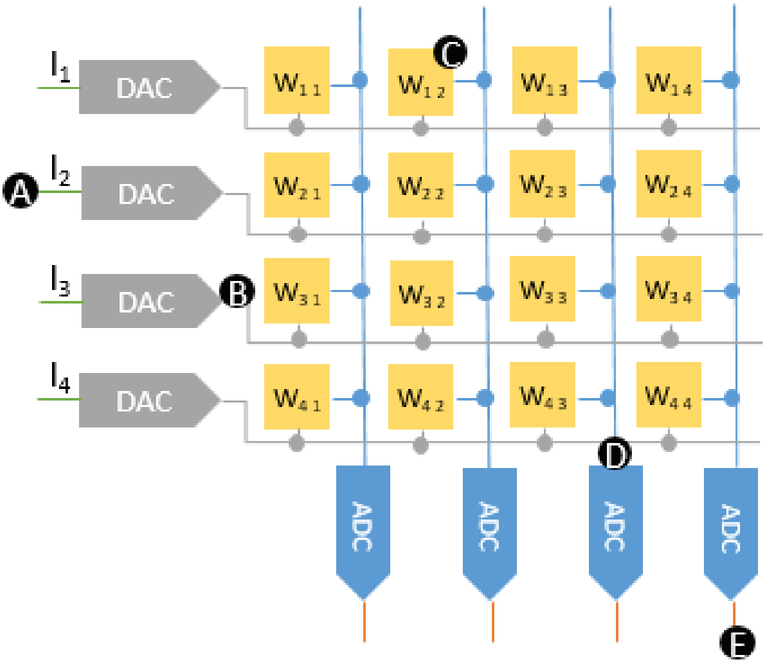


Figure 5.2 General concept of mixed-signal in-memory crossbar (A) The digital activation of the computed layer. (B) DACs convert the digital input into an analog signal to be applied to the memory cell. (C) Memory cell storing the kernel value of the currently computed layer. (D) The summation line which accumulates the result signal out of the memory cell representing the operation results. (E) ADCs convert back the result into the digital domain for any further processing.

within memory cell and using a digital to analog converter (DAC) to represent the input feature value as an applied voltage. The result of such multiplication operation between the input and weight values are represented by the output signal of the memory cell as explained previously. Based on kirchhoff law, the result of different multiplications along the BL is accumulated and finally forwarded to the analog to digital converter (ADC) unit that yields the final result of the performed MAC operations.

Several architectures [20][21] adopt this crossbar organization as their main processing element. This structure became a very popular crossbar structure because of its very high throughput as well as matching the dominant MAC operation. However, this approach holds couple of drawbacks as it requires a number of ADCs and DACs which reflects on the chip area and the power consumption of the overall processing element. These components

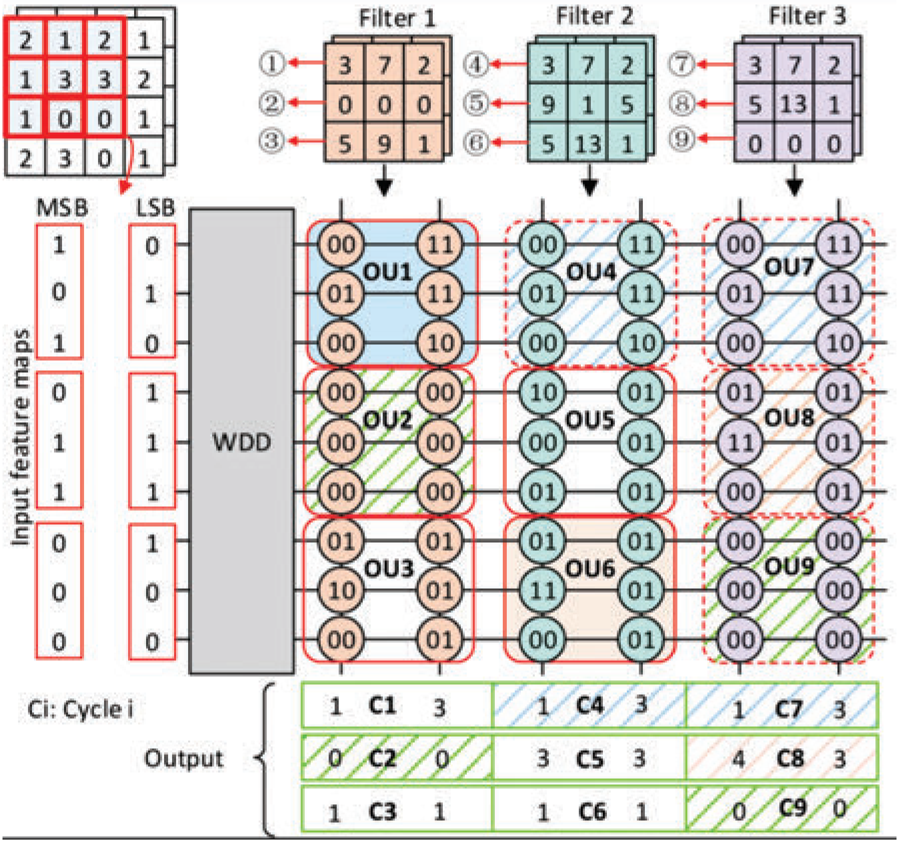


Figure 5.3 Eliminated the DACs and instead serialize the activation by applying only a single bit at each cycle [24].

can amount up to 23% and 61% of the system area and power respectively as shown in [22] or in extreme cases up to 99% and 85% as in [23]. To limit these drawbacks, several architectures [24][25] eliminated the DACs and instead serialize the activation by applying only a single bit at each cycle as shown in Figure 5.4. This approach also reduce the ADCs size as the accumulated analog value is also smaller. However, this approach requires more cycles to perform single operation (usually number of cycles equivalent to the activation precision.),

Another approach to limit such drawback was to adapt a bit decomposition approach by either decomposing the activation as the previously

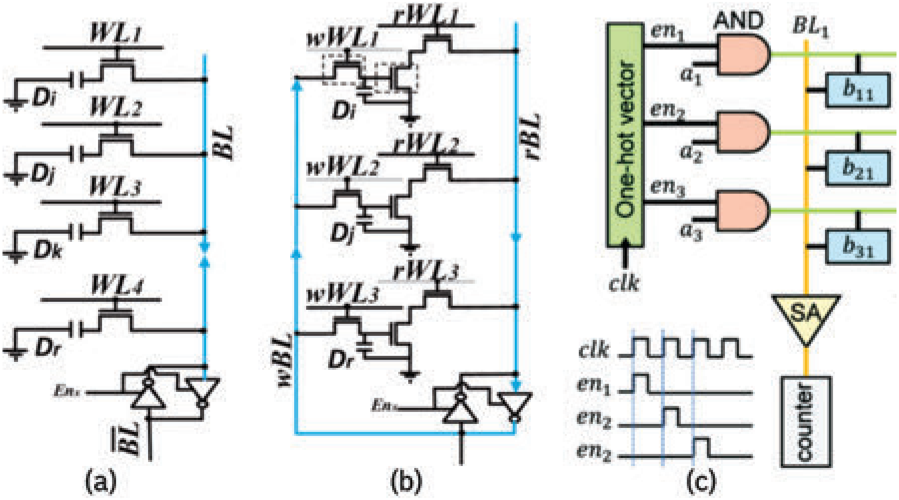


Figure 5.4 (a) Several row activation approach such as Ambit's TRA [28]. (b) Changing subarray unit cell structure whether with extra transistors or operation mode as in DRISA 3T1C [29]. (c) Activating only one row at a time and use the row activation as an operand as in FlexPim [28].

mentioned approach or by decomposing the weight stored as well [26][27]. In this approach, a compromise between the number of cycles needed and the size of ADCs and DACs is investigated to balance the throughput, area and power of the architecture.

5.2.1.2 Digital approach

Another way for in-memory computation is adapting a completely digital approach. Such structure depends on either decomposing both the weights and activations completely or quantizing the parameters to binarized representation. This in return converts the MAC operation into bulk logical bitwise operations that need to be followed by additions, shiftings and comparisons. The memory cells are usually used to perform the bitwise operations and the rest of operations are done by supporting computational blocks. Several architectures realize the bulk operations as shown in Figure 5.4 through parallel sub-array activations representing the operands [28] or through single row activation based on one of the operands [29][30]. Another realization is possible through modifying the cell to perform the target logical operation as in [31][32].

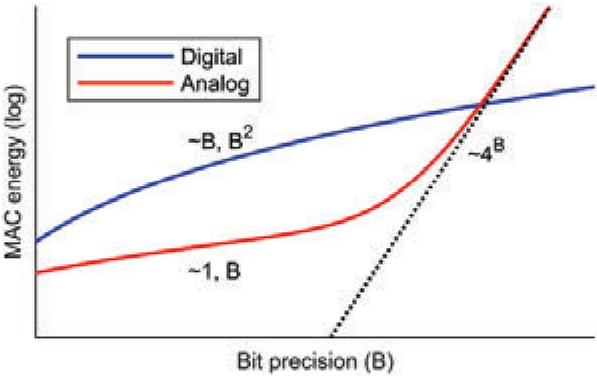


Figure 5.5 The relation between the energy cost for digital and analog MAC operations versus bit precision. [33].

Compared to the mixed signal approach, this approach allows for high speed due to the eliminations of the analog blocks as well as high power efficiency. However, the decomposition of the MAC operations reflects on the operation latency. Also, low bit quantization limits the architecture usage to neural networks models that can tolerate such quantization.

5.2.2 Target Network Quantization

In this section, we investigate the possible targeted neural network weights' quantization and representation. Ranging from floating point representation to binary representation, wide range of presented architectures has been offered with each targeting a specific representation or in some cases try to be flexible and target several possible weights' quantization. As highlighted in earlier sections and shown in Figure 5.5, such network properties affect directly the architecture choices such as the computational domain, selected technology, etc but also it is reflected on the network accuracy and possibility of using the architecture for training as well as inference.

5.2.2.1 Floating point architectures

Floating point representation is considered as the most accurate form of the targeted network since it is usually the representation used during the training and design phase. Architectures targeting such representation are usually used for training mainly [34][35]. The main advantage such architectures are offering is the elimination of accuracy loss from network representation.

Also, these architectures targets the largest range of networks as the only limitation in that case is the support of network layers type or not.

However, these architectures are usually suffer from a trade off between high power consumption or lower throughput. Depending on the design choices, generally expensive power blocks are used which maintain high throughput on the expense of high power consumption. This makes them way efficient when compared to general purposed computing devices such as graphic processing units (GPUs) but losing the edge compared to low power ASIC designs.

5.2.2.2 Fixed-point architectures

Architectures targeting fixed-point weight representation are the most popular among in-memory architectures. Due to advanced neural network optimization techniques [36][22], weights can be represented using fixed-point precision as low as 4-bit in large complex networks. With such low representation, these architectures store the weight in the memory cell which boost the system throughput and performance as shown previously. However, such architectures suffer from several drawbacks related to accuracy losses that can occur due to sever compression. Also, such representation limit the usage of these architectures in training related tasks and confine them more to inference based tasks.

5.2.2.3 Binarized architectures

Motivated by the extremely reduced memory/computational requirements with marginal degradation in accuracy for some networks [37][38], several architectures are built to target binary/ternary operations. In these architectures [39][5], the main operation performed by the memory cell is usually very simple logical operation. Also, these architectures reduce the memory cell irregularities to the minimum as each cell stores a single bit.

However, the binarization limits the architecture usage to a limited number of networks that can currently adopt such representation. The main argument these architecture is dependant on is the consistent advance in the binarization techniques that can allow for more networks to be using such architecture.

5.2.2.4 Flexible precision architectures

A recent popular growing trend is to target various representations simultaneously where these representations can range from binarized to full precision floating point as in [26][30]. In these architectures, the weights and inputs precision can be traded for either higher throughput or reduced power consumption.

Most of these architectures depend on weight bit decomposition or input serialization explained earlier. Such flexibility allows for the use of these architectures for a wider range of networks and for both inference and training purposes. However, such flexibility comes with a cost compared to fixed representation architectures. For example in [29], to achieve such flexibility, a hierarchical network on chip is required which adds extra hardware either to the chip busses or the complexity of the chip control.

References

[1] Y. Liang, K. Gopalakrishnan, P. B. Griffin, and J. D. Plummer, "From dram to sram with a novel sige-based negative differential resistance (ndr) device," in *IEEE InternationalElectron Devices Meeting, 2005. IEDM Technical Digest.*, 2005, pp. 959–962.

[2] N. S. Kim, K. Flautner, D. Blaauw, and T. Mudge, "Circuit and microarchitectural techniques for reducing cache leakage power," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 12, no. 2, pp. 167–184, 2004.

[3] B. H. Calhoun and A. P. Chandrakasan, "A 256-kb 65-nm sub-threshold sram design for ultra-low-voltage operation," *IEEE Journal of Solid-State Circuits*, vol. 42, no. 3, pp. 680–688, 2007.

[4] A. I. Soumitra Pal, Subhankar Bose, "Design of memristor based low power and highly reliable reram cell," in *Springer*, 2019. [Online]. Available: https://link.springer.com/article/10.1007/s00542-019-04582-1

[5] K. Ando, K. Ueyoshi, K. Orimo, H. Yonekawa, S. Sato, H. Nakahara, M. Ikebe, T. Asai, S. Takamaeda-Yamazaki, T. Kuroda, and M. Motomura, "Brein memory: A 13-layer 4.2 k neuron/0.8 m synapse binary/ternary reconfigurable in-memory deep neural network accelerator in 65 nm cmos," in *2017 Symposium on VLSI Circuits*, 2017, pp. C24–C25.

[6] B. Moons, R. Uytterhoeven, W. Dehaene, and M. Verhelst, "Envision: A 0.26-to-10tops/w subword-parallel dynamic-voltage-accuracy-frequency-scalable convolutional neural network processor in 28nm fdsoi," in *2017 IEEE International Solid-State Circuits Conference (ISSCC)*, 2017, pp. 246–247.

[7] Y. Chen, T. Luo, S. Liu, S. Zhang, L. He, J. Wang, L. Li, T. Chen, Z. Xu, N. Sun, and O. Temam, "Dadiannao: A machine-learning supercomputer," in *2014 47th Annual IEEE/ACM International Symposium on Microarchitecture*, 2014, pp. 609–622.

[8] J. Li, G. Yan, W. Lu, S. Jiang, S. Gong, J. Wu, and X. Li, "Smartshuttle: Optimizing off-chip memory accesses for deep learning accelerators," in *2018 Design, Automation Test in Europe Conference Exhibition (DATE)*, 2018, pp. 343–348.

[9] R. V. Wicaksana Putra, M. Abdullah Hanif, and M. Shafique, "Drmap: A generic dram data mapping policy for energy-efficient processing of convolutional neural networks," in *2020 57th ACM/IEEE Design Automation Conference (DAC)*, 2020, pp. 1–6.

[10] J. Meena, S. Sze, U. Chand, and T.-Y. Tseng, "Overview of emerging non-volatile memory technologies," *Nanoscale Research Letters*, vol. 9, pp. 1–33, 09 2014.

[11] M. Hasan and B. Ray, "Tolerance of deep neural network against the bit error rate of nand flash memory," in *2019 IEEE International Reliability Physics Symposium (IRPS)*, 2019, pp. 1–4.

[12] Y. Cai, S. Ghose, E. F. Haratsch, Y. Luo, and O. Mutlu, "Error characterization, mitigation, and recovery in flash-memory-based solid-state drives," *Proceedings of the IEEE*, vol. 105, no. 9, pp. 1666–1704, 2017.

[13] N. Mielke, T. Marquart, Ning Wu, J. Kessenich, H. Belgal, E. Schares, F. Trivedi, E. Goodness, and L. R. Nevill, "Bit error rate in nand flash memories," in *2008 IEEE International Reliability Physics Symposium*, 2008, pp. 9–19.

[14] C.-W. Hu, K.-M. Chang, C.-H. Tu, C.-N. Chiang, C.-C. Lin, S. Sze, and T.-Y. Tseng, "Nisige nanocrystals for nonvolatile memory devices," *Applied Physics Letters*, vol. 94, pp. 062 102–062 102, 02 2009.

[15] D. Ielmini, "Reliability issues and modeling of flash and post-flash memory," *Microelectronic Engineering - MICROELECTRON ENG*, vol. 86, pp. 1870–1875, 07 2009.

[16] Y. Long, D. Kim, E. Lee, P. Saha, B. A. Mudassar, X. She, A. I. Khan, and S. Mukhopadhyay, "A ferroelectric fet-based processing-in-memory architecture for dnn acceleration," *IEEE Journal on Exploratory Solid-State Computational Devices and Circuits*, vol. 5, no. 2, pp. 113–122, 2019.

[17] D. B. U. S. T. S. Böscke1, J. Müller and U. Böttger, "Ferroelectricity in hafnium oxide thin films," *Applied Physics Letters*, 08 2011.

[18] M. Trentzsch, S. Flachowsky, R. Richter, J. Paul, B. Reimer, D. Utess, S. Jansen, H. Mulaosmanovic, S. Müller, S. Slesazeck, J. Ocker, M. Noack, J. Müller, P. Polakowski, J. Schreiter, S. Beyer, T. Mikolajick, and B. Rice, "A 28nm hkmg super low power embedded nvm technology based on ferroelectric fets," in *2016 IEEE International Electron Devices Meeting (IEDM)*, 2016, pp. 11.5.1–11.5.4.

[19] S. Dünkel, M. Trentzsch, R. Richter, P. Moll, C. Fuchs, O. Gehring, M. Majer, S. Wittek, B. Müller, T. Melde, H. Mulaosmanovic, S. Slesazeck, S. Müller, J. Ocker, M. Noack, D. Löhr, P. Polakowski, J. Müller, T. Mikolajick, J. Höntschel, B. Rice, J. Pellerin, and S. Beyer, "A fefet based super-low-power ultra-fast embedded nvm technology for 22nm fdsoi and beyond," in *2017 IEEE International Electron Devices Meeting (IEDM)*, 2017, pp. 19.7.1–19.7.4.

[20] A. Biswas and A. P. Chandrakasan, "Conv-ram: An energy-efficient sram with embedded convolution computation for low-power cnn-based machine learning applications," in *2018 IEEE International Solid - State Circuits Conference - (ISSCC)*, 2018, pp. 488–490.

[21] Q. Dong, M. E. Sinangil, B. Erbagci, D. Sun, W. Khwa, H. Liao, Y. Wang, and J. Chang, "15.3 a 351tops/w and 372.4gops compute-in-memory sram macro in 7nm finfet cmos for machine-learning applications," in *2020 IEEE International Solid- State Circuits Conference - (ISSCC)*, 2020, pp. 242–244.

[22] A. Shafiee, A. Nag, N. Muralimanohar, R. Balasubramonian, J. P. Strachan, M. Hu, R. S. Williams, and V. Srikumar, "Isaac: A convolutional neural network accelerator with in-situ analog arithmetic in crossbars," in *2016 ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA)*, 2016, pp. 14–26.

[23] L. Ni, Y. Wang, H. Yu, W. Yang, C. Weng, and J. Zhao, "An energy-efficient matrix multiplication accelerator by distributed in-memory computing on binary rram crossbar," in *2016 21st Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2016, pp. 280–285.

[24] Y. Zhang, Z. Jia, Y. Pan, H. Du, Z. Shen, M. Zhao, and Z. Shao, "Pattpim: A practical reram-based dnn accelerator by reusing weight pattern repetitions," in *2020 57th ACM/IEEE Design Automation Conference (DAC)*, 2020, pp. 1–6.

[25] G. Murali, X. Sun, S. Yu, and S. K. Lim, "Heterogeneous mixed-signal monolithic 3-d in-memory computing using resistive ram," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 29, no. 2, pp. 386–396, 2021.

[26] T. Soliman, R. Olivio, T. Kirchner, M. Lederer, T. Ka'mpfe, A. Guntoro, and N. Wehn, "A ferroelectric fet based in-memory architecture for multi-precision neural networks," in *2020 33rd IEEE International System-on-Chip Conference (SOCC)*, 2020.

[27] Y. Cai, T. Tang, L. Xia, B. Li, Y. Wang, and H. Yang, "Low bit-width convolutional neural network on rram," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 39, no. 7, pp. 1414–1427, 2020.

[28] V. Seshadri, D. Lee, T. Mullins, H. Hassan, A. Boroumand, J. Kim, M. A. Kozuch, O. Mutlu, P. B. Gibbons, and T. C. Mowry, "Ambit: In-memory accelerator for bulk bitwise operations using commodity dram technology," in *2017 50th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, 2017, pp. 273–287.

[29] Y. Long, E. Lee, D. Kim, and S. Mukhopadhyay, "Flex-pim: A ferroelectric fet based vector matrix multiplication engine with dynamical bitwidth and floating point precision," in *2020 International Joint Conference on Neural Networks (IJCNN)*, 2020, pp. 1–8.

[30] Y. Long, D. Kim, E. Lee, P. Saha, B. A. Mudassar, X. She, A. I. Khan, and S. Mukhopadhyay, "A ferroelectric fet-based processing-in-memory architecture for dnn acceleration," *IEEE Journal on Exploratory Solid-State Computational Devices and Circuits*, vol. 5, no. 2, pp. 113–122, 2019.

[31] S. Liu, H. Zhu, C. Chen, L. Zhang, and C. . Richard Shi, "Xnoram: An efficient computing-in-memory architecture for binary convolutional neural networks with flexible dataflow mapping," in *2020 2nd IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, 2020, pp. 21–25.

[32] S. Li, D. Niu, K. T. Malladi, H. Zheng, B. Brennan, and Y. Xie, "Drisa: A dram-based reconfigurable in-situ accelerator," in *2017 50th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, 2017, pp. 288–301.

[33] B. Murmann, "Mixed-signal computing for deep neural network inference," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 29, no. 1, pp. 3–13, 2021.

[34] S. Gupta, M. Imani, H. Kaur, and T. S. Rosing, "Nnpim: A processing in-memory architecture for neural network acceleration," *IEEE Transactions on Computers*, vol. 68, no. 9, pp. 1325–1337, 2019.

[35] S. R. Nandakumar, I. Boybat, V. Joshi, C. Piveteau, M. Le Gallo, B. Rajendran, A. Sebastian, and E. Eleftheriou, "Phase-change memory

models for deep learning training and inference," in *2019 26th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, 2019, pp. 727–730.

[36] P. Chi, S. Li, C. Xu, T. Zhang, J. Zhao, Y. Liu, Y. Wang, and Y. Xie, "Prime: A novel processing-in-memory architecture for neural network computation in reram-based main memory," in *2016 ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA)*, 2016, pp. 27–39.

[37] M. Courbariaux, Y. Bengio, and J. David, "Binaryconnect: Training deep neural networks with binary weights during propagations," *CoRR*, vol. abs/1511.00363, 2015. [Online]. Available: http://arxiv.org/abs/1511.00363

[38] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi, "Xnor-net: Imagenet classification using binary convolutional neural networks," *CoRR*, vol. abs/1603.05279, 2016. [Online]. Available: http://arxiv.org/abs/1603.05279

[39] H. Valavi, P. J. Ramadge, E. Nestler, and N. Verma, "A 64-tile 2.4-mb in-memory-computing cnn accelerator employing charge-domain compute," *IEEE Journal of Solid-State Circuits*, vol. 54, no. 6, pp. 1789–1799, 2019.

6

Artificial Intelligence Advancements for Digitising Industry

Ovidiu Vermesan[1], and Reiner John[2]

[1]SINTEF AS, Norway
[2]AVL List GmbH, Austria

Abstract

In the digital transformation era, when flexibility and know-how in manufacturing complex products become a critical competitive advantage, artificial intelligence (AI) is one of the technologies driving the digital transformation of industry and industrial products. These products with high complexity based on multi-dimensional requirements need flexible and adaptive manufacturing lines and novel components, e.g., dedicated CPUs, GPUs, FPGAs, TPUs and neuromorphic architectures that support AI operations at the edge with reliable sensors and specialised AI capabilities.

The change towards AI-driven applications in industrial sectors enables new innovative industrial and manufacturing models. New process management approaches appear and become part of the core competence in the organizations and the network of manufacturing sites.

In this context, bringing AI from the cloud to the edge and promoting the silicon-born AI components by advancing Moore's law and accelerating edge processing adoption in different industries through reference implementations becomes a priority for digitising industry.

This article gives an overview of the ECSEL JU AI4DI project that aims to apply at the edge AI-based technologies, methods, algorithms, and integration with Industrial Internet of Things (IIoT) and robotics to enhance industrial processes based on repetitive tasks, focusing on replacing process identification and validation methods with intelligent technologies across

automotive, semiconductor, machinery, food and beverage, and transportation industries.

Keywords: Artificial intelligence, edge computing, industrial internet of things, deep learning, hardware, silicon technologies, silicon-born AI, computer vision

6.1 AI at the Edge in Industrial Processes

With more increased computing power, intelligent sensors and IIoT devices can collect large volumes of data these devices generate, reason over that data, and turn it into knowledge. AI can process this data closer to where it is produced and getting distributed to the edge. Multi-parameter sensing IIoT devices, AI everywhere, and serverless computing drive this new intelligent edge era.

Defining "intelligence" in the AI context requires a careful approach since different choices lead the research in different directions. The current field of AI is a mixture of multiple research fields, each with its own goal, methods, practical applications, etc. They are all called "AI" mainly for historical rather than theoretical reasons. Many AI definitions are provided in the literature (published papers, articles, books, research studies) to reflect the activities of research fields that the definitions mirror [2][3]. However, the definitions of AI systems are too vague and broad, requires further clarification.

As a starting point, AI in the context of the AI4DI project was defined as a machine's ability to collect information, perform logical analysis, acquire/produce knowledge, and adjust to an environment that varies over time or in a given context [1][5]. These abilities include the collective attributes of a machine (e.g., computer, robot, intelligent IIoT device, etc.) capable of performing functions such as learning, decision making, or other intelligent functions and tasks that mimic human behaviours [5].

The manufacturing industry is in transition, driven by Industry 5.0 concepts that transform the entire manufacturing value chain through a technology-driven change in capabilities and expectations. It is not simply about substituting people with machines, but instead about how people, interconnected sensors, machines, IIoT devices, distributed ledger technologies (DLTs), digital platforms, and AI can work together more effectively, using fewer resources and minimising the carbon dioxide footprint. Technological advancement drives manufacturers to increase productivity, efficiency, growth, deliver quality products, satisfy customers, and achieve higher profitability and sustainability. The Industry 5.0 AI-based driven processes change the tasks execution and impact manufacturing at the individual operation level. These digital-driven capabilities advance manufacturing across industries, value chains and value networks. That means that to remain competitive, manufacturers must adopt new AI technologies and integrate them into the manufacturing processes.

AI becomes a critical element to advanced manufacturing, product life cycle management and enterprise asset management.

Despite its potential, AI has several drawbacks that prevent the full exploitation of AI-based technologies. A few of these drawbacks are listed below:

- **Insufficient reliability and robustness of AI systems -** despite the numerous relevant technological advances, AI systems are still associated with low reliability, reflected in their relatively low penetration and utilisation.
- **High complexity -** the complexity of AI tasks has increased steadily to address new paradigms for automating, conceptualising, designing, and implementing such AI-based systems that include sensors, hardware, software, models, and algorithms.
- **High costs -** The development, implementation and deployment of AI-based solutions require vast investments as AI-based systems are unusually complex. Their repair and maintenance require significant effort. The AI systems call for frequent upgrades to meet to the changing environment's needs and make machines more "intelligent" day by day. In severe breakdowns, the procedure to recover lost codes and re-instating the AI-based system might require enormous time and cost.
- **Energy consumption -** AI models consume a relatively extensive amount of energy, and these energy requirements are increasing as AI technology is deployed in different industrial applications. Using deep learning (DL) algorithms, the computational resources needed to produce performant AI models increase significantly every year. In this context, AI has a significant carbon footprint, and if industry trends proceed, it will soon become considerably more severe.
- **Training data shortage -** AI-based models require large amounts of data, and their performance relies heavily on the size of training data sets available. For most industrial sectors, it is not easy to create training datasets that are large enough and include information that allow different industrial stakeholders to use the same data sets for

benchmarking similar AI models. These data sets exhibit tremendous potential for optimising industrial processes in cases where traditional approaches, like stochastics, analytical or numerical models, can no longer be used.

- **Absence of improvement with experience -** technical drawbacks related to the lack of progress with experience of the AI systems is a challenge. AI-based systems store a large amount of data. The data can be accessed and used differently by human and machine intelligence. Machines today can still not alter their responses to changing environments without being re-trained or updated/upgraded.
- **Lack of original creativity -** while AI systems can help humans create, they do not match yet the power of thinking that the human brain has or the originality of a creative mind.

The AI4DI project addresses several drawbacks mentioned above and provides solutions that enable AI to optimise industrial processes, energy efficiency, and processing at the edge.

6.2 A pan-European AI Framework for Manufacturing and Process Technology

The purpose of AI4DI is to benefit from recent research in the fields of semiconductor, intelligent systems, process control, IIoT, connectivity and edge computing to extend the potential of state-of-the-art AI technology for industrial applications to address the current main challenges in the industry:

- **Flexibility**: Factories and processes need to adapt to dynamic demand and compensate for failures quickly. AI can support and automate processes, planning, decisions, and system optimisation during all design phases.
- **Complexity**: Design processes, supply chains, manufacturing sites and the final products become increasingly complex. Managing this complexity can no longer be handled by humans. AI can reduce this complexity by generating simplified representations and even automating control. Importantly, this also includes the complexity of the AI system itself, which means that humans must interpret and understand its decisions and actions.
- **Process locally and think holistic**: Strong requirements for low latency, high reliability, and data privacy in industrial applications preclude outsourcing AI to existing cloud services. To successfully adopt AI in

the industry, AI must be deployed at the edge to support distributed on-site data processing with state-of-the-art AI components, algorithms, techniques, and methods.

- **Transparency**: The advance of digitisation will yield massive amounts of heterogeneous and possible unstructured data from many different sources. AI-based analysis and synthesis methods will be mandatory to analyse these large data sets and make them transparent for human operators and decision-makers.

Current AI tools are optimised for cloud services and therefore do not always fulfil the robust requirements of production environments (real-time capability, safety, reliability, guaranteed service quality, etc.). The AI4DI demonstrators provide and adopt AI methods and tools suited for moving the intelligence and analytics at the edge by addressing the following areas:

- **Hardware**: Develop AI-based microcontrollers, embedded systems and IIoT devices designed for industrial environments and edge processing. AI systems in the industry are implemented using a decentralised architecture with AI components distributed over a heterogeneous set of devices aligned with the software infrastructure.
- **Software and Libraries**: Adapt the existing software and libraries and develop new ones to address the critical requirements for AI in the industry regarding safety and reliability. As these requirements are not reflected in current AI tools, the project partners are extending the features of the existing tools, including appropriate workflows for testing and validation.
- **Algorithms**: Existing AI algorithms, in most cases, run on high-performance hardware. The AI4DI is advancing the optimisation of AI algorithms and models at the edge for running on IIoT devices and embedded systems with limited resources in terms of processing power, connectivity, and energy sources. The new optimisation methods and techniques for reducing the computational requirements are applied to neural networks (e.g., reducing the number of layers, adaptation to less precision) and quantify the impact of these optimisations on performance. New AI methods are investigated to improve the real-time operation capabilities and online learning.
- **Data**: Data and its quality play a critical role in AI industrial applications both for learning/training and testing the AI-based models and algorithms. In many manufacturing facilities, data is extremely sensitive and expensive to collect. Consequently, there is a strong need for an

explanation that can describe the needed information for various AI methods and techniques. Automated data anonymisation methods are required to allow data sharing for training without exposing confidential information.

AI4DI's mission is bringing AI from the cloud to the edge and making Europe a leader in silicon-born AI by advancing Moore's law and accelerating edge processing adoption in different industries through reference demonstrators [1]. The project focuses on five objectives to achieve its mission, as illustrated in Figure 6.1 and listed below.

1. Develop AI applications to be demonstrated under conditions as close as possible to real-life.
2. Formulate roadmaps, exploitation studies, business cases for AI technologies applications in industrial environments.
3. Provide a deployment plan showing how to develop and valorise AI technology in industrial sectors.
4. Establish an AI community in Europe, which is complementary to other initiatives.
5. Build and sustain dynamic AI technology ecosystems in Europe, ensuring ethical, responsible, and trusted AI for safety-critical real-time applications.

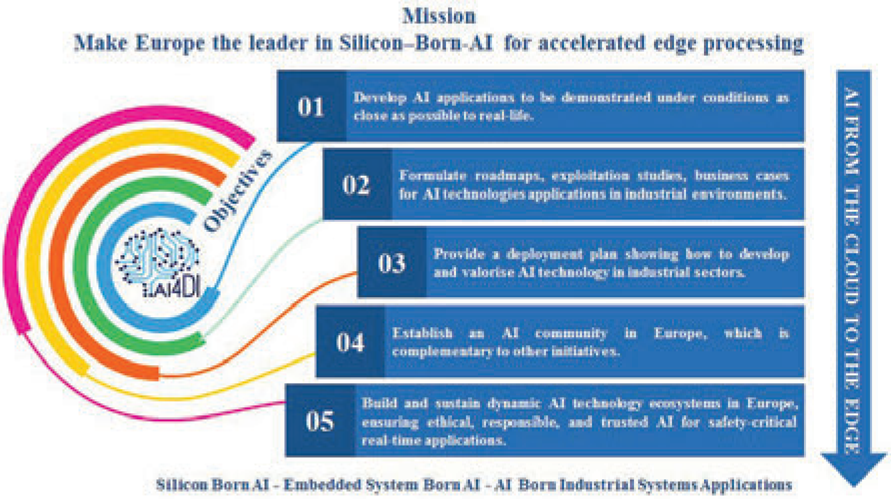

Mission
Make Europe the leader in Silicon-Born-AI for accelerated edge processing
Objectives
AI4DI
01
Develop AI applications to be demonstrated under conditions as close as possible to real-life.
02
Formulate roadmaps, exploitation studies, business cases for AI technologies applications in industrial environments.
03
Provide a deployment plan showing how to develop and valorise AI technology in industrial sectors.
04
Establish an AI community in Europe, which is complementary to other initiatives.
05
Build and sustain dynamic AI technology ecosystems in Europe, ensuring ethical, responsible, and trusted AI for safety-critical real-time applications.
AI FROM THE CLOUD TO THE EDGE
Silicon Born AI - Embedded System Born AI - AI Born Industrial Systems Applications

Figure 6.1 AI4DI Objectives.

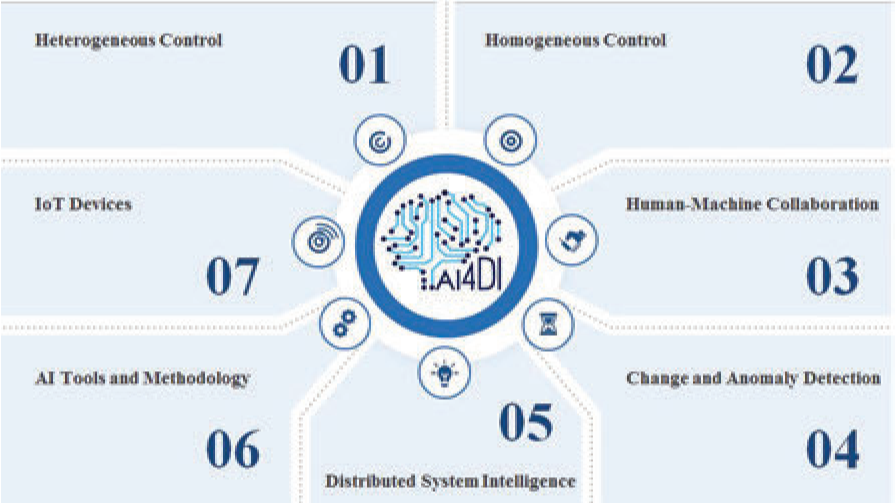


Figure 6.2 AI4DI Key Targets.

The AI4DI reproducible approach for implementing AI methods in the industrial sectors includes a structure that comprises seven key targets (KTs), as illustrated in Figure 6.2.

Each key target is a generic element representing a common characteristic of different processes, collaborations, methods etc., that is necessary for the successful cross-industry implementation of AI methods. A short description of each key target is provided in the following paragraphs.

Heterogeneous control - addresses the development and implementation of AI methods for heterogeneous control in manufacturing facilities. The main feature of heterogeneous production is assembling differentiable products from individual parts in a discrete manufacturing process with many different sequential steps. The exact sequence may vary from product to product and strongly depends on customer needs. Examples include the manufacturing of vehicles, aircraft, computers, and furniture. Various production steps make heterogeneous manufacturing highly complex and require human experts to identify and trace back issues. In this context, AI methods can collect knowledge, help to make transparent decisions based on current and saved data, and enable the optimisation of single production steps on the global level. One main task in the control of heterogeneous manufacturing processes is the scheduling of materials and production facilities. Production planning methods like material requirements planning rely heavily on correct data about the manufacturing facility, production history, predicted future required demands etc. AI methods have recently proven a considerable capability in generative modelling and can considerably contribute to more efficient production planning by providing more reliable external and internal factors models in the planning process.

AI can provide tools and methods for continuous quality monitoring at every process step and using the collected data to optimise the global control of the manufacturing process. Applications include automatic visual inspection with neural networks, pattern detection in measurement data, and proactive control based on predictive modelling. With distributed AI, devices already operating in the field can provide feedback directly available to the production process. A unique advantage of using AI to model and store data about the production process is that knowledge can be directly shared between the different plants. Introducing AI to control heterogeneous production processes become a fundamental prerequisite for managing increasingly complex products and growing product diversity (e.g., lot size one production in Industry 5.0). AI methods can model processes based on ontologies and knowledge databases. This information can be used to interpret data from the plant in an abstract, condensed, and semantically meaningful way, proactively recommending actions to the operator.

Homogeneous control - addresses the control of homogeneous production processes that manufactures products by combining raw materials from different supplies in a continuous and non-interruptible process. Typical examples of produced goods include pharmaceuticals, plastics, liquids, wine, and food. The various steps of homogeneous production processes are not isolated and usually irreversible. Control is therefore highly time-critical and directly related to the quality of the product. Very often, homogeneous processes are part of a larger heterogeneous manufacturing environment. The control of homogeneous production processes is based on continuous control loops often implemented as proportional–integral–derivative (PID) controllers. The parameters of these loops are directly related to the process output and the quality of the end product. Setting these parameters cannot be done locally only since changes at one stage of the system can affect all later stages due to the continuous nature of the process.

Moreover, specific sets of parameters can increase the overall robustness of the system at the global level even though local control loops might not be operating at the optimum level. The identification and dynamic application of these parameter sets require collecting and evaluating system data collected at all stages of the manufacturing process. Analysing this data, correlating them to favourable production parameters, and dynamically generating appropriate control parameters is a high-dimensional optimisation problem that is hard to track with traditional methods. Machine learning-based AI methods have achieved increased performance in control tasks but have not been applied in process control yet. Adopting these methods in industrial environments is highly promising and requires establishing standards to quantify important aspects such as reliability and real-time capability.

In large manufacturing facilities, homogeneous processes are typically observed in dedicated control rooms where experts have access to all sensors and actors in the system. Operating the control level requires a high level of expertise and concentration. Pre-processing and interpreting this data with domain-aware AI methods enable adding semantic information to raw measurements and identifying system states at an abstract level. Process control equipment in these facilities has specific safety and real-time requirements, and there is a strong need to move the required AI methods from the cloud to the edge.

Human-Machine collaboration - includes how AI methods can empower and enhance human-machine collaboration, including new ways of interaction, ethics, and value standards. The human-machine interactions in industrial environments can apply AI-based techniques locally at the edge close to industrial machinery by using latency communication and distributing intelligence between humans and machines on the industrial manufacturing floor. The collaboration approach is comparable to how an ants colony works, with each ant applying its separate intelligence towards the suitable solution of a common task. Examples of human-machine collaboration include spatiotemporal semantic awareness for enhanced worker safety, human-machine natural interaction for better design-to-manufacturing information transfer, correct instruction for acceptance of AI, and high-level understanding of continuous logistics flow on-premises and global level. The spatiotemporal semantic awareness for enhanced worker safety comprises cases, where machines are capable of a sophisticated semantic understanding of the environments and can move industrial robots to reduced risk towards the human worker, with less compromise on overall productivity. AI-based edge perception and activity recognition technologies applied at the level of the edge and IoT devices can be used to automatically understand the spatiotemporal relative position between workers and machines operating under his/her supervision, thus identifying potentially dangerous situations with ultra-low latency and much higher precision. The human-machine natural interaction for better design-to-manufacturing information transfer consists of on the edge node speech recognition AI-based solutions. Natural language processing provides a way to naturally interact with machines using voice-based commands even in very noisy environments. More advanced and friendly AI-based interfaces can improve human-machine collaboration on the manufacturing floor. Correct instruction for acceptance of AI includes advanced human-machine interfaces enabled by AI to deal with human knowledge. The correct instruction brings issues related to the ethical dimension, e.g., workers might be worried about the security and privacy aspects of interacting with an "aware" machine. Correct and transparent instructions on how to interact with AI-augmented devices encourage acceptance in the industry. A high-level understanding of continuous logistics flow on-premises, and global level relates to the capability to extract information at all manufacturing stages and understand complex logistic relationships to enhancing manufacturing facility-level productivity. AI performed at the edge provides a clear pathway in this direction, as too-abundant raw data can be locally characterised, inferred upon using AI techniques (ML, DL), summarised in a high information density format, and then immediately used to act accordingly. For industrial environments characterised by complex integrated value chains often spanning many geographically distant manufacturing facilities and industrial plants, utilising data analytics collected directly at the manufacturing level to smooth out logistics provides a significant productivity improvement.

Change and anomaly detection - addresses all methods and tools required to continuously monitor and analyse both heterogeneous and homogeneous production processes, both in real-time at runtime and offline after data collection. Tasks such as failure detection and quality control require in-depth domain knowledge and are usually limited to covering a predefined set of problem cases. Complex issues that depend on the specific production environment or a particular load profile are hard to detect and cannot be modelled in advance. More importantly, many changes in a production environment are not limited to a single stage in the production process but accumulate or spread over multiple phases. Detecting such changes becomes increasingly hard in complex production processes and can no longer be done by humans in dynamic reconfigurable factory environments of Industry 5.0. AI and ML are key factors for managing this complexity, and topics such as diagnostics and predictive maintenance, security and anomaly detection, distributed ledger are addressed. Diagnostics and predictive maintenance deals with the detection of failure or wear and tear on a single machine.

Usually, this task is carried out during maintenance or by a skilled worker operating the machine. Using time series analysis and prediction methods enables continuous monitoring based on either internal data recorded by the equipment or externally accessible state information that humans also use (e.g., vibration, sound, temperature sensing, image, video). While change detection can be computed on embedded devices directly at the machine, the actual training of the required models will, in most cases, rely on data analysis performed in the cloud. The results are transferred to edge devices that perform inferencing. Security and anomaly detection relates to the detection of intrusion and malicious system behaviour. In the ongoing digitising of production systems, more and more devices in the production process communicate via the network. The network connectivity is essential for many IIoT and AI-based tools and makes the system vulnerable to threats. Security is a significant issue in production systems and an important application for anomaly detection methods that detect deviations from a processing equipment's expected behaviour or output. For distributed systems used in future Industry 5.0 production environments, anomaly detection at the whole system level becomes particularly relevant since a single issue can propagate throughout the entire network. Distributed ledger offers the possibility of implementing a mechanism that guarantees the overall consistency of a distributed production system. This is particularly important for small lot sizes where the production process can differ considerably based on customer requirements. Implementing a distributed ledger enable guaranteed traceability of all process steps and is indispensable for safety- critical products. It also can increase the system's security since it allows for the direct detection of illegal operations. All mechanisms for change detection contribute to the awareness of the system and enable the detection of inconsistent, inefficient, and unsafe states. The detection performance heavily depends on the data available. Interfaces and exposed state data usually differ for every single machine, which makes the integration highly challenging. AI-based methods are a crucial contribution since they enable the processing of natural input data from IIoT devices that human experts also use to detect failure, wear out and other anomalies.

Distributed system intelligence - addresses the infrastructure on which the AI methods are implemented. Its focus is distributed intelligence systems, consisting of several subsystems taking each decision, with no need for central intelligence. As such, it constitutes one of the fundamental infrastructural blocks necessary for deploying the other key targets. The principal driver for the deployment of AI in complex industrial settings is collecting information, analysing it, and reacting immediately. Various pieces of machinery operate in concert towards the manufacturing of a single object or device, making it unfeasible to centralise intelligence in a unique central driver, as in the classical cloud-based AI model. Instead, like in an ant colony, a more effective strategy is distributing intelligence between decentralised actors located on the field, at the industrial machinery level. AI embedded in the edge devices and directly in the many IIoT sensors deployed inside the industrial equipment can progressively build hierarchical, advanced representations of the raw data collected. Edge and IIoT devices augmented with capabilities to perform AI-based functions for vision, perception, sensor fusion etc., can collect information, distil it in a high information density format, and then transfer it to a more performant on-premises edge processing capable of using this information for building a context awareness and acting accordingly to it. The edge processing unit might then be reporting to a centralised broker for data collection or a cloud-based AI service. However, the main functionality of the system is deployed locally, at the edge, enabling advanced low-latency local behaviour such as human-machine cooperation and change detection.

AI tools and methodology - addresses the development of tools and methods required for implementing AI in production systems. The focus is on establishing a toolchain that moves AI services from the cloud to the edge and applies them to safety-critical domains with real-time solid capability and reliability requirements. This toolchain includes the technical implementation and a principled methodology that allows system developers and integrators to apply AI in their specific domain. The fast progress of AI provides tools that helped speed up research and development by standardisation and state-of-the-art open-source methods to a broad community. Software tools like Google TensorFlow have set a standard that enables simple code sharing and quick reproduction of results. The development is fuelled by openly accessible project repositories such as GitHub, free databases with training data such as ImageNet, and dynamic training environments such as OpenAI Gym. Current AI tools are optimised for research or cloud services and therefore do not fulfil the robust requirements of production environments (real-time capability, safety, reliability, guaranteed service quality, etc.). Adopting AI methods and moving them from cloud to edge therefore addresses the areas such as hardware, software/libraries, data, and algorithms. The AI-based industrial equipment requires hardware that addresses the factory floor's needs (e.g., high-performance, energy efficiency, reliability, real-time capabilities etc.) and can be integrated into IIoT devices. AI systems in the industry are implemented in decentralised architecture and distributed over a heterogeneous set of devices, which define the requirements for the software infrastructure. Software and libraries are needed to address the safety and reliability requirements for AI-based applications in the industry. These requirements are not reflected in current AI tools, and deep analysis is required to identify and add missing features, including appropriate workflows for testing and validation. Data from manufacturing facilities is critical for the training/learning of AI-based models and algorithms.

In many cases, the industrial data is classified and costly to collect. New AI methods and techniques for data anonymisation are needed that allow data sharing for training without exposing confidential information. AI algorithms and run in many applications on high-performance hardware. To enable analytics and processing at the edge, optimisations of these algorithms are needed to run at the edge on resource-constrained IIoT devices without compromising the performance. This also applies to the investigation of AI methods capable of real-time operation and online learning. Introducing AI methods requires a structured methodology for selecting appropriate AI tools for a given task and integrating them following the requirements of the domain [4].

IoT Devices - considers the hardware/software aspects related to the practical implementation of the other key targets on IIoT devices deployed in industrial machinery. This includes components for sensing, actuating, connectivity, and end node IoT processing.

In the industrial scenario, the primary constraint about IIoT sensor/smart devices is their physical footprint. Sensor devices must be placed directly within parts of moving machinery, putting severe limitations on their size and capability to be wired. A second constraint is that these devices need long battery life and require "zero" maintenance, ideally outliving the piece of machinery mounted on with "zero" human intervention. Fulfilling both constraints require sophisticated and state-of-the-art IIoT device design. Traditional architectures for these end node devices leverage small-scale microcontrollers (e.g., ARM Cortex-M0 class) to minimise the compute power envelope, coupled with button batteries to ensure long life and minimal footprint. End nodes of this kind can collect sensory information, send it to a central server via wireless communication and go back to sleep.

Implementing distributed intelligence and deploying advanced AI-based functions directly on the IIoT devices requires much higher performance available on the device, which makes fitting within the constraints discussed above much more difficult. Therefore, devices designed within the project must use advanced high-performance, AI-dedicated/capable microcontrollers with much better power efficiency techniques. The wireless capabilities need to include energy-efficient communication and exploit an overall more advanced architecture, integrating multiple sensors with embedded in-plane analytics and dedicated hardware architectures for AI and inference.

6.3 AI Technologies

The success of industry-grade embedded AI on edge devices directly depends on the availability of dedicated central processing units (CPUs), graphics processing units (GPUs), and hardware accelerators architectures for electronics components and systems that are powerful enough to fulfil the full potential of state-of-the-art AI methods.

By extending Moore's law, as illustrated in Figure 6.3, silicon-born AI takes full use of advances in semiconductor technology and even improve

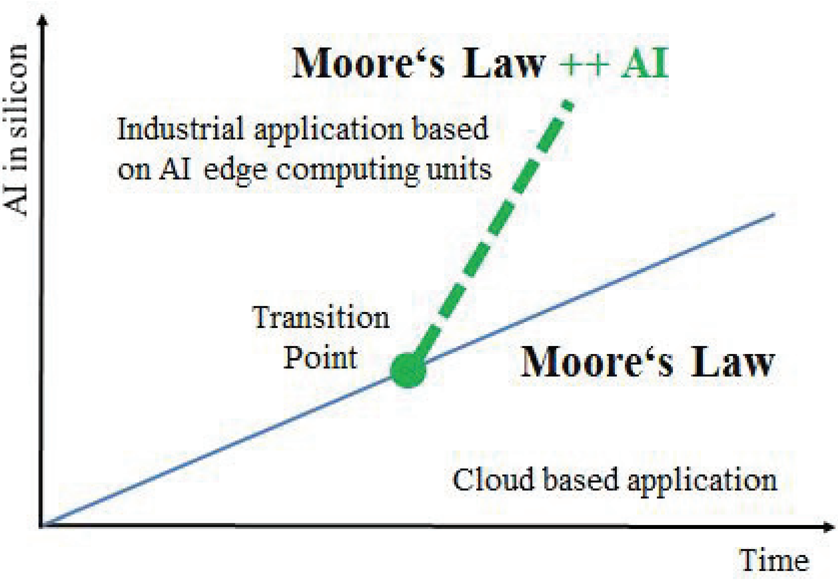


Figure 6.3 Silicon-born AI effect on Moore's Law beyond the current silicon technology developments.

the performances by leveraging additional scaling effects of “More Moore” and “More than Moore”. The adoption of AI technologies in the industry can be substantially accelerated by making the required compute power available at the edge and thereby enable completely new AI applications that are not available for industrial applications yet.

In the short term, AI is implemented on readily available AI capable CPUs that include essential support for accelerating AI models (e.g., extension for vector operations). This is realised by model compression techniques that can reduce parameters in neural networks by factors of up to 50.

Dedicated AI methods can be implemented purely in software to enable industry-grade AI for data processing at the edge. Industry-grade AI must comply with the robust requirements of industrial applications and can therefore not be implemented by applying only the existing AI algorithms to industrial tasks.

AI techniques and methods are optimised to execute on AI-based hardware architectures designed for industry and featuring enhanced connectivity capabilities for fast communication in distributed networks of embedded IIoT devices.

However, growing amounts of data and new AI methods require larger models and more AI performance support. In the mid-term, “More Moore” will enable the design of AI-enhanced processing units that handle more data and larger models at lower response times.

Dedicated silicon-born AI hardware components, design languages, application generators, design automation tools, and respective standardisation can address AI features directly in the chip design to leverage performance speedup through the advancement of Moore’s law.

“More than Moore” technologies allow heterogeneous functional processing units on a single chip in the long term. This makes novel neuromorphic processing units (NPUs) tailored to the execution of large- scale neural models in real-time with maximum power efficiency available for industrial applications. Neuromorphic computing can also support future brain-inspired AI technologies.

More dense system integration enabled by “More Moore” technologies increases the AI algorithms performance, and more heterogeneous integration enabled by “More than Moore” technologies increases the AI functionalities. Both enlarge the potential for industrial application of AI even further.

The silicon-born AI maximises the benefits of Moore’s law and revives it beyond the current semiconductor/silicon technologies while enabling native AI computing and the native embedding of AI algorithms directly in silicon.

Through this approach, progress in semiconductor technology automatically translates to better performance for AI applications running on embedded edge computing. “More than Moore” technologies additionally address the integration of AI-specific computational units and sensor/actuator devices on a single chip and thereby also accelerate the speedup of silicon-born AI for industrial applications.

Implementing a roadmap that builds on silicon-born AI supports and accelerates the adoption of AI by European’s industry to address its most urgent priorities in digitisation, such as mastering complexity, increasing flexibility, maximising efficiency by moving the intelligence to the edge and providing new distributed reference architectures [6] that are aligned with the industrial requirements. The demand for high AI performance is fuelled by technological (e.g., intelligent sensors/IIoT devices generating more useful data) and industrial factors (e.g., moving from linear to network processes).

The new industry-grade AI methods require to be tailored to the European industry’s specific needs, and the development of AI-based hardware keep up with the growing demand for AI through the advancement of Moore’s law.

6.4 AI Application Areas

Various industrial sectors are currently experiencing the most radical changes since assembly lines and the rise of mass production. Product complexity is continually increasing while customers simultaneously demand individually configured and manufactured products.

Many industrial AI systems are built around a centralised paradigm where machine learning solutions are delivered as a part of cloud-based APIs and software packages deployed on remote servers of AI providers. The future requires a paradigm shift by moving toward decentralised and distributed AI that can run and train at the edge on local intelligent devices in industrial applications or make decisions in decentralised networks like blockchain. The transition to decentralised and distributed AI is enabled by new technologies that allow for crowd-training of ML algorithms, device-centred AI that runs and trains ML models on mobile IIoT devices, and AI in decentralised autonomous organisations on heterogeneous networks.

Intelligence on an edge device allows it to process information locally and respond quickly to situations instead of communicating with a central cloud or server. For instance, an autonomous AI system must respond in real-time to what’s happening on the production line. Decisions are time-sensitive, and latency is critical for many mission-critical industrial processes.

The AI4DI provides AI-based technologies at the edge for digitising the industry by reducing costs, save time, optimising/improving processes/products/services, increasing quality by enhancing industrial processes, and built and sustain a dynamic AI technology ecosystem in Europe.

The project develops IIoT technologies, AI-based hardware, software, models, and algorithms to enhance processes based on repetitive tasks, focusing on replacing process identification and validation methods with intelligent technologies across automotive, semiconductor, machinery, food/beverage, and transportation industries.

The following sub-sections provide an overview of the topics covered in the five industrial sectors. The different use cases are presented are presented at different levels of detail in [7].

6.4.1 Automotive

Digitisation is an essential prerequisite for tracing the production process along the entire supply chain and enabling future innovations in the automotive manufacturing industry. Growing data and new non-linear manufacturing paradigms yield massive data sets that humans can no longer interpret. AI, therefore, becomes an essential tool for processing this data. It will accelerate the automotive industry also have a significant impact on automotive companies' finance and control. Maximum data transparency is essential for AI-enabled analysis, optimisation of automotive production processes and supply chains. When the required data is available in real-time, the potential of AI methods such as DL, ML, expert systems and distributed autonomous agents is enormous. So far, the planning and operation of automobile production processes still require human planning and could be made more responsive and automated with AI.

Improving the responsiveness and automation using AI-based technologies covers the complex logistic processes across deep supplier networks with the potential of optimising the complete supply chain, including prediction of future system states or even autonomous control.

AI4DI addresses the AI-based technologies and applications for optimising logistic processes to reduce transport costs and the environmental impact.

The AI-based technologies and applications in the automotive industry cover two main areas, AI-supported automotive manufacturing and logistics and real-time predictive maintenance. A list of the demonstrators implemented under each application areas is presented below:

Inbound logistic process optimisation

- *Inbound logistics process optimisation* - addresses the systemic analysis and decision-making for responding to critical supply chains. Different data streams are injected into the AI core to react to disruptions as quickly as possible with suitable measures.
- *Assembly process optimisation* - based on computer vision systems and deep learning methods, ensures correct installation, and enables an ergonomic evaluation of the workers' activities.
- *Autonomous reconfigurable battery system* - aims to combine various retired batteries with very heterogeneous performance characteristics within one battery system. For this purpose, it is essential to accurately determine the state parameters, like the state of health and state of charge (SoC) of each single battery cell during the operation.
- *Virtual AI training platform for robot learning* - uses reinforcement learning (RL) to address the challenge of bringing autonomy in industrial robotic manipulation. Advanced simulations are used to virtually train the policy network by providing multitudes of realistic synthetic data.
- *Bluetooth low energy (BLE) localisation in asset tracking* - focused on indoor asset tracking based on Bluetooth wireless technology. The functionality of low-cost and BLE based components are enhanced by AI technology designed to analyse the non-deterministic signal received from tracking tags.
- *Autonomous mobile robotic agent* – addresses a multi-purpose robotic platform for indoor use intended for autonomous transportation of the factory's material, goods, or tools. AI algorithms deliver autonomous and cooperative behaviour even in complex environments, and AI trajectory planning manage distributed intelligent traffic control ensuring fast and reliable delivery within the factory.

Real-time predictive maintenance

- *Predictive health-monitoring system for machines on the level of a digital twin* – addresses a combination of AI methods and mathematical damage models connected to the operation of an e-motor unit for real-time failure prediction and diagnostics. To evaluate and monitor the asset's current health status, the system processes operation data in real-time at the edge to detect anomalies and conclude upcoming failure occurrences or required maintenance actions.

6.4.2 Semiconductor

The AI technologies open various opportunities for semiconductor manufacturing by using AI-based systems in different co-existing models in the datacentres and on-premises at the edge and embedded in the semiconductor manufacturing equipment. These AI-based systems optimise and improve the efficiency of processes for different semiconductor technology nodes and support the acceleration of the design and manufacturing of multiple hardware architectures (e.g., CPU, GPU, NN accelerators, FPGA, dedicated ASICs, etc.), addressing a large set of heterogeneous applications.

AI-based technologies support semiconductor manufacturing facilities optimise and improve efficiency during the research and chip-design phase. The AI methods are used for eliminating defects and out-of-tolerance process steps that can decrease/avoid time-consuming iterations, accelerate yield ramp-up, and lower the costs required to maintain yield. The AI-based techniques are used to automate the time-consuming physical layout design and verification processes.

The AI-based technologies and applications in semiconductor manufacturing industry, address the following areas AI-based failure modes and effects analysis (FMEA) generator, AI-based 3D inspection for quality assurance, fault package detection, automatic interpretation of scanning electron microscope (SEM) images from semiconductor devices, silicon package fault detection and digitised support for product definition. A list of the demonstrators implemented under each application areas is presented below:

AI-based FMEA generator

AI-based FMEA assistant – development of an FMEA assistant tool to support the engineers to analyse the existing information efficiently. FMEA assistant is created by using existing data from the manufacturing process like structured or semi-structured FMEA, Failure Analysis (FA), 8D documents, and other domain-specific unstructured texts like production tools manuals, handbooks, and process descriptions patents and similar.

AI-based 3D inspection for quality assurance

- *Neural network for predicting critical 3D dimensions in MEMS inertial sensors* – addresses the use of ML to predict product parameters of inertial sensors, which are determined by the 3-dimensional shape and dimensions of the MEMS device. Data is collected from several process sources, including product measurements in various process steps and processing machine conditions.

- *Machine vision system developed in the wafer inspection production line* - processes the microscopic images of semiconductor wafers and detect surface defects, providing the results in a readable form, either in a table with coordinates and size of each defect or in the form of a heatmap of defect location on a wafer.

Fault package detection

- *Wafer fault classification* - provides a device-integrated solution for the wafer classification problem. The device gets pictures from a camera and perform real-time data analysis, giving the category to which, the wafer default belongs and binary faulty/non-faulty information.

Automatic interpretation of SEM images from semiconductor devices

- *Automatic inspection of SEM cross-section images for technology verification* – addresses a fully automated measurement toolchain. Research focuses on computer vision tasks and additionally on methods for automated analysis techniques of semiconductor front-end technologies.

Silicon package fault detection

- *Anomaly detection on wire bond process trace data* - covers the supply chain's relevant functionalities: developing the AI-based model, deployment, and visualisation. The work addresses the limitations regarding the availability, scalability (number of eq. and recipes) and degree of integration into the production data landscape.
- *Optical outgoing inspection* - provides an optical inspection solution working on the same or similar hardware and software environment providing anomaly detection with a pre-trained neural network (NN) for detecting deviations, image labelling for supervised learning and deployment of the AI-based model for image analysis and prediction.

Digitised support for product definition

- *Digitising product definition* – addresses the assessment of product definition via automated application simulations as planned via ML and formulate requirements human- and machine-readable to boost automation in the design and development phases.

6.4.3 Industrial Machinery

AI technologies are becoming a necessary part of manufacturing and automation across engineering, operations, and maintenance in the machinery and industrial equipment industry. AI applications start to be used more in high-end machinery and gradually migrate toward simpler machinery, such as palletisers and packaging machines.

In the machinery and industrial equipment industry AI is deeply embedded in the controller, the engineering tool, or the devices controlling the manufacturing line. The AI embedded solutions used for taking decisions and replacing the programmable logic controller (PLC), require techniques that are near 100% transparent to be accepted in a conservative industry such as industrial automation. The growth of industrial personal computers (IPCs) in manufacturing transformed what AI-based PLCs are capable of. In the machinery and industrial equipment industry, AI can be integrated into engineering and programming tools with embedded natural language processing (NLP) autocorrect features or automatically suggesting code and changing programming controllers. Using low-cost, robust and energy-efficient high-performance AI chips, AI becomes a necessary part of the controllers in the automated production lines.

New approaches to computer vision and ML open a variety of possibilities in industrial automation to optimise processes and improve the safety of human operators in the industrial environment. The areas of improvement cover areas from detecting defects of the goods and the erroneous behaviour of machinery to the detection and classification of all the objects present and acting in the working area.

The supply chain in machinery and industrial equipment develops to integrate the support of DL in the industrial environment, allowing the dynamic adaptation of the behaviour of the machinery with a re-training of the AI support on cloud level and the deployment at the edge of many precise high-efficient devices for the local processing and analytics. The interaction with the machinery and the data retrieved from different types of sensors and IIoT devices produce a consistent amount of data processed by the AI-based services to improve machine learning and the continuous re-training of the AI-based embedded modules.

The AI-based technologies and applications in the industrial machinery industry comprise two main areas, wood machinery with innovative human-machine interface (HMI) and smart robots. A list of the demonstrators implemented under each application areas is presented below:

Wood machinery with innovative HMI interface

- *Wood machinery with the perception of the surrounding environment* – addresses the use of specific sensors (e.g., ultrasonic sensor grid sensors) to detect the presence of obstacles near a woodworking machine, slowing down or stopping the machine's cabinet in case of detection. AI-based techniques are used for the refinement of the sensing and detecting capabilities to guarantee a higher level of reliability of the detection.

Smart robot

- *Smart robot* - addresses how to enable robots to "see", "feel", and interface with humans and the environment around them using a universal multi-modal cognitive sensing platform providing synthetic real-life like data generation for AI-training, intuitive human-machine interaction, and usage of Robot Operating System (ROS) for adaptability of different industrial robots, sensors, and other equipment.

6.4.4 Food and Beverage

The implementation of IIoT and robotics solutions in the food and beverage industry sector has supported overcome critical issues related to production and execution by eliminating the possible human errors while reducing the redundancy in work performed by manual labour. AI fuels innovation in the production and packaging of food and beverage to reach expectations regarding the quality of the products delivered to the consumers and their related impact on the cost. To attain the potential trade-off between quality and price, industry stakeholders are actively leveraging the potential of AI across various applications, such as product design, quality control, maintenance, and consumer engagement, among others.

The integration of AI technology increases the efficiency improvements in the food and beverage industry, with significant reductions in downtime, repair costs, and additional labour requirements and cost. Companies in the food and beverages production and manufacturing industry leverage the benefits of AI through the use of NNs, ML techniques, advanced analytical tools, combined with image recognition and computer vision technologies for optimising the manufacturing processes.

The food and beverage processing lines include continuous monitoring IIoT technologies used in the predictive maintenance process that collects real-time data from multiple and varied IIoT sources placed on motors/equipment, combines them and uses ML techniques to anticipate equipment failure before it happens. Predictive maintenance of production machinery is for instance based on sound or vibration analysis computed directly by IIoT devices and vision-based quality control of the product at the edge for production process optimisation. Parts of the collected data is sent to a cloud service that continuously optimises a detection model. Control applications directly influence the production process and are therefore especially critical considering their real-time capability and reliability.

The AI-based technologies and applications in the food and beverage industry comprise two main areas, beverage production - Champagne and food production - soya beans. A list of the demonstrators implemented under each application areas is presented below:

Beverage production - Champagne

- *Environmental monitoring system* – addressing the implementation of an industrial monitoring system that enables an accurate analysis of the production process in the vineyards and caves. The data obtained by this monitoring infrastructure enable accurate decision-making accordingly to an external environmental condition that can impact the production step under analysis.
- *Autonomous environment-aware* - addresses the implementation of an AI-based method of capturing images and data using an autonomous robot to support cameras and sensors. The data analysis from the vineyards allows precise decision-making regarding the yield, vine diseases and missing vines.
- *Quality control system* – addresses the setting up an image acquisition system in the Champagne presses facilities. The data analysis from the presses allows the neural network training based on the quality classification of the grapes.

Food production - Soya beans

- *Production process optimisation* – addresses soybeans production process optimisation using IIoT-based sensors for visual analysis, temperature, humidity, and moisture throughout the preparation phase and correlates real-time data in those parameters using AI-based models.
- *Predictive maintenance* – addresses a solution for implementing an intelligent monitoring system that separates the equipment's normal condition from abnormal conditions. IIoT-based sensors are installed to measure different parameters such as vibration, current, sound, temperature etc. Data from the IIoT devices are sent to AI-based models that correlate with normal and abnormal conditions and implement a predictive maintenance solution.

6.4.5 Transportation

Mobility-as-a-Service (MaaS) based on vehicle sharing changes people's transport habits and introduces new mobility modes. Future automated vehicles enable 24/7 driving and serving people with fewer vehicles on roads. MaaS steps taken without automation aim to improve the availability of public transportation when needed.

The main challenge with public transport is the mean of transportation available when needed. Buses, trams, etc., are bind to the schedules and cannot offer ad-hoc on-demand solutions.

Therefore, on-demand taxis and mass transportation have great potential to change people's mobility in cities and rural areas where the ageing population is suffering from available transport services when needed. In addition, fewer vehicles and buses mean fewer pollutions thus, leading to better transport sustainability.

The application of AI in the transportation industry is accelerating the next generation of Intelligent Transportation Systems (ITS). Intelligent edge computing technology supported by high-speed connectivity is used to process AI decision-making at the vehicle and edge level without connecting to a server in the cloud.

AI technologies in traffic management enhance the efficiency of the mobility systems it integrates with and play a significant role in developing and deploying new and innovative environmentally friendly solutions to operate vehicles for travel and transportation.

The AI-based technologies and applications in the transportation industry cover one central area, MaaS development of AI-based fleet management for supporting multi-modal transport. The demonstrator implemented under this application area is presented below:

MaaS, development of the AI-based fleet management for supporting multi-modal transport

- *MaaS - AI-based fleet optimisation tool* – addresses an AI fleet management of MaaS solution, in which two automated last-mile vehicles are controlled according to the transport demands of the users. The data processing is done in the vehicles and the infrastructure by reducing links to the cloud and increasing decentralisation. Novel computation platforms are utilised for accelerated processing. A neural network (NN) based analysis program for predicting travel times is implemented. The machine learning-based, improved data pipeline leads to improvements in terms of waiting times for passengers.

6.5 AI Technology Roadmap for Digitising Industry

The transition towards a more integrated technology converging combining AI with other cutting-edge technologies like IIoT, edge computing, and connectivity is essential to reframe the challenges that the European electronic components and systems community need to tackle in the future. AI4DI stakeholders drive activities for community consensus and take the lead on the compilation of multi-annual research roadmaps addressing human-centric AI technologies aligned with roadmaps of other related technologies. The roadmap guides how the European electronic components and systems community can obtain a competitive advantage in designing, developing and deploying silicon-born AI, AI-based embedded systems in industrial sectors.

Building the AI roadmap results from exchanging ideas and concepts at the European level and aligning the work with activities at the global level. The approach for interaction with related industrial sectors is primarily focused on analysing existing challenges and gaps of the related technology areas and specific workshops exploring the intersection of AI technologies with key stakeholder groups.

The perception of the AI technologies by European citizens and the industrial sectors that it affects play an essential role in the broader adoption debate of AI technologies. Industrial AI solutions may lack direct consumer scrutiny, but they are under the evaluation of the industry stakeholders that strongly and robustly influence the sector regulations and standards.

The AI4DI work on the AI road-mapping activities provides an excellent framework for the industrial stakeholders to prioritise resources and align the vision of the electronic components and systems community to focus on essential breakthroughs to reach the next level of AI technology evolution for digitising industry.

The shift of AI methods from cloud to edge is the primary approach of AI4DI for digitising industries and marks the starting point of a comprehensive transition regarding the control of industrial processes and functionality of devices. The AI4DI roadmap lists the significant milestones of this transition driven by AI methods operating on the edge.

The AI technology developments influence the evolution of IIoT devices, silicon-born AI, embedded systems-born AI, AI methods, models, algorithms, and integration in the manufacturing processes. The integration of complex AI-based systems is highly linked and dependent on all these elements. The increase in computing power and industrial user experience with single AI methods supports integrating interconnected machines and automated data analysis and optimisation of processes. This development enables the transition from linear manufacturing processes to AI-controlled value chains as a network of many flexible interconnected machines as part of a distributed production line. More powerful AI methods implemented on IIoT devices at the edge increase these devices' functionality aiming at self-learning and self-optimisation features. The ongoing parallel development of AI methods for cloud applications plays a significant role in developing efficient processes and functional devices at the edge along with the roadmap. Implementing industry-grade embedded AI on edge devices directly depends on the availability of dedicated silicon-born AI architectures for electronics components and systems that are powerful enough to fulfil the full potential of state-of-the-art AI methods.

6.6 Conclusion

Intelligence on the edge devices in industrial environments allows it to process information locally and respond fast to situations instead of communicating with a central cloud or server.

AI raises new ethical and legal questions related to liability or potentially biased decision-making in industrial environments. AI4DI actively supports the activities for progressing ethical guidelines on AI development in industrial environments by guiding the industrial stakeholders on the new challenges brought by AI and the interpretation of the liabilities in the light of technological AI developments to ensure legal clarity for industrial consumers and producers.

The article gives an overview of the ECSEL AI4DI project that develops AI-based solutions to bring intelligence processing from the cloud to the edge by providing intelligent technologies across industrial sectors such as automotive, semiconductor, machinery, food and beverage, and transportation.

The project aims to provide AI-based technologies at the edge for digitising the industry by reducing costs, saving time, and increasing quality by enhancing industrial processes. The project's advancements enable optimising/improving industrial processes, products, services, and support building and sustaining a dynamic AI technology ecosystem in Europe.

Acknowledgements

This work is conducted under the framework of the ECSEL AI4DI "Artificial Intelligence for Digitising Industry" project. The project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 826060. The JU receives support from the European Union's Horizon 2020 research and innovation programme and Germany, Austria, Czech Republic, Italy, Latvia, Belgium, Lithuania, France, Greece, Finland, Norway. The authors want to express their great appreciation to Dr Cristina De Luca for her valuable and constructive ideas, suggestions and inputs during the preparation and writing of this chapter. She was the ECSEL JU AI4DI project coordinator, working for Infineon Technologies AG until March 2022, and her dedication and hard work supported the project to obtain the best results.

References

[1] ECSEL AI4DI project. Artificial Intelligence for Digitising Industry. Available online at: https://ai4di.eu/

[2] The Artificial General Intelligence Sentinel Initiative (AGISI), "A working list: Definitions of Artificial Intelligence and Human Intelligence". Online at: http://agisi.org/doc/AGISI_DefinitionsIntelligence.pdf

[3] P. Wang (2019). "On Defining Artificial Intelligence" Journal of Artificial General Intelligence, Vol.10, No.2, 2019, pp.1-37. https://doi.org/10.2478/jagi-2019-0002

[4] AI4DI Deliverable D1.12 - Requirements and specifications for smarter food and beverage production based on AI-technologies. 2019

[5] AI4DI Deliverable D1.2 - AI for digitising industry road-mapping requirements. 2020

[6] AI4DI Deliverable D2.1 - Report on hybrid reference system level architecture design for the digitising industry. 2020

[7] O. Vermesan, R. John, C. De Luca, and M. Coppola (2021). Artificial Intelligence for Digitising Industry - Applications. Gistrup: River Publishers. Available online at: https://www.riverpublishers.com/pdf/ebook/RPE9788770226639.pdf

7

Impact of AI and Digital Twins on IIoT

Bin Han[1], Björn Richerzhagen[2], Hans Schotten[1], Davide Calandra[3], and Fabrizio Lamberti[3]

[1]Technische Universität Kaiserslautern, Germany
[2]Siemens AG, Germany
[3]Politecnico di Torino, Italy

Abstract

We discuss the role and impact of AI on the Industrial Internet of Things (IIoT) as envisioned by the European flagship project on 6G, Hexa-X. The envisioned ecosystem of trustworthy collaborative digital twins (DTs) lays the foundation for emergent intelligence (EI) and utilization of AI for industrial scenarios. One important building block for utilization of AI in IIoT is the inclusion of the human: we therefore provide insights on AI at the intersection between DTs and human-machine interfaces (HMIs).

Keywords: Hexa-X, industrial internet of things, digital twin, artificial intelligence, emergent intelligence, human-machine interfaces, immersive technologies

7.1 Introduction to the Hexa-X Project

Hexa-X[1] is the European flagship project on 6G. It defines the vision, use cases, as well as key performance and value indicators for upcoming 6G systems. The project studies technical enablers for novel 6G capabilities and provides an initial end-to-end architecture for 6G systems. The key

[1] hexa-x.eu

societal values of sustainability, trustworthiness, and inclusiveness drive the contributions in the project [1].

Use cases considered in Hexa-X span seven families, as detailed in [1]. The use case families *from robots to cobots* and *massive twinning* capture key characteristics of IIoT for which technical enablers and concepts are being developed in the project. In the following, we focus on DTs and the use of AI and novel HMIs as technical enablers and their impact on IIoT. First, we outline an ecosystem concept for DT that highlights the relations between twinned aspects of the IIoT and the underlying information flow enabled by a 6G system with its novel sensing and processing capabilities. We discuss the concept of EI being enabled by (collaborating) DTs and its impact on IIoT and elaborate on the potential of collaboration among local and global management entities and their respective DTs to benefit from additional local insights in AI-based decision making and optimization. Before concluding the discussion, we analyse the role of AI at the intersection between DTs and novel HMIs.

7.2 An Ecosystem Concept for Digital Twins in IIoT

With the massive deployment of DTs, in the era of 6G, conventional cyber-physical systems (CPSs) that have been widely used in industrial scenarios is envisaged to evolve into a human-centric industrial ecosystem, which is illustrated in Figure 7.1. With a generic framework to support constructing and maintaining a digital replica for an arbitrary physical entity, it allows every machine, every person, and every component of the data infrastructure that is involved in the industrial process to offload its context information to the digital intelligence (which is commonly deployed in the cloud or at the network edge), analyse it online, and exchange such information with other involved entities or DTs in an agile, efficient, and secured fashion.

To support such an ecosystem, future IIoT must leverage the numerous advantages and conveniences provided by 6G DTs, which are including, among others: the ubiquitous and ultra-dense connectivity to support massive twinning; the timely status synchronization between the physical entities and their DTs; the data-driven intelligence that generates empirical insights on the physical environment and processes. Empowered by these technical enablers, various novel use cases can be envisioned, which we have clustered into eight categories upon the flow of information between the cyber and physical/human worlds, as shown in Figure 0-1. In the following sections of this chapter, we will focus on three selected technical aspects to demonstrate

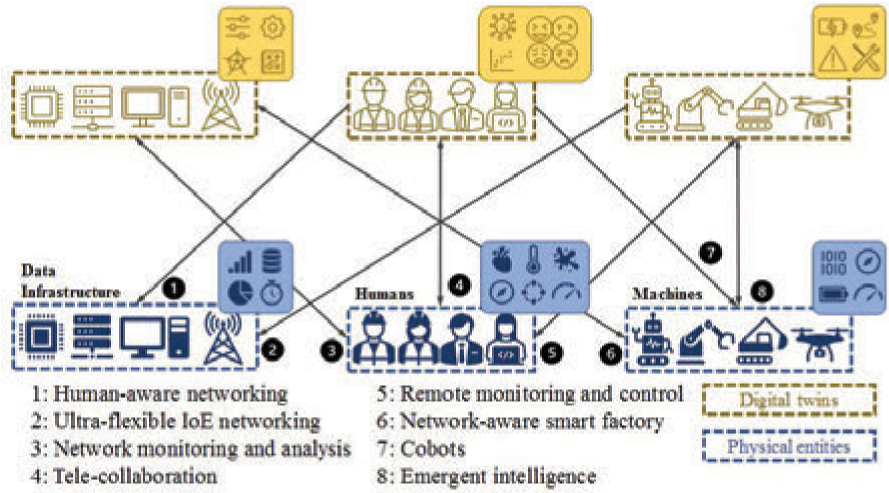

Data Infrastructure
Humans
Machines
1
2
3
4
5
6
7
8
1: Human-aware networking
2: Ultra-flexible IoE networking
3: Network monitoring and analysis
4: Tele-collaboration
5: Remote monitoring and control
6: Network-aware smart factory
7: Cobots
8: Emergent intelligence
Digital twins
Physical entities

Figure 7.1 The ecosystem of 6G human-centric industrial DTs, with the arrows indicating the direction of the information flow.

how the combination of AI and DT will impact these emerging use cases. More specifically, we will introduce 1) what is Emergent Intelligence (EI) and why it benefits from massive twinning; 2) how network-aware DTs can be used to generate local insights to support smart factory applications; and 3) how AI-empowered DTs can be exploited in human-machine interface to realize collaborative robots (cobots) and Extended Reality (XR).

7.3 Digital Twins for Emergent Intelligence

Future IIoT is envisaged to connect everything and everybody, not only the physical entities but also their DTs. Rich physical and context information can be therewith efficiently collected, shared, and exploited. Such ubiquitous interconnection and universal information sharing among equipment, products, infrastructure, and human participants will help to deliver an immersive AI capable of accomplishing future industrial tasks, which are not only complex, but also polymorphic and flexible (e.g., the manufacturing process may vary significantly from one product to another, and an occasional update of the AI solution is demanded in the future flexible manufacturing scenario).

Nevertheless, while promising numerous new use cases, the immersive AI in 6G IIoT is also raising concerns in safety, security, and data privacy. Most conventional AI solutions require the aggregation of user data at a central unit that makes decisions for all users as the AI engine, which leads not only to a concern of privacy leakage, but also a security risk of model manipulation through malicious data injection. Over the recent years, technologies like Federated Learning (FL) have been intensively studied and well developed to address the privacy concern in AI by distributing the responsibilities of data aggregation and model training to agents. Nevertheless, they cannot yet eliminate the risk of manipulation due to their central-model-based nature. As a model-less mechanism to implement complex system behaviour, EI may play an important role in future AI applications as a secured, privacy-intolerant alternative and complement to conventional solutions such as FL.

The concept of EI was first proposed in the late 1980s as a biological term, which describes the intelligence of animals originating spontaneously and emergently from many simple units that are interconnected and interacting with each other in a complex manner [3]. Thereafter, this phenomenon was rapidly noticed in the engineering field and has inspired to develop bionic intelligent approaches. The most typical and significant instance of artificial EI is the family of approaches known as particle swarm optimization [2]. Distinguished from classical AI approaches that require the task-specific global knowledge to be explicitly integrated into a problem solver, EI approaches exploit the numerous agents involved in the task to opportunistically operate upon their representation-specific local knowledge, whereas the task-specific knowledge can be separated from the distributed problem solver, i.e., the agents. A comparison is briefly illustrated in Figure 7.2. In the framework of classical centralized ML, data are aggregated from users to a central node, where a task-specific global model is trained and shared by all users.

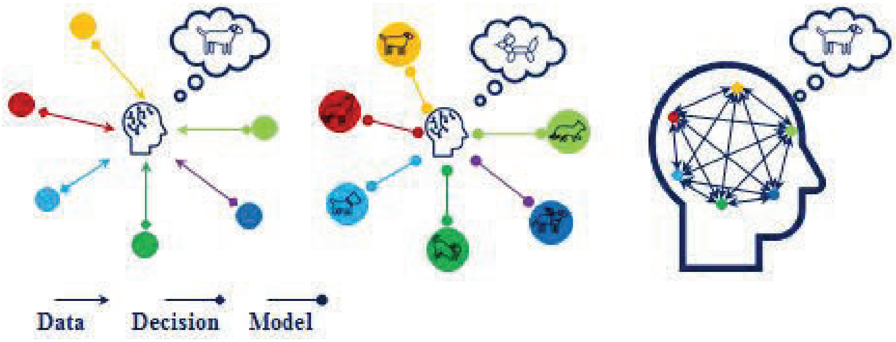


Figure 7.2 Comparing the conventional AI solutions based on centralized AI (left) and FL (middle) to EI (right).

Decisions are also usually made at the central node and sent back to the users, respectively. In the FL framework, instead of having one global model that applies for all users, the central node keeps only a so-called central model.

This model is shared with the users, so that every user locally trains it with own data and make decisions regarding the local model. The local model parameters of different users (instead of the raw user data) are aggregated and exploited at the central node to update the central model, which is then distributed to the users again to assist improving their local models. The framework of EI, in contrast, does not contain a central node, nor does it set up any explicit task-specific model. Instead, it relies on the decentralized information exchange among the users, which are architecturally equal and have no knowledge of the global task. From the reaction of each user to the information it collects from the others, some advanced behaviour pattern of the "colony" of all users can spontaneously emerge.

On the one hand, this model-less and emergent nature grants EI approaches several outstanding features that can benefit 6G IIoT, including low computational complexity, minimized computation and communication latencies, high robustness against local malfunction at arbitrary agent, data privacy, security, and scalability. On the other hand, 6G will also be able to enhance the performance of EI: it promises to deliver a ubiquitous, massive, and reliable connectivity in the IIoT environment, which will support to build a gigantic system with numerous agents networked with each other. Enhancements will be therewith introduced regarding the dimension and complexity of the networked system, as well as the efficiency of interaction between different system components. All these aspects have been proven to have critical impacts on the performance of EI solutions. In short words, 6G and EI are match made in heaven.

Nevertheless, it shall be remarked that the requirements of system scale and communication efficiency can be usually opposite each other. For example, when the number of agents increases within a limited coverage, the therewith increased access density may cause traffic congestion, resulting in either a higher latency or a lower link reliability. In another case where the access density remains consistent but the spatial dimension of the network increases, the coverage of a single radio access point becomes an issue. Message relaying will allow agents to interact over a long distance, but significantly increases the latency. Alternatively, it can be an effective low-latency solution to limit the communication range of agents but leads to a degradation in convergence performance. Furthermore, in addition to the user plane data exploited by the agents to make decisions, a significant signalling overhead must be generated to setup and accomplish the communication sessions between agents, which significantly reduces the energy efficiency and sustainability of the IIoT system.

To address these issues and support the deployment of EI in 6G IIoT, DTs will play an important role. In a massive twinning scenario, every EI agent can have not only its real-time status, context information and semantic model stored, analysed, and maintained at its DT, but also its decision engine migrated thereto as well. Thus, the information exchange between different agents can be shifted from the physical radio environment to the cyber world, and the radio link between every pair of agents can be replaced by an agent-cloud link for each individual agent, which will not only dramatically reduce the traffic load, but also mitigate the massive radio signalling overhead. Therewith, DTs will improve the radio resource efficiency and reduce the communication latency for EI applications.

7.4 Network-aware Digital Twins for Local Insight Generation

Industrial DTs of machines, processes, or whole factories might contain sensitive and business-critical information that needs to be retained within a local management domain (e.g., a private network or locally managed IT/OT systems). Traditionally, industrial DTs did not focus on the network, but on the industrial application and machinery. With an increasing share of wireless communication enabling novel Industry 4.0 scenarios and the vision of 6G as a network of networks, supporting local, independently managed network *islands* or *sub-networks*, this is changing significantly. One way to allow industrial DTs to benefit from network-awareness and utilize additional sources of data offered by novel capabilities of a 6G system (e.g., localization, sensing, computation, or AI as a Service) is the collaboration of DTs as illustrated in Figure 7.3.

The local DT on the left-hand side of the figure captures relevant aspects of the Industry 4.0 application or process being executed by several collaborating machines and humans. Local network infrastructure (wireless and wired) enabling this collaboration is also represented in the local DT to aid in network management and optimization tasks. This local loop of configuration and optimization based on the local DT is augmented with information from the 6G DT and its capabilities. Relevant aspects include the joint optimization of compute resources by utilizing the respective 6G services, or the joint optimization of network resources across management domains. Both

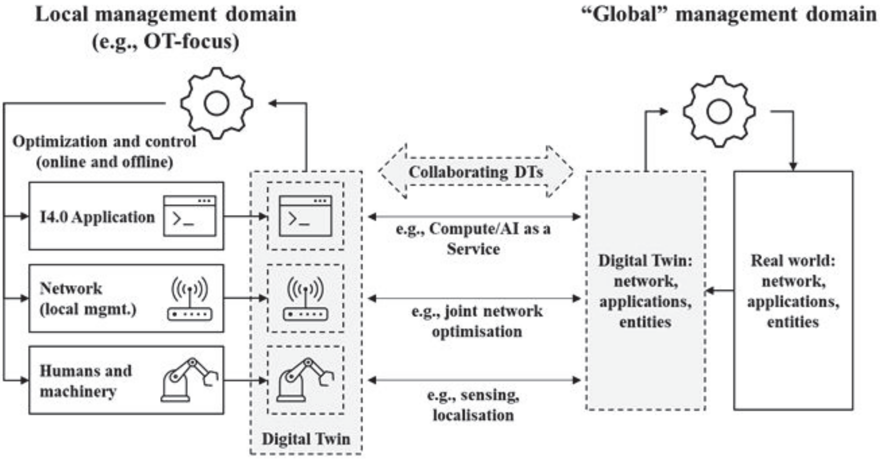


Figure 7.3 Illustration of collaborating DTs in IIoT.

domains could further benefit from a privacy-preserving exchange of sensing information to increase, e.g., location accuracy or confidence in measurement data for specific use cases. One example of such an exchange with mutual benefit is the joint optimization of trajectories of automated guided vehicles to increase process productivity while at the same time making better use of available communication resources. Instead of sense-and-react, both sides can benefit from the proactive exchange of information as foundation for AI-based decision making in the respective processes.

Being able to limit the exchange of data to trustworthy entities and act in a privacy-preserving fashion by sharing only the most relevant information among DTs allows cross-layer optimization for both, local and global management domains while still maintaining full control over own processes and data.

7.5 AI at the Intersection between DTs and HMI in Industrial IoT

The idea behind DTs is to create intangible replicas of physical assets or processes capable to capture key information that can be used to support design and planning activities, as well as to help operation and supervision tasks [4]. Initially developed in the context of, e.g., industrial plants and city infrastructure, today are progressively widening to encompass any real entity, including human beings [5].

There are several technological enablers that are easy to recognize as key to the implementation of DT solutions. One of them is indeed represented by mobile communications. In particular, there is a great expectation for the deployment of 6G networks, as their latencies and data rates are regarded as capable to make applications such as, e.g., autonomous driving and remote surgery, finally feasible [6].

The previous sections discussed the major role played in this context by AI. In fact, with AI, insights gathered through DTs can allow humans to make better operational decisions. AI can also make DTs more intelligent, to the point that they can even get able to make decisions and prescribe actions to the physical world on their own.

Using AI techniques, the DT of a city could, for instance, leverage information about road works and closures, pollution levels, or even citizens' habits to manage in real-time connected vehicles traffic [7]. Similarly, the DT of an industrial plant could constantly monitor machines' status and use collected data to instantaneously reconfigure processes to mitigate, e.g., downtimes and bottlenecks [8]. In logistics, the AI abilities could allow DTs to make fact-driven decisions regarding planning and scheduling based, e.g., on demand and distribution models, and support the implementation of optimization and control strategies aimed to improve efficiency and, ultimately, profitability [9].

Indeed, the traditional application domain for the DT paradigm is the industrial one, within the context of IIoT. A recent review of the role of AI in this context is reported in [10]. Within the commonly pictured scenarios for digital twinning, an area in which AI is expected to foster important developments is that of HMIs. Thanks also to forecasted advancement in mobile networks and edge computing capabilities, ever new ways in which the human and machine intelligences cooperate in CPSs can be envisaged. A typical use case is that of robotics, in which computer vision technology is essential for the navigation of mobile robots [11] or the interaction with collaborative robots (or cobots) [12]. Another typical application of AI techniques is that of human-action recognition from images and data collected by other sensors (like depth cameras) to perform, e.g., trajectory forecasting and path planning for safety assurance in scenarios involving the operation of co-located human and robotic agents [13][14].

A final family of technological tools that shall be mentioned in relation to AI-powered DTs and HMI is that of XR, a term generally used to refer to a blend of tools like Virtual Reality (VR), Augmented Reality (AR) and Mixed Reality (MR). XR plays a primary role in the scenarios depicted above

[15]. As a matter of example, VR-based simulations are commonly used for visualization purposes in pre-production processes or in the planning of surgery interventions, whereas AR is typically exploited in customer service applications or in Head-up Displays typically mounted aboard autonomous vehicles.

It is worth observing that DTs coupled with AI and XR are expected to represent extremely powerful tools also towards sustainability. In fact, the possibility to rely on virtual, distant copies of real-world entities means avoiding unnecessary travels to the physical location of such entities. This can be the case, e.g., of remote healthcare or maintenance applications [16]. It also means less energy consumption and waste since, as said, machine failures can be predicted in advance, and designs validated and tested before being realized [17].

7.6 Conclusion

In this chapter, we discussed the impact of AI on IIoT from the perspective of the 6G European research project Hexa-X. We outlined an ecosystem of collaborating DTs as a potential enabler for emergent intelligence and local insight generation in a privacy-preserving and trustworthy way. We further elaborated on the role of AI when it comes to the intersection between the DT and the way humans interact with it by means of novel HMIs in an industrial context. In Hexa-X, we study additional enablers for trustworthy, collaborative DTs and the utilization of gathered data for flexible resource allocation and dependable operation of applications and services as important cornerstones for most IIoT use cases.

Acknowledgements

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

References

[1] M. A. Uusitalo, et al. "6G Vision, Value, Use Cases and Technologies from European 6G Flagship Project Hexa-X." IEEE Access 9 (2021): 160004-160020.

[2] James Kennedy, “Swarm intelligence,” Handbook of nature-inspired and innovative computing, Springer, Boston, MA, 2006. 187-219.

[3] W. D. Hillis. "Intelligence as an emergent behavior; or, the songs of eden." Daedalus (1988): 175-189.

[4] A. Fuller, Z. Fan, C. Day, C. Barlow, “Digital twin: Enabling technologies, challenges and open research,” IEEE Access, vol. 8, pp. 108952-108971, 2020.

[5] B. R. Barricelli, E. Casiraghi, J. Gliozzo, A. Petrini and S. Valtolina, “Human Digital Twin for Fitness Management,” IEEE Access, vol. 8, pp. 26637-26664, 2020.

[6] L. U. Khan, W. Saad, D. Niyato, Z. Han, C. S. Hong, “Digital-twin-enabled 6G: Vision, architectural trends, and future directions,” IEEE Communications Magazine, vol. 60, no. 1, pp. 74-80, 2022.

[7] G. Mylonas, A. Kalogeras, G. Kalogeras, C. Anagnostopoulos, C. Alexakos, L. Muñoz, “Digital twins from smart manufacturing to smart cities: A survey,” IEEE Access, vol. 9, pp. 143222-143249, 2021.

[8] N. Kousi, C. Gkournelos, S. Aivaliotis, K. Lotsaris , A. C. Bavelos, P. Baris, G. Michalos and S. Makris, “Digital twin for designing and reconfiguring human–robot collaborative assembly lines,” Applied Sciences, vol. 11, 4620, 2021.

[9] A. Belfadel, S. Hörl, R. J. Tapia, J. Puchinger, “Towards a digital twin framework for adaptive last mile city logistics,” Proc. 6th International Conference on Smart and Sustainable Technologies, 2021.

[10] Z. Huang, Y. Shen, J. Li, M. Fey, C. Brecher, “A survey on AI-driven digital twins in Industry 4.0: Smart manufacturing and advanced robotics,” Sensors, vol. 21, no. 19, 6340, 2021.

[11] M. Minos-Stensrud, O. H. Haakstad; O. Sakseid, B. Westby, A. Alcocer, “Towards automated 3D reconstruction in SME factories and digital twin model generation,” Proc. 18th International Conference on Control, Automation and Systems, 2018.

[12] A. A. Malik, A. Bilberg, “Digital twins of human robot collaboration in a production setting,” Procedia Manufacturing, vol. 17, pp. 278-285, 2018.

[13] T. Wang, J. Li, Y. Deng, C. Wang, H. Snoussi, F. Tao, “Digital twin for human-machine interaction with convolutional neural network,” International Journal of Computer Integrated Manufacturing, vol. 34, no. 7-8, pp. 888-897, 2021.

[14] J. A. Douthwaite, B. Lesage, M. Gleirscher, R. Calinescu, J. M. Aitken, R. Alexander, J. Law, “A modular digital twinning framework for safety assurance of collaborative robotics,” Frontiers in Robotics and AI, vol. 8, 2021.

[15] S. Rabah, A. Assil, E. Khouri, F. Maier, F. Ababsa, V. Bourny, P. Maier, F. Mérienne, “Towards improving the future of manufacturing through digital twin and augmented reality technologies,” Procedia Manufacturing, vol. 17, pp. 460-467, 2018.

[16] H. Laaki, Y. Miche, K. Tammi, “Prototyping a digital twin for real time remote control over mobile networks: Application of remote surgery,” IEEE Access, vol. 7, pp. 20325-20336, 2019.

[17] V. Havard, B. Jeanne, M. Lacomblez, D. Baudry, “Digital twin and virtual reality: A co-simulation environment for design and assessment of industrial workstations,” Production & Manufacturing Research, vol. 7, no. 1, pp. 472-489, 2019.

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[3], Hans Schotten[5], Davide Calandra[6], and Fabrizio Lamberti[6]

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

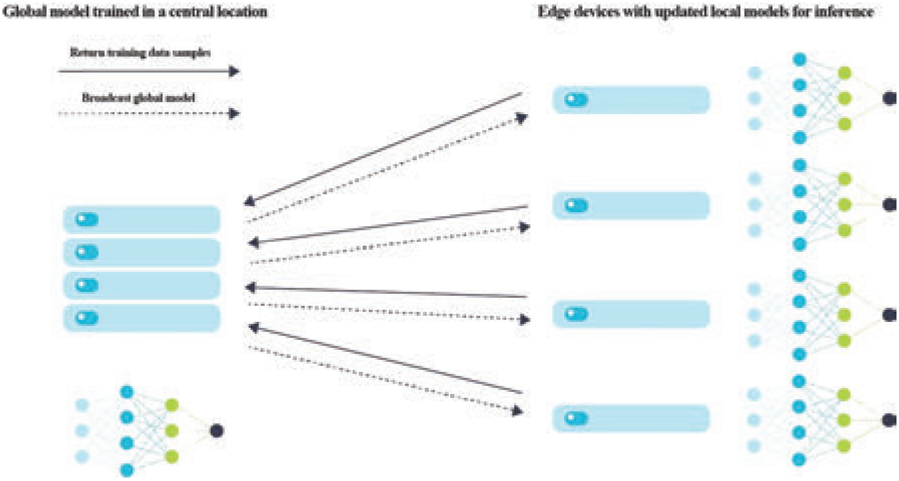

Global model trained in a central location
Edge devices with updated local models for inference
Return training data samples
Broadcast global model

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

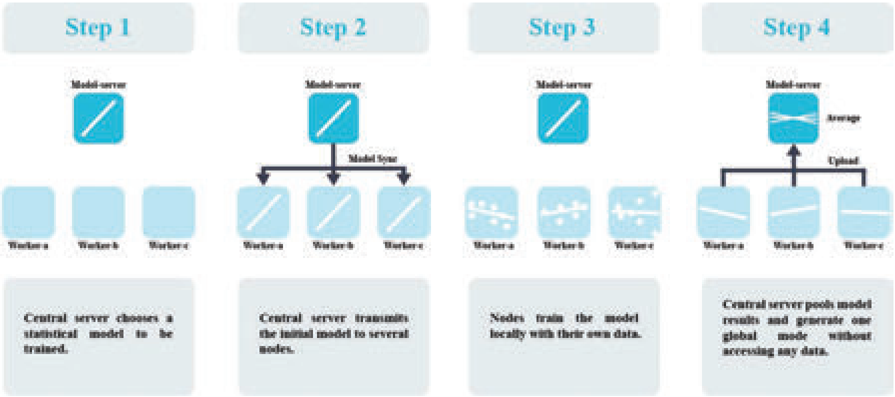

Step 1
Model-server
Worker-a
Worker-b
Worker-c
Central server chooses a statistical model to be trained.
Step 2
Model-server
Model Sync
Worker-a
Worker-b
Worker-c
Central server transmits the initial model to several nodes.
Step 3
Model-server
Worker-a
Worker-b
Worker-c
Nodes train the model locally with their own data.
Step 4
Model-server
Average
Upload
Worker-a
Worker-b
Worker-c
Central server pools model results and generate one global mode without accessing any data.

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.

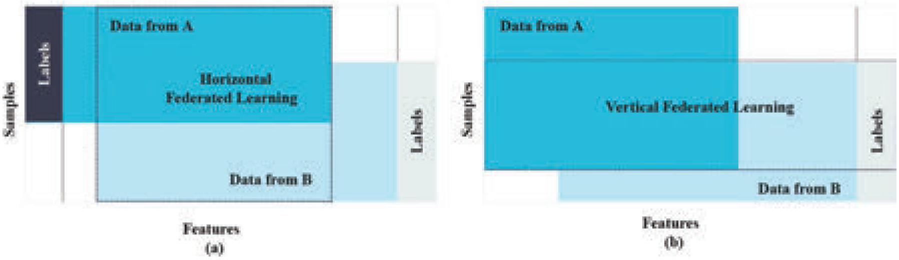


Figure 8.3 FL scenarios according to how the data is split across clients. (a) Horizontal FL. (b) Vertical FL.

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][27][28][29][30][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.

8.5 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][35][36][37]. In the following, we will focus on forward error correction (FEC) decoding, since this is the most computationally intensive part on the receiver side, which also introduces additional latency since we usually use iterative decoding schemes. In addition, short packets, which are common in machine communication system, reduce 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 no longer feasible.

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

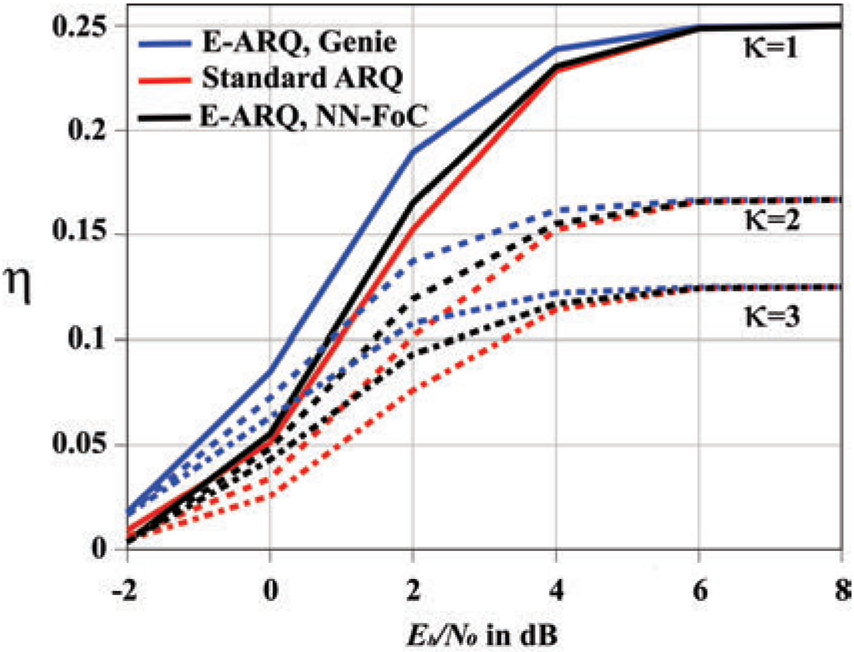


Figure 8.4 Efficiency η over SNR for standard ARQ scheme in comparison to E-ARQ with NN-FoC forecasting and a Genie forecaster for different decoder delays κ

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].

8.6 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

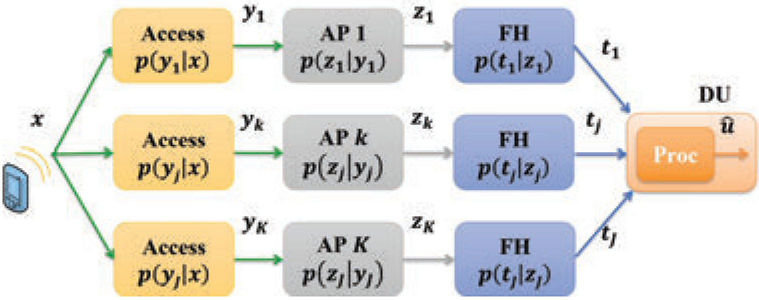


Figure 8.5 Distributed communication system with **J** access points forwarding compressed messages to the DU.

discrete alphabets $\mathcal{Z}_j$ with $|\mathcal{Z}_j| \ll |\mathcal{Y}_j|$ by the local quantizer function $z_j = Q_j(y_j)$. 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 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 $p(x)$

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

$\mathcal{Q}$ 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

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

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

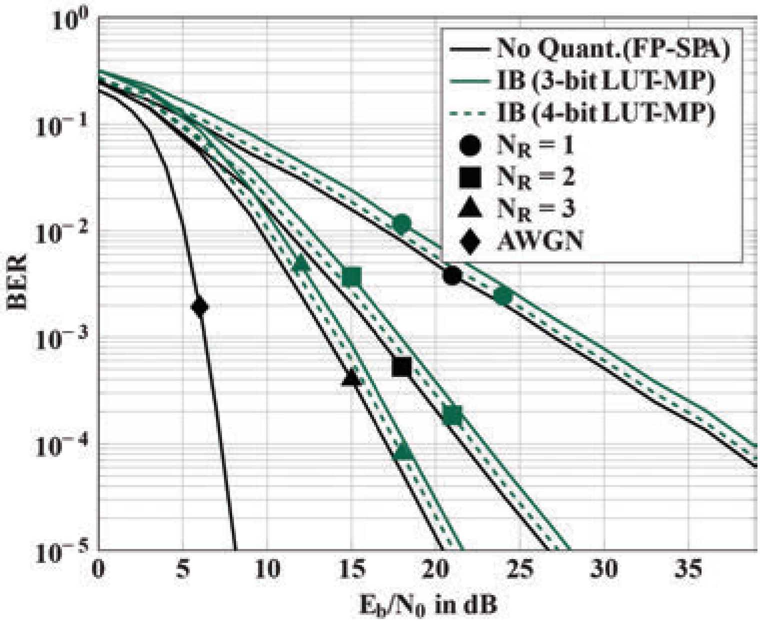


Figure 8.6 BER performance for 16-QAM with RAPs applying SNR-adapted 6-bit quantizer per AP and REMC in DU for $\mathbf{J} \geq \mathbf{1}$.

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.

8.7 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.

8.8 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

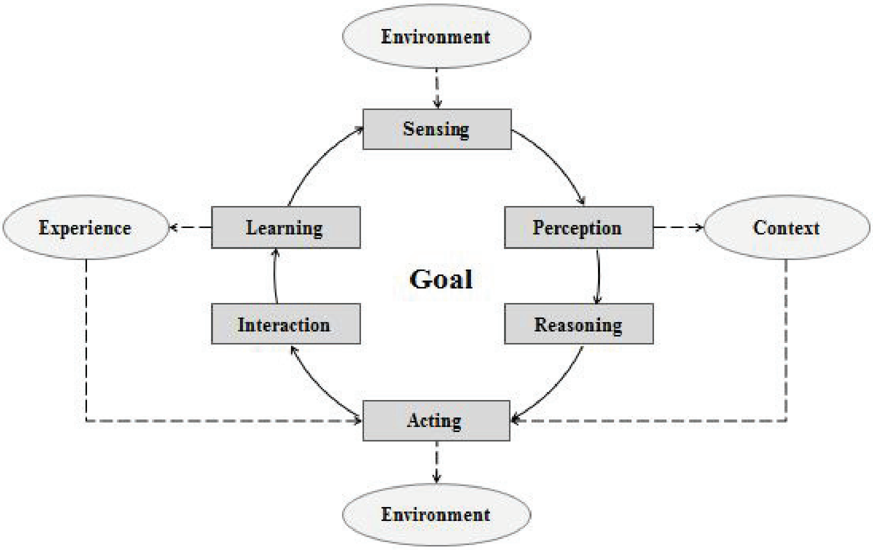


Figure 8.7 The cognitive cycle of an embodied intelligence agent.

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, 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][75][76][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.

8.9 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].

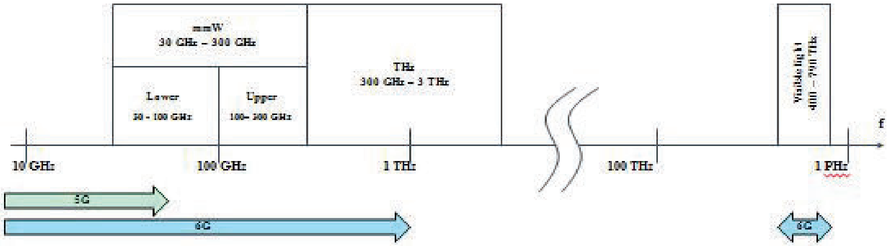


Figure 8.8 Overview of mmW frequencies. 5G bands expand up to 50 GHz, 6G is expected to reach 1 THz and also include visible light communications.

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 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][81][82][83]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

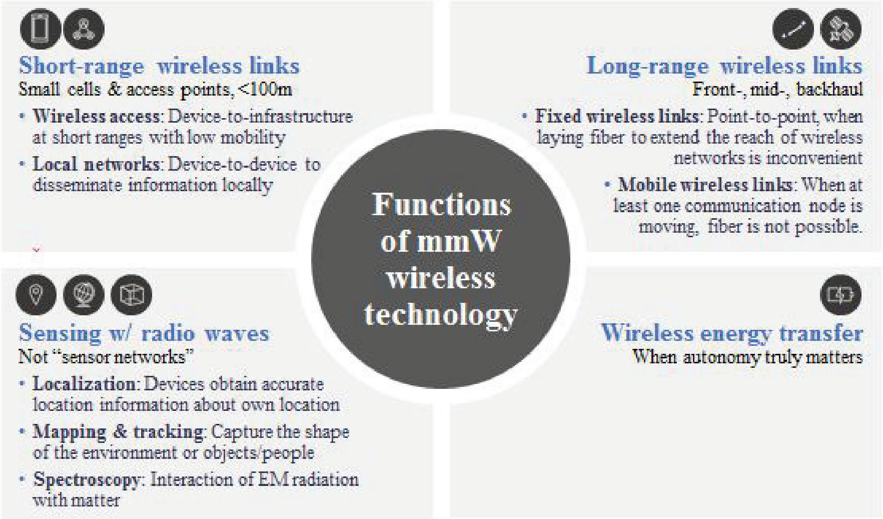

Short-range wireless links
Small cells & access points, <100m
• Wireless access: Device-to-infrastructure at short ranges with low mobility
• Local networks: Device-to-device to disseminate information locally
Long-range wireless links
Front-, mid-, backhaul
• Fixed wireless links: Point-to-point, when laying fiber to extend the reach of wireless networks is inconvenient
• Mobile wireless links: When at least one communication node is moving, fiber is not possible.
Functions of mmW wireless technology
Sensing w/ radio waves
Not "sensor networks"
• Localization: Devices obtain accurate location information about own location
• Mapping & tracking: Capture the shape of the environment or objects/people
• Spectroscopy: Interaction of EM radiation with matter
Wireless energy transfer
When autonomy truly matters

Figure 8.9 Overview of the functions of mmW wireless technology.

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 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.

8.10 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 29th 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. Koneènı, 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”. 27th 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, S. Dörner, S. Cammerer, and S. ten Brink, "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, D. Wübben, and A. Dekorsy, "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., "Finite-Alphabet Message Passing using only Integer Operations for Highly Parallel LDPC Decoders", 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.2014.70634 90.

[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.

9

Ethical Considerations and Trustworthy Industrial AI Systems

Ovidiu Vermesan[1], Cristina De Luca[2], Reiner John[3], Marcello Coppola[4], Björn Debaillie[5], and Giulio Urlini[6]

[1] SINTEF AS, Norway
[2] Silicon Austria Labs GmbH, Austria
[3] AVL List GmbH, Austria
[4] STMicroelectronics, France
[5] imec, Belgium
[6] STMicroelectronics, Italy

Abstract

The AI ethics in industrial environments is a new field within applied ethics, with notable dynamics but no well-established rules and no standard overviews. It poses many more challenges than similar consumer and general business applications, and the digital transformation of industrial sectors has brought into the ethical picture even more considerations to address. This relates to integrating AI and autonomous learning machines based on neural networks, genetic algorithms, and agent architectures into manufacturing processes.

This article presents the ethical challenges in industrial environments and the implications of developing, implementing, and deploying AI technologies and applications in industrial sectors in terms of complexity, energy demands, and environmental and climate changes.

It also gives an overview of the ethical considerations concerning digitising industry and ways of addressing them, such as potential impacts of AI on economic growth and productivity, workforce, digital divide, alignment with trustworthiness, transparency, and fairness.

Additionally, potential issues concerning the concentration of AI technology within only a few companies, human-machine relationships, and behavioural and operational misconduct involving AI are examined.

Manufacturers, designers, owners, and operators of AI—as part of autonomous industrial systems—can be held responsible if harm is caused. Therefore, the need for accountability is also addressed, particularly related to industrial applications with non-functional requirements such as safety, security, reliability, and maintainability supporting the means of AI-based technologies and applications to be auditable via an assessment either internally or by a third party. This requires new standards and certification schemes that allow AI systems to be assessed objectively for compliance and results to be repeatable and reproducible.

This article is based on work, findings, and many discussions within the context of the ECSEL JU AI4DI, ArchitectECA2030 and AI4CSM projects.

Keywords: Artificial intelligence, ethics, digitising industry, industry- grade AI, industrial internet of things, machine ethics, explainable AI, trustworthiness, responsible AI, technology ethics.

9.1 Introduction

We all remember the frequent quotation of Isaac Asimov and his famous Three Laws of Robotics (1942). First Law: A robot may not injure a human being or, through inaction, allow a human being to be harmed. Second Law: A robot must obey orders given by humans, except where such orders conflicts with the First Law. Third Law: A robot must protest its own existence, provided such protection does not conflicts with First and Second Law. These three laws perfectly reflect the need for future use of AI not to harm human beings. On the other side, the definition of AI proposed in the European Commission's Communication on AI [2][3][4][5] states that "Artificial intelligence (AI) refers to systems that display intelligent behaviour by analysing their environment and taking actions – with some degree of autonomy – to achieve specific goals. AI- based systems can be purely software-based, acting in the virtual world (e.g., voice assistants, image analysis software, search engines, speech and face recognition systems), or AI can be embedded in hardware devices (e.g., advanced robots, autonomous vehicles, drones or Internet of Things (IoT) applications)."

In the context of the ECSEL JU AI4DI project under European Union's Horizon 2020 research and innovation programme [1], AI is defined as a machine's ability to perform logical analysis, acquire knowledge, and adapt to an industrial environment that varies over time or in context. These abilities include the collective attributes of a machine (i.e., computer, robot, or intelligent IoT device) to perform functions such as perception, understanding, reason, prediction, learning, decision making and action.

Another definition [6] mentions that AI is an activity dedicated to creating machine intelligence. Intelligence is a quality that allows an entity to function appropriately and with insight and foresight in its environment.

The increased number of intelligent machines, products and services, (i.e., equipment, industrial IoT devices with embedded AI, etc.), based on machine learning (ML), artificial neural networks (ANNs) and deep learning (DP), deployed in industrial environments, require to open the discussion on ethical principles and how these relate to AI.

AI is defined based on outcomes and actions [23]. The ethics of AI in industrial environments are evolving due to discussions around industrial AI trust, technical problems that focus on achieving the desired outcome for AI-based technologies and applications in manufacturing sectors. This is a new field within applied ethics and comes with notable dynamics, controversial issues, a lack of standards and no common agreement on principles about ethics.

Trust in an industrial AI system has multiple dimensions combining system dependability characteristics (e.g., privacy, security, safety, reliability, availability, resilience, connectability and maintainability) with human and machine behaviour. There is a need for a greater understanding of how individuals interact with machines and how machines/things interact with other machines/things to extend the concept of trust.

Trust in industrial AI systems is a characteristic of human-to-machine and machine-to-machine relationships formed with different industrial AI-based systems. In industrial processes, a further understanding of how individuals interact with AI-based machines and how these machines/things interact with other machines/things is critical for building the industrial AI trust concept. In many industrial processes, AI trust is developed by considering the performance (e.g., accuracy, robustness, stability, speed, data quality, etc.) of AI, the ML model, the operations (compliance, dependability, response to uncertainty, monitoring, governance, etc.) of the industrial AI system and the set of rules, guidelines, and standards (e.g., ethical, technical, etc.) in the industrial workflow. The rules/guidelines related to, for example, transparency, explainability, bias, and fairness apply to both the design of the industrial AI system, how it is used and how its functions are explained in the industrial process.

Uncertainty and vulnerability are two of the core elements of AI trust. In addressing industrial AI trust issues, industrial stakeholders must select strategies that reduce uncertainty or decrease vulnerability, depending on the context of the problems. Design for industrial AI trust requires evaluating the operating assumptions and examining how those assumptions can function to put some users of the AI system at risk. Understanding and designing AI trust systems require an understanding of the rules of the AI system and the functions of autonomous/cognitive elements.

From an industrial AI technology perspective, trust refers to trust measurement capabilities. This requires the use of trust assessment approaches, such as recommendation and reputation systems, which calculate the trustworthiness of one industrial AI system to match it against the need for trust of another industrial AI system.

As industrial AI technologies are maturing and AI-based applications are proliferating in different industrial sectors, new standards are demanded that describe measurable and testable levels of transparency are required; in this way the AI-based systems are objectively assessed for compliance to be reliable, safe, trustworthy, and operate with integrity.

The developed economies understand the game-changing nature of AI and have embraced different approaches to accelerate and control the development of AI technologies and applications.

Industrial AI depends on addressing the trade-off between incorporating the benefits and mitigating the potential disadvantages of AI by simultaneously avoiding the misuse and underuse of AI technologies in industrial environments.

Embracing an ethical approach to industrial AI provides what is considered a twin benefit of using ethics to allow industrial organisations to take advantage of AI's value and anticipate, avoid, and minimise expensive missteps and errors.

A framework of industrial AI principles is based on statements of the values or principles that guide the development and deployment of AI in society and that have already been proposed by different multi-stakeholder organisations and initiatives [22][23][24][25][26][27][28][29]. Asilomar principles [26] provide the greatest number of such principles organised under three issues: research, ethics and values, and longer-term issues. Regarding these principles five topics emerge as key for AI ethics:

- Autonomy as the element to use for whether or not to delegate.
- Beneficence as related to doing only good and providing a benefit.
- Non-maleficence as related to causing no harm and damages.

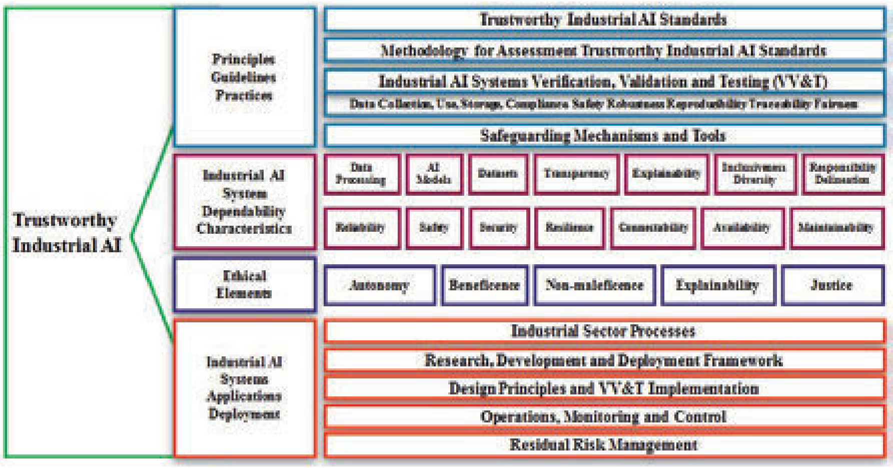

Trustworthy Industrial AI
Principles Guidelines Practices
Trustworthy Industrial AI Standards
Methodology for Assessment Trustworthy Industrial AI Standards
Industrial AI Systems Verification, Validation and Testing (VV&T)
Data Collection, Use, Storage, Compliance Safety Robustness Reproducibility Traceability Fairness
Safeguarding Mechanisms and Tools
Industrial AI System Dependability Characteristics
Data Processing
AI Models
Datasets
Transparency
Explainability
Inclusiveness Diversity
Responsibility Delineation
Reliability
Safety
Security
Resilience
Connectability
Availability
Maintainability
Ethical Elements
Autonomy
Beneficence
Non-maleficence
Explainability
Justice
Industrial AI Systems Applications Deployment
Industrial Sector Processes
Research, Development and Deployment Framework
Design Principles and VV&T Implementation
Operations, Monitoring and Control
Residual Risk Management

Figure 9.1 A framework for trustworthy industrial AI systems.

- Explainability as related to how the AI-based system does its work and who is responsible for the way it works.
- Justice as related to promoting fairness, and prosperity, preserving solidarity and avoiding unfairness, bias, and discrimination.

A framework for trustworthy industrial AI systems including the elements and principles presented above is illustrated in Figure 9.1.

One key opportunity for the industrial sector in Europe to be competitive is to ensure the take-up of AI technology across its industry. The development of higher efficient electronic components and systems, circuits specifically built to run AI operations (neuromorphic circuits), high-performance computers, quantum technologies and technologies for mapping the human brain accelerate the possible applications of AI-based technologies in industrial sectors and urgently require addressing the issues of ethical challenges that the AI brings.

9.2 Ethics and Responsible AI in Industrial Environments

Nowadays, AI is impacting many aspects of industrial activities. There is a need to understand how AI should be designed to i) operate responsibly, ii) meet stakeholders' expectations and iii) applicable regulations and concerns relating to reliability, privacy data leakages, information transparency, explainability and ethical considerations [16].

Addressing these ethical dilemmas and concerns when developing industrial AI-based solutions strengthens a manufacturer's credibility for delivering products and services and enhances an organisation's reputation in the marketplace. Nevertheless, this is not an easy task, as industrial applications have much higher requirements (e.g., reliability, verifiability, safety, etc.) than AI-based products designed for the consumer market.

Many industrial companies address the ethical and environmental concerns around the responsible use of AI in corporate social responsibility strategy to make socially significant decisions and consider using "ethical algorithms" to reduce the risk of unethical behaviours.

There is no single definition for what responsible AI means, and organisations will usually develop their terminology and methodology. Nevertheless, designing AI to operate responsibly means at its core following design principles that allow AI systems to justify and be held responsible for their decisions. In industrial environments, this ultimately comes down to allowing human inspection of the functionality of AI algorithms and models. The development of AI systems is complex, involving many sub-systems with different ethical considerations, making it challenging to inspect and evaluate such systems. The complexity arises from the fact that ethically compliant sub-systems do not necessarily make the overall system ethically compliant. The subsystems interact with each other and exchange feedback, which may change conditions in the application's environment, conditions that cannot always be anticipated during development. This may be the case with AI systems that continue to learn after deployment. Therefore, re-evaluation of ethical compliance must be conducted regularly or with every change of the application context, especially in AI systems with widespread or profound ethical issues. Safety-critical systems, where industry regulation would make the re-evaluation mandatory, can be such a case. A schematic representation of the elements in the development process of AI systems is illustrated in Figure 9.2.

In this context, it is essential to note that designing and developing responsible AI is not a one-time process but rather entails continuous striving to maintain responsible AI systems and keep up with technological advances that may bring new ethical implications.

9.3 Requirements for Industry-Grade AI

Defining the requirements for industry-grade AI is crucial as advanced machines and Industrial Internet of Things (IIoT) devices with enhanced AI

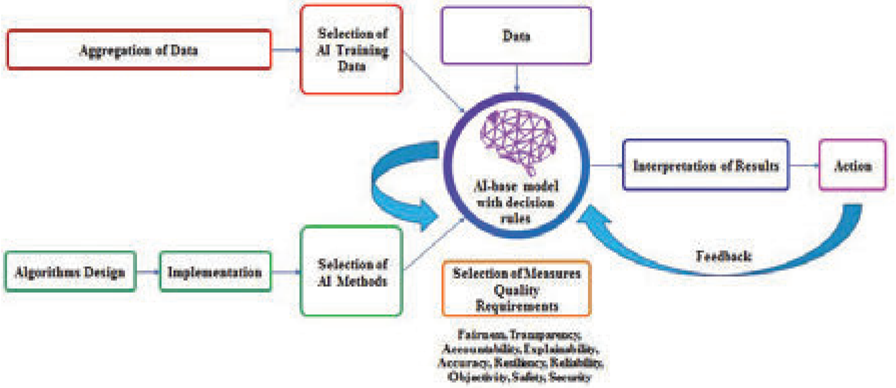

Aggregation of Data
Selection of AI Training Data
Data
AI-base model with decision rules
Interpretation of Results
Action
Feedback
Algorithms Design
Implementation
Selection of AI Methods
Selection of Measures Quality Requirements
Fairness, Transparency, Accountability, Explainability, Accuracy, Resiliency, Reliability, Objectivity, Safety, Security

Figure 9.2 Complexity of applicability of ethical considerations resulting from the interaction of subsystems.

capabilities may operate in ways that were not envisaged when the AI-based system was designed and put into operation.

The requirements for industry-grade AI technologies and applications identified by the AI4DI project [1] are illustrated in Figure 9.3. A short

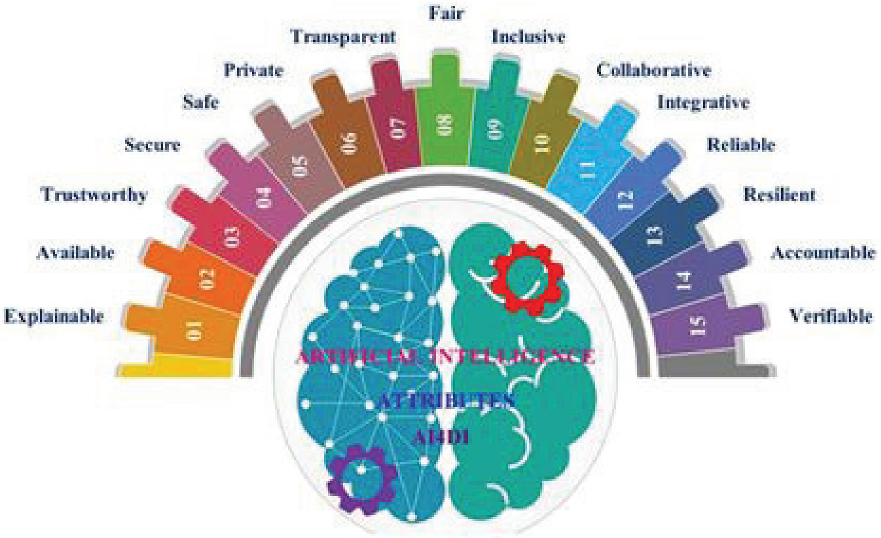


Figure 9.3 Requirements for industry-grade AI [1].

explanation of these requirements is presented in the next paragraphs. Some of these requirements are further addressed in specific sub-sections.

Explainable: Humans must comprehend the decision of AI systems to track down failure and assess decisions, and the AI systems must either provide enough information required to explain its actions and decisions or possibly even explain the output itself (explainable AI).

Available: Industrial applications for AI will target mission-critical tasks along the complete production line, and system outages will have a direct economic impact. Industry-grade AI systems, therefore, need to fulfil high availability standards. In a second step, they should also perform autonomously via online learning over their lifetime to avoid maintenance. Moreover, they should also be quickly available in terms of integration into new applications and process steps.

Trustworthy: When more and more AI-enabled devices become connected through the IIoT, trustworthiness will become an indispensable requirement for AI systems. It is essential that the identity of every AI system can be verified, and that vulnerabilities and inconsistencies are immediately reported.

Secure: Industry-grade AI must implement security measures to ensure robustness against all types of attack vectors through different devices, workers, operators, etc. This also includes securing the AI system by making it robust against adversarial attacks and manipulated input data. The communication between edge computing devices needs to be secured with encryption and authentication mechanisms. Security is very important, particularly when we execute AI on the edge. Therefore, protecting embedded ML models against attacks by safeguarding the integrity (fooling decision) and confidentiality of the data is vital when IIoT systems are deployed in the field. With AI, the overall attack surface is large since we are gathering algorithmic attacks (such as adversarial examples) and physical attacks (side-channel and fault injection analysis). To address these issues, it is important to check the robustness of models, add cryptographic-based authentication schemes and add secure boot-like technologies to enforce the trust of embedded AI systems against malicious tampering.

Safe: AI systems that operate physically next to and collaboratively with humans through robots or other machines must comply with current and future safety standards to prevent accidents. Notably, the employed AI systems must be robust against implausible data and operate with extremely low latency to quickly react to unforeseen events. Likewise, AI for the control of safety-critical processes must also comply with the latest safety standards.

Private: Industry-grade AI will operate on mission-critical personal data from the manufacturer and customers and business-critical information (data security, confidentiality, etc). This data must be kept confidential and protected from external access. This precludes external cloud storage and the application of typical big data methods. Instead, information must be processed locally at the edge and only leverage data available within privacy limits (smart data).

Transparent: The state, actions, and decisions of an AI system must be inspectable and understandable at any point in time. This will be supported by digital twins that represent the complete system state at any point in time. AI methods for data visualisation can further enhance transparency and make the systems state easier to understand.

Fair: AI technologies that support or automate decision processes must adhere to the same fairness and compliance standards as defined by the industrial sector regulation.

Inclusive: AI systems need to include humans and existing systems in their operations to avoid the formation of isolated non-AI capable sub-systems within a process, production system or supply chain.

Collaborative: Industry-grade AI will not be concentrated on a single device or system. Instead, many different AI-enabled sub-systems will be distributed (distributed AI) across IoT nodes, embedded devices, and other edge devices (AI-Born embedded systems). These devices need to self- organise and collaborate to ensure coherent operation at the level of the whole system. They also need to collaborate with humans physically (e.g., human-robot collaboration) and by exchanging information (human-machine interfaces).

Integrative: Industry-grade AI systems must be open and flexible to ensure that they can be integrated seamlessly into existing systems and processes. This is a key prerequisite of establishing AI methods in the industry according to a sustainable roadmap.

Reliable: Reliability and dependability (dependable AI) are key prerequisites for AI systems that are put into continuous operation with short maintenance time in mission-critical production environments. AI must not harm productivity by an unreliable operation that requires regular human intervention or even causes system outages.

Resilient: Industry-grade AI must remain stable even when other parts of the process fail. In the future, they should even be able to detect the failure and initiate measures for compensating it.

Accountable: Industry-grade AI that supports or even replaces human decisions must be implemented to ensure that it can be made accountable for its output (e.g., via the supplier of the AI system).

Verifiable: AI systems for industrial applications must fulfil the same standards as legacy systems and will be applied to safety-, mission-, and business-critical tasks. This requires that Industry-grade AI systems can be validated (to reach correct results), verified (verifiable AI) and certified (certifiable AI) for the targeted applications.

9.4 Industrial AI Challenges

The industrial AI technologies are powered by complex programming and algorithms run on high-performance energy-consuming computing units. Hence, the AI technologies are affecting how we interact with the environment and the resources used to power the different AI-based solutions.

While AI has many positive impacts, the widespread use of AI solutions in industrial environments can have indirect adverse and hidden effects that can harm the environment.

In many cases, phrases like “the data is the new oil” are used to highlight the digital transformation without considering that as for the oil, when data is used excessively, it is polluting the environment.

Raw data has no value in itself. Instead, the value is created when collected effectively and accurately, connected to other relevant data, done on time, processed, and refined. When well refined, usable data immediately becomes a decision-making tool – information – allowing companies to use it in the manufacturing decision-making and process automation.

Processing the information requires advanced AI-based hardware accelerated devices, software, algorithms, model, storage, computing, and connectivity capabilities that increase the complexity of the systems, i.e. use more natural resources, energy, and pollute the environment and generate new waste.

Large-scale deployment of AI could have both positive and negative impacts on the environment. Positive impacts can improve the user experience and the durability of machines. Thanks to preventive maintenance, better products can be made by adapting the production process to external situations. Negative impacts include increased complexity, use of natural resources, pollution, waste, and energy consumption.

In the following paragraphs, these challenges are highlighted to present a comprehensive overview of the trade-off that must be considered when developing AI-based IIoT and other types of systems in industrial environments.

9.4.1 Complexity

Ethical implications and challenges are present even in the simplest AI systems. In many cases, the more complex AI systems, the greater the challenges associated with their unpredictability and lack of transparency.

Industrial AI-based solutions result from multidisciplinary cooperation, and almost all AI-based systems are complex systems integrating IIoT devices, hardware, software, models, algorithms, and platforms.

The robustness and performance of models and algorithms are strongly dependent on their learning abilities; hence, improving learning ability performance will increase complexity. For instance, in the case of deep neural networks, widening or deepening the network will enhance the learning ability and the performance of the overall AI-based solution, but in many cases, it will also increase its complexity. The many internal hidden layers will be more challenging to penetrate for the purpose of analysis, including verification of ethical compliance.

In industrial environments, complex problems have multiple layers, each of which has multiscale parameters and characteristics, with the different layers correlated to each other. The AI-based industrial complex systems consist of numerous elements/components with a spatiotemporal multiscale structure between the system and elements/components scale due to the collective effect of these factors.

The complexity of the AI-based systems in industrial environments is in many cases determined by the difference between intelligent machines and human thinking. As demonstrated in many AI applications, statistical methods of ML, including the field of neural networks, vary in many forms from biological concepts of understanding, thinking, decision-making or learning.

9.4.2 Use of Natural Resources

Large-scale deployment of AI could have both positive and negative impacts on the environment. AI is creating positive environmental impacts in many applications but can negatively impact others where extensive use of natural resources has already damaged the environment. By increasing the demand for natural resources to increase automation and yield, AI can accelerate environmental degradation.

Data used in AI applications must be captured, stored, analysed, and transferred to different locations, which requires significant amounts of processing power. It is estimated that 175 zettabytes (ZB) worth of data will be stored globally by 2025 [18], which means extensive use of natural resources to address the need of energy, cooling, water, buildings etc.

AI will probably increase the demand for new materials needed for batteries that power the devices based on AI algorithms to perform intelligent functions on the manufacturing floor.

9.4.3 Pollution and Waste

AI is being used in applications to combat waste pollution. However, as more companies across more industries begin to use AI, there is growing concern that AI technologies will also extend the climate crisis.

AI and ML algorithms are training for longer and longer, using more and more sensors/devices generating data and consuming more and more energy, thus directly or indirectly increasing pollution by generating more and new types of waste that can be detrimental to the environment.

The use of power-intensive GPUs, energy-inefficient algorithms, a large amount of data to run ML training are all considered contributing to increased carbon dioxide emissions.

In this context, researchers are proposing ways to monitor the carbon footprint of AI algorithms and evaluate the pollution generated by AI applications. Code can be attached to the AI models and algorithms to track the energy use of individual AI-based processing units. An online calculator tool is used by [13] to give the raw carbon emissions produced and the approximate offset carbon emissions (depending on the grid used by the cloud provider).

9.4.4 Energy

AI requires extensive amounts of energy for manufacturing and training/learning. This will increase the carbon footprint of manufacturing products based on AI and the overall energy consumption across the entire lifetime of a product that needs continuously retraining and learning.

Training AI algorithms is an energy-intensive process, and estimates hint that the carbon footprint of training AI is as much as 284 tonnes of carbon dioxide equivalent, which represent five times the lifetime emissions of an average personal vehicle [12][14].

In most AI-based solutions, the energy-inefficiency of AI algorithms begins with the need to fine-tune the model for particular tasks, translate from one language to another and perform many iterations until the expected results and performance are obtained.

For AI-based technologies and applications, the solutions to energy consumption issues are using renewables to power the computing capabilities responsible for processing, storing, and training data, distributing the processing and analytics at the edge, designing more energy-efficient algorithms, software/hardware systems, and connectivity (e.g., cellular, wireless).

9.5 Ethical Considerations for Digitising Industry

Digital ethics offers a critical reflection about the changes in the industry and manufacturing processes shaped by digital and AI technologies. The digital divide extends from a technical phenomenon to broader ethical issues related to free competition, economic monopolies, silos that can affect the industrial environments.

Autonomous and intelligent AI-based systems have become more pervasive and designed to reduce human intervention in industrial processes and accelerate automation.

In this context, new ethical considerations must be addressed, and topics such as trustworthiness, fairness, transparency, accountability, explainability, and control must be discussed to develop guidelines, standards, and embedding norms in AI-based systems to support their governance in industrial environments. The following paragraphs offer a short overview of these topics and the challenges linked to AI-based systems.

9.5.1 AI Trustworthiness

Digital technologies in manufacturing are pervasive, and AI trustworthiness is imperative for the manufacturing processes to work correctly. To provide risk-free and reliable operation, intelligent machines and processes require continuous supervising ML algorithms used to make decisions. The control and supervision require essential time and resources, to the point that using digital technologies could become very expensive. On the other hand, not controlling the AI-based process may lead to severe risks for the safety and security of the entire production line. AI trustworthiness is based on technology robustness, bias, fairness, transparency and explainability.

The "EU Ethics guidelines for trustworthy AI" [4] provides high- level requirements/principles for trustworthy systems: Human agency and oversight (empowering human beings in order to make informed decisions; keeping the human in the loop), technical robustness and safety (resilient, reliable system functioning), privacy and data governance, transparency, diversity, non-discrimination and fairness, societal and environmental well-being and lastly, accountability (responsibility and accountability for AI systems).

According to NIST, an AI application's trustworthiness value is derived from several variables such as accuracy, explainability, resiliency, safety, reliability, objectivity, security, and accountability [18].

Several of these variables are addressed in the following sub-sections.

9.5.2 Bias and Fairness

Considering that more decisions are delegated to AI in industrial processes, it is crucial to ensure that the decisions and findings are free from bias and unfairness.

Biases prevent AI applications from making fair decisions in the same way as biases affect humans, and they can reside in both the AI training data and the algorithms, both of which are generated by humans.

Data sets can often contain hidden biases due to being incomplete and not covering the whole ground; in other cases, data sets can originate from sources outside the organisation, exhibiting slightly different ethical values.

Developers may also unintentionally programme biases into AI systems, although this is less often the case in industrial environments than in the consumer market.

In many cases, it is impossible to know in the design phase what algorithms based on neural networks are learning when they are trained with a specific data set. In industrial processes, the selection of the training sets, the test sets, and the verification and validation of the results to assess the efficiency and fairness of different algorithms are part of accepting or rejecting the use of the algorithms in the industrial process.

Fairness requires knowing why an AI-based automated process made a particular decision and the mechanisms that may change the decision and is thus connected to the AI models' interpretability and transparency of the training, design, development, and deployment processes with which the models were created.

The absence of fairness that results from the performance of an AI-based industrial system is in most cases due to algorithmic bias generated by a particular categorical distinction.

In this context, in industrial environments, it is critical to identify the root cause for introducing bias in AI systems, if any, and how it can be prevented throughout the lifecycle of the AI-based solution.

AI bias in industrial environments - whether in AI algorithms or training data - can promote distrust and generate distorted outcomes, which decreases the potential of AI for the industry. Introducing AI-based solutions in industrial sectors ensures that AI technologies strengthen human decision making. The industry's stakeholders aim to support scientific advancement and standards that can minimise AI bias.

9.5.3 Transparency

Implementing trustworthy AI-based solutions in industrial environments is closely related to some of the other elements presented in this section, such as fairness, accountability, and transparency.

Transparency relates to the capability of an AI system to, always, be able to provide a satisfactory explanation for its decisions, auditable either by an in-house or an independent human authority assessment. In the case of failure causing harm, it should be possible to ascertain why.

AI transparency must be addressed over the lifecycle development of an AI-based solution from the concept, design, deployment, operation, maintenance, upgrade/update and disposal. In approaching AI transparency in many cases, algorithmic transparency and algorithmic decision-making are the starting point.

In industrial environments, several AI components can be based on black-box solutions. To achieve AI transparency, the openness of the development process must be considered when designing AI-based solutions to allow for explainability concerning interpretability and trust in the AI-based systems.

9.5.4 Accountability

The assumption that human beings are the ultimate decision-makers is one of the fundamental premises most laws and regulations rely on when attributing responsibility. As AI-based autonomous devices become more advanced and ubiquitous, that will increasingly be less true when the "decision-maker" is a machine and not a person [8].

In industrial processes using AI technologies and applications, the responsibility for the AI's action/inaction/malfunction is attributed to an actor that is part of a business agreement, the owner, designer/developer, manufacturer, operator of an AI technology or application.

As the autonomous systems develop and become more intelligent new decisions can be made by the AI-based system, and the intelligent machines hold a certain level of responsibility for their actions. A responsibility gap is created when the behaviour of AI-based products deviates from the initial programming of the developer/designer to become a product of its interactions with its environment making the ascription of responsibility highly complex and unclear [7].

9.5.5 Explainability

In industrial environments, ethical concerns may arise when inaccurate and even incorrect predictions are reached related to either the product or the process. To address these concerns, industrial AI developers need to be able to explain how algorithms predict using various technical approaches and the factors that impact the decisions.

The AI-based technology used in industrial processes must explain WHAT it was designed to do, HOW it was designed to do such functions, and WHY it was designed in that distinct way instead of some other way.

Ensuring that AI-based hardware, software, and algorithms do what are intended to do and that there are no biases or unintended consequences must be addressed through validation and evaluation of the AI-based solutions during development by measuring the performance of an AI-based system through implementation to detect bugs, biases, and incorrect assumptions.

AI-based industrial systems can miss essential facts about the environment, and it is crucial to verify that these systems are operating as intended., including whether the AI models accurately estimate what they are supposed to.

AI explainability should be formulated for different systems, such as sensing, perception, and decision-making. Assessment of industrial AI explainability and explanations needs to be aligned with the industrial context, benchmarking, and targeted use cases, applications, or stakeholders (e.g., developers, users, consumers, etc.). High-level requirements for AI explainability need to be defined by industrial regulations or international standards. They should be aligned with the definition of transparency and verifiability for AI applications in various industrial contexts and at different cognition levels.

9.5.6 Control

Control is another matter that impacts trust, explicitly concerning how much control to exercise over AI, ranging from complete human control to complete AI autonomy. Balancing these two extremes is always possible, so the question is rather what form of control can be exercised and how it can be exercised without hampering the benefits of AI.

One approach is to build self-assessment capability into the AI system before deployment to enable the system to take corrective actions during operation, if necessary, even shutting itself down if harm is anticipated.

The idea to control AI-based technologies is to make ongoing self-evaluations and to test an integral part of a system's operation to diagnose how the AI-based system performs and correct any errors.

Ethical data sets could be used to continuously monitor and check for deviant behaviour, implementing an effective and observant response to ethical behaviour deviations of the algorithms.

Another approach is to keep humans in the loop able to intervene and override decisions that may cause harm.

9.5.7 Human-Machine Interaction and Manipulation of Behaviour

When developing human-machine relationships on the manufacturing floor, it is challenging to prognosticate the psychological effects of forming relationships with different intelligent machines.

Straightforward collaboration between humans and machines in industrial environments requires the interactions to be intuitive, seamless, and unobtrusive. This must be reflected in the implementation of AI-based interfaces built to control and manage these interactions.

Relationships with machines may affect human users' mental and social development and create barriers for humans in understanding the relationships between machines.

The cooperation of mixed groups of machines and humans in automated production lines can affect the performance of groups and the perception of their efficiency.

As technology advances, AI algorithms used in industrial processes can develop capabilities to manipulate human behaviour - to identify and exploit human practices, weaknesses, and vulnerabilities. Algorithms can detect the feelings of the humans involved in the production, including fear, disgust, joy, and relaxation.

In industrial environments, this concern is not related to AI taking over but instead aims to raise awareness of the risks involved when human decision- making has been tampered with. For example, a risk is present if an AI machine or an IIoT smart device no longer operates efficiently because this critical element in AI-powered machines has not been set as a goal.

AI-based applications in industrial environments must consider a risk-based approach and differentiate AI uses according to whether they create an unacceptable risk, a high risk or a low risk. The risk of manipulating behaviour is unacceptable if it poses a clear threat to the regular operation of the manufacturing process, security, quality of the outcome and personal safety involved.

9.5.8 Autonomous Industrial Systems

Advances in industrial automation systems and AI have brought autonomous systems into the focus of digital ethics as intelligent machines that can adapt to the environment are strongly interconnected with autonomy.

Machine autonomy in industrial processes is related to the absence of human intervention. Autonomy is characterised by the ability of an AI-based system to make decisions and justify its actions based on its sensing capabilities to adapt to changes, which occur within the system itself, other systems it interacts with, its operation environment, or in the given task.

Autonomous industrial systems can perceive their environment via sensing perception capabilities, create a plan of action according to the context or scenario-related constraints and execute the planned actions safely and reliably using intelligent actuators.

The autonomous industrial systems have characteristics related to process execution, adaptability, self-governance, self-contentedness, and the corresponding abilities that can connect with non-functional requirements for the AI-based system.

The non-functional requirements or capabilities of autonomous industrial systems are interlinked with the system's skill to perform different tasks. The abilities needed to give the AI-based system the characteristics of an autonomous industrial system differ from case to case and depend on the context.

Autonomous systems must operate without the intervention or assistance of human operators and within the requirements defined by the industrial ethical framework.

The applicability of ethical considerations needs to consider the various aspects of inherent decision-making autonomy, mitigation in abnormal situations, and communication with other machines and with human operators.

A simplified high-level reference architecture for AI-based autonomous systems in industrial environments is illustrated in Figure 9.4. The simplified high-level reference architecture is used in the ECSEL AI4DI [1], ArchitectECA2030 [30], and AI4CSM [31] projects to provide an overview and organise the AI-based systems and their functions.

Communication with the environment relies on sensors for observing the environment and actuators for changing environmental conditions to achieve the objectives. Communication and collaboration with humans and other machines in the industrial environment provide information and feedback on the performance and actions of the system. In abnormal situations, capabilities for cognitive information processing allows the system to fall back to a safe operating state or to hand over control to a human operator and take it back when the situation is normal again.

These capabilities rely on mechanisms for self-regulation controlling the various modules, including knowledge bases, and are constantly adapted

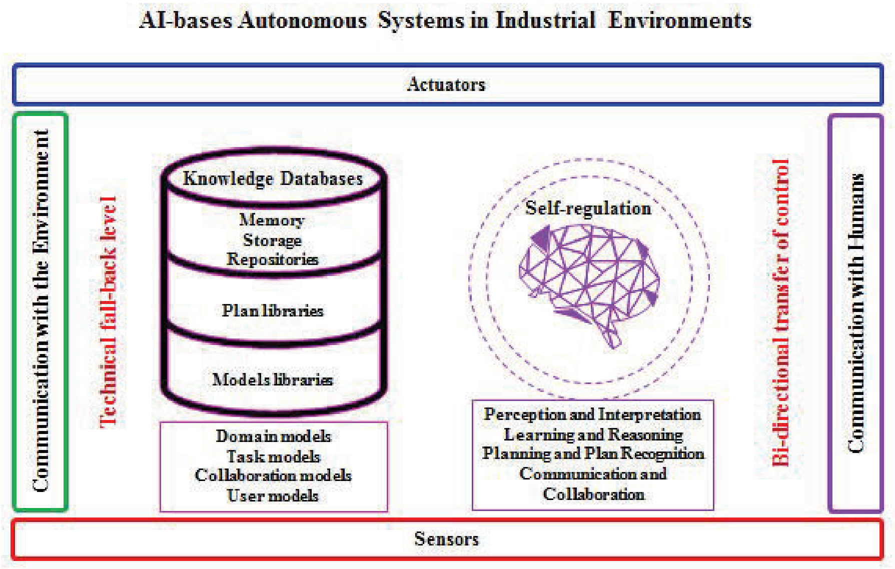

AI-bases Autonomous Systems in Industrial Environments
Actuators
Communication with the Environment
Technical fall-back level
Knowledge Databases
Memory
Storage
Repositories
Plan libraries
Models libraries
Domain models
Task models
Collaboration models
User models
Self-regulation
Perception and Interpretation
Learning and Reasoning
Planning and Plan Recognition
Communication and
Collaboration
Bi-directional transfer of control
Communication with Humans
Sensors

Figure 9.4 Reference architecture for AI-based autonomous systems in industrial environments.

through perception and regulation, learning and reasoning, planning, and plan recognition.

As autonomous industrial systems assume more responsibility in industrial processes in the circumstances previously overseen by human judgment, it is compelling to consider the associated ethical implications. These reflections should include analysis of ethical issues from multiple perspectives, including those of machines and intelligent IIoT devices' designers, operators that control the intelligent machines, and the machines themselves that need to act with ethical correctness.

9.5.9 Machine Ethics

Debates about machine ethics are not new, nor are the arguments about whether AI machines have obligations and rights as humans and animals do. The topic has revived in recent years; questions surrounding AI accountability and autonomy have begun to be addressed more rigorously. The question of whether AI is responsible for its own decisions, actions and consequences, or whether this responsibility falls to the humans that design, develop, operate and assess AI can no longer be answered directly. Many interpretations and nuances are involved in answering this question.

Machine ethics includes how humans design, build, use, and treat AI-based machines, robots, IIoT intelligent devices and how the decision-making process of these machines are respecting ethical principles defined for a manufacturing facility, an industrial sector, a region, or a country.

When discussing machine ethics, the debate raises the question of how to regulate autonomous and intelligent systems-related technologies legally and the appropriate legal treatment of systems that deploy these technologies [8]. What if AI machines (e.g., intelligent IoT devices, robots, etc.,) instead of being considered people in a human sense, are put on the same legal level as corporations? It is important to remark that corporations' legal personhood can currently shield the natural persons behind them from the implications of the law [9].

As intelligent machines evolve into entities that can perceive, feel, think, and act, with intelligence comparable with animals, new regulatory and legal frameworks must be implemented to define their legal status.

9.5.10 Automation and Employment

While the concern related to AI- and automation-driven mass job losses has been a topic of concern in recent years, changes in the inherent nature of work due to AI and automation in industrial environments will have a more substantial impact than just job losses. New forms of employment and new competencies will become the norm.

Rapid advances in AI and automation technologies can significantly disrupt labour markets. AI technology generally increases productivity in the industrial environment and, at the same time, diminish some of today's valuable employment opportunities.

The AI and automation increase and augments the productivity of some workers, and the technologies most probably replace the work done by others and likely transform all professions to some degree. The rise in automation is accelerating and occurring in a period of increasing economic inequality, fostering fears of mass technological unemployment and a reiterated call for policy efforts to address the consequences of technological change [15].

The AI effect on labour relates to automation and transforming human behaviour to assume more complex roles, departing from the physical work that dominated the industrial era repetitive administrative functions to the cognitive tasks that characterise an efficient and more productive industrial landscape.

The automation and the use of AI in industrial manufacturing can drastically cut down the human workforce, which means that revenues go to fewer persons as the wealth created by machines does not include the machines themselves. Individuals who have ownership in industrial AI-driven companies make all the money. This can widen the wealth gap, where fewer and fewer persons take a substantial portion of the economic surplus created by machines.

In this context, the ethical dilemma is connected to the occupation of humans that rely on their jobs in industry to generate income to sustain themselves and their relatives and contribute to human society.

9.6 AI and the Future Digitising Industry

The AI-based autonomous machines in industrial environments could in the future omit their traditional operating environments and increasingly move into problem areas that have earlier been available to humans due to their dynamic nature and complexity. During this transition, the intelligent machines will not be able to avoid acquiring some of the humans' limitations (e.g., learning from experience as the basis of flexible behaviour, experience as an accumulation of errors, etc.).

The development requires identifying who is responsible for the actions of machines over which humans could not have adequate control and determine ways to address the responsibility gap in moral practice and legislation [7].

9.7 Ethical Guidelines for AI in Industrial Environments

AI is increasingly impacting all industrial sectors and triggered many industrial groupings and professional bodies to provide several sets of ethical principles for AI with new ethical guidelines emerging from the British Standards Institute and the IEEE Standards Association.

The IEEE focuses on researchers' need to operate with a 'safety mindset' to pre-empt unintended or unanticipated behaviours and suggested that social and moral norms need to be considered in the design of AI technologies and applications.

The proliferation of AI-based solutions for industrial processes raises the concern that AI-related degree programmes fail to equip designers with appropriate knowledge of ethics. Related to the status of AI-based systems, IEEE [8] claims that AI should not be granted the status of "personhood", and the existing and future laws should not practically give AI legal autonomy.

As a result of globally shared needs and concerns, the industry-driven Ethics Certification Program for Autonomous and Intelligent Systems (ECPAIS) program was launched by IEEE [20]. The program's goal is to advance transparency, accountability, and reduction in algorithmic bias in autonomous and intelligent systems by setting out five core principles to consider in the design and implementation of AI and ethics: adherence to existing human rights frameworks, improving human wellbeing, ostensibly to ensure accountable and responsible design, transparent technology, and the ability to track misuse.

Another significant initiative refers to the OECD principles on Artificial Intelligence promoting innovative and trustworthy AI, respecting human rights and democratic values. The principles were adopted in May 2019 by OECD member countries when they approved the OECD Council Recommendation on Artificial Intelligence [21].

9.8 Recommendations for Ethical AI in Industrial Environments

Based on the experience presented in [14] when proposing an AI model that is intended to be retrained for downstream use, such as retraining on a new domain or fine-tuning on a new task, the designers should inform training time and computational resources required, as well as model sensitivity to hyperparameters. This form of reporting and transparency can enable direct comparison across models, allowing subsequent users of the AI-based models to accurately assess whether the required computational resources are compatible with their setting.

The development of an industrial AI framework to enhance the explainability of AI systems is critical for all autonomous system and especially the ones that make socially significant decisions. Key to such a framework is the ability off industrial stakeholders to acquire a factual, direct, and clear explanation of the decision-making process, in the event of unwanted consequences. The specific issues addressed by different industrial sectors require the adaptation and extension of the framework to different industries.

For the evaluation of the performance of AI-based solutions for industrial applications, it is recommended to develop standardised benchmark tools, hardware-independent measurement techniques of training time (e.g., gigaflops required to convergence), and standard measurements of model sensitivity to data and hyperparameters (e.g., variance concerning hyperparameters searched).

In this context it is recommended to develop metrics for the trustworthiness of industrial AI products and services, to be used across industrial sectors. These metrics should serve as the basis for an evaluation framework that enables a user-driven benchmarking of all marketed industrial AI offerings.

The promotion of an industrial AI ethical framework must incentivise the inclusion of technical, ethical, legal, and social considerations in AI research projects and stimulate new concepts for including ethical principles into AI industrial technological developments and support the co-creation of industrial policies, standards, best practices, and rules.

AI transparency in industrial environments should be addressed from the AI system's perspective and not only from individual algorithms or components viewpoint.

AI transparency must be considered and applied concept to be interpreted in a particular context, mitigated by knowledge, information asymmetries, model-related explainability, and a set of competing interests (e.g., technological, economic). Consequently, AI transparency balances interests in industrial manufacturing processes, demanding a multidisciplinary approach that needs to be adequately addressed.

It is recommended to advance further research on computationally efficient AI algorithms, hardware, software that significantly reduce the energy consumption of AI-based solutions.

Industrial stakeholders should develop new AI technologies that advance trustworthy industrial AI to increase economic output, manufacturing efficiency, and productivity; protect natural environments; reduce emissions; and revitalise inclusive growth, sustainable development, and well-being.

They should create strategies for implementing trustworthy industrial AI across industrial sectors throughout the AI system life cycle. These include autonomy, beneficence, non-discrimination, non-bias, fairness, non-maleficence, diversity, explainability, data protection, justice, and internationally recognised labour rights.

They should implement mechanisms and safeguards, the capacity for human decisions to supervise AI-based systems.

In the following years, research and development should focus on the design of robust, secure, and safe industrial AI technologies throughout the entire life cycle in all conditions (e.g., normal use, foreseeable use or misuse, and other adverse conditions) to function appropriately and not pose any unreasonable safety risk.

The issue of industrial AI traceability should be addressed by providing mechanisms to ensure traceability (e.g., concerning datasets, processes and decisions made during the AI system lifecycle) to enable an analysis of the industrial AI system's outcomes and responses to inquiry appropriate to the industrial context.

Industrial AI stakeholders should commit to transparency and responsible disclosure regarding industrial AI systems, provide meaningful information appropriate to the industrial context to support the understanding of AI systems, make other stakeholders aware of their interactions with industrial AI systems, and understand the outcome of these systems.

Stakeholders operating in different industrial sectors should continuously develop and implement a systematic risk management approach to each phase of the industrial AI system lifecycle to address risks related to industrial AI systems.

Stakeholders designing, developing, and deploying industrial AI systems should be responsible and accountable for the proper functioning these systems and should respect the industry principles based on their roles, the industrial context, and consistent with the sector regulations and applicable laws.

Industrial actors should consider a long-term investment in research, development, and interdisciplinary activities to stimulate trustworthy AI innovation that focuses on challenging technical issues and AI-related social, legal, and ethical implications and policy issues.

In this context, it is highly recommended to foster and strengthen an interactive and collaborative European ecosystem for trustworthy industrial AI and provide mechanisms for sharing AI knowledge across industrial sectors to exchange datasets, tools, and toolchains to support the safe, fair, legal, and ethical sharing of data.

9.9 Conclusion

Digitising industry processes integrate AI-based solutions into manufacturing using autonomous learning machines based on many complex AI technologies and architectures. The digital transformation of industrial sectors thus creates new situations that call for new ethical considerations to be addressed.

It was provided a comprehensive overview of these considerations, challenges and trade-offs linked to developing AI-based intelligent stand- alone systems, IIoT systems in industrial environments as a basis for developing guidelines, standards, and norms to support their governance in industrial environments.

One such challenge relates to the question of who is responsible for the actions of an AI system?

In established industrial environments, the responsibility is attributed to a human actor, such as the owner, developer, manufacturer, or operator. However, as autonomous and learning AI-based systems become more pervasive and designed to reduce human intervention in industrial processes and accelerate automation, this may no longer be the case.

The manufacturer or operator is not always able to predict future machine behaviour, and thus in specific cases cannot be held responsible. This calls for new regulations to be in place to support decisions related to who is accountable or faces a responsibility gap that traditional concepts cannot bridge.

This article pases awareness of diverse and complex ethical concerns arising from the deployment of AI in industrial environments: from the degradation of the environment to job losses due to automation. These concerns may differ in interpretation, focus, and weight within various industries and organisations, mainly because ethical terminology, principles, and approaches – although necessarily aligned to society's common and recognised values – will vary to adapt to the ecosystem in each industry or organisation. Consequently, no one solution can mitigate all concerns, so this article aims to spark new topics for further research.

Digitising industry processes integrate into the manufacturing processes AI-based solutions using autonomous learning machines based on neural networks, genetic algorithms, and agent architectures. The digital transformation of industrial sectors creates a new situation.

Acknowledgements

Part of the work presented in this chapter was supported by the European Commission within the European Union's Horizon 2020 research and innovation programme funding, ECSEL Joint Undertaking project AI4DI under Grant Agreement No. 826060, ECSEL Joint Undertaking project ArchitectECA2030 under Grant Agreement No. 877539, ECSEL and ECSEL Joint Undertaking project AI4CSM under Grant Agreement No. 101007326.

References

[1] AI4DI (2019). Artificial Intelligence for Digitising Industry. Available at: https://ai4di.eu/.

[2] European Commission (2018). Communication from the Commission to the European Parliament, the European Council, the Council, the European Economic and Social Committee and the Committee of the Regions on Artificial Intelligence for Europe, Brussels, 25.4.2018 COM (2018) 237 final. Available online at: https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:52018DC0237&from=en

[3] European Commission (2020). On Artificial Intelligence - A European approach to excellence and trust. White Paper. Available online at: https://ec.europa.eu/info/sites/default/files/commission-white-paper-artificial-intelligence-feb2020_en.pdf

[4] European Commission - High-Level Expert Group on Artificial Intelligence (2019). Ethics guidelines for trustworthy AI. Available online at: https://digital-strategy.ec.europa.eu/en/library/ethics-guidelines-trustworthy-ai

[5] European Commission (2021). Commission staff working document. Impact assessment. Accompanying the proposal for a regulation of the European Parliament and of the Council. Laying down harmonised

rules on artificial intelligence (Artificial Intelligence Act) and amending certain union legislative act. Available online at: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex%3A52021SC0084

[6] N. J. Nilsson (2010). The Quest for Artificial Intelligence: A History of Ideas and Achievements, Cambridge, UK Cambridge University Press.

[7] A. Matthias (2004). The responsibility gap: Ascribing responsibility for the actions of learning automata. Ethics and Information Technology, Sept 2004, Vol. 6, Issue 3, 175-183. Available online at: https://link.springer.com/content/pdf/10.1007/s10676-004-3422-1.pdf

[8] IEEE - The Institute of Electrical and Electronics Engineers (2017). Ethically Aligned Design: First Edition. A Vision for Prioritizing Human Well-being with Autonomous and Intelligent Systems. Available online at: https://standards.ieee.org/content/dam/ieee-standards/standards/web/documents/other/ead_v2.pdf

[9] E. Bird, J., Fox-Skelly, N., Jenner, R., Larbey, E.. Weitkamp and A., Winfield (2020). The ethics of artificial intelligence: Issues and initiatives. Available online at: https://www.europarl.europa.eu/RegData/etudes/STUD/2020/634452/EPRS_STU(2020)634452_EN.pdf

[10] W. Knight (2020). AI Can Do Great Things—if It Doesn't Burn the Planet. Available online at: https://www.wired.com/story/ai-great-things-burn-planet/

[11] S. Meinecke (2018). AI could help us protect the environment — or destroy it. Available online at: https://www.dw.com/en/ai-could-help-us-protect-the-environment-or-destroy-it/a-44694471

[12] D. Lu (2019). Creating an AI can be five times worse for the planet than a car. Available online at: https://www.newscientist.com/article/2205779-creating-an-ai-can-be-five-times-worse-for-the-planet-than-a-car/

[13] ML CO2 Impact. Available online at: https://mlco2.github.io/impact/#home

[14] E. Strubell, A. Ganesh, A. McCallum (2019). Energy and Policy Considerations for Deep Learning in NLP. Available online at: https://arxiv.org/pdf/1906.02243.pdf

[15] M. R. Frank, et al. (2019). Toward understanding the impact of artificial intelligence on labor. Available online at: https://www.pnas.org/content/pnas/116/14/6531.full.pdf

[16] Y. Wang, X. Mengran, and H. G. T. Olya (2020). Toward an Understanding of Responsible Artificial Intelligence Practices. In Proceedings

of the 53*rd* Hawaii International Conference on System Sciences. Available online at: https://scholarspace.manoa.hawaii.edu/bitstream/10125/64352/0491.pdf

[17] WEF-World Economic Forum (2018). Harnessing Artificial Intelligence for the Earth. Available online at: http://www3.weforum.org/docs/Harnessing_Artificial_Intelligence_for_the_Earth_report_2018.pdf

[18] NIST (2019). US Leadership in AI: A Plan for Federal Engagement in Developing Technical Standards and Related Tools. Washington: NIST (US Department of Commerce), 8. Available online at: https://www.nist.gov/system/files/documents/2019/08/10/ai_standards_fedengagement_plan_9aug2019.pdf

[19] A. Patrizio (2018). IDC: Expect 175 zettabytes of data worldwide by 2025. Available online at: https://www.networkworld.com/article/3325397/idc-expect-175-zettabytes-of-data-worldwide-by-2025.html#:~:text=By%202025%2C%20IDC%20says%20worldwide,cloud%20as%20in%20data%20centers.&text=IDC%20has%20released%20a%20report,study%2C%20the%20numbers%20are%20staggering.

[20] The Ethics Certification Program for Autonomous and Intelligent Systems (ECPAIS). Available online at: https://standards.ieee.org/industry-connections/ecpais.html

[21] Organization for Economic Co-operation and Development (2019). Principles on Artificial Intelligence. Paris: OECD. Available online at: https://legalinstruments.oecd.org/en/instruments/OECD-LEGAL-0449

[22] Imperial College London (2017). Written Submission to House of Lords Select Committee on Artificial Intelligence [AIC0214]. Available online at: http://bit.ly/2yleuET

[23] T. King, N. Aggarwal, M. Taddeo, and L. Floridi (2018). Artificial Intelligence Crime: An Interdisciplinary Analysis of Foreseeable Threats and Solutions. Available online at: https://ssrn.com/abstract=3183238

[24] Montreal Declaration for a Responsible Development of Artificial Intelligence. (2017). Announced at the conclusion of the Forum on the Socially Responsible Development of AI. Available online at: https://www.montrealdeclaration-responsibleai.com/the-declaration

[25] Partnership on AI. Safety Critical AI. Tenets. Available online at: https://partnershiponai.org/program/safety-critical-ai-2/

[26] Asilomar AI Principles. (2017). Principles developed in conjunction with the 2017 Asilomar conference [Benevolent AI 2017]. Available online at: https://futureoflife.org/ai-principles

[27] J. Cowls, and L. Floridi (2018). Prolegomena to a White Paper on Recommendations for the Ethics of AI (June 19, 2018). Available online at: https://ssrn.com/abstract=3198732

[28] House of Lords Artificial Intelligence Committee (2018). AI in the UK: ready, willing and able? Available online at: https://publications.parliament.uk/pa/ld201719/ldselect/ldai/100/10002.htm

[29] European Group on Ethics in Science and New Technologies (2018). Statement on Artificial Intelligence, Robotics, and "Autonomous" Systems. Available online at: https://op.europa.eu/en/publication-detail/-/publication/dfebe62e-4ce9-11e8-be1d-01aa75ed71a1

[30] ArchitectECA2030 (2020). Trustable Architectures with Acceptable Residual Risk for the Electric, Connected and Automated Cars. Available at: https://autoc3rt.automotive.oth-aw.de/

[31] AI4CSM (2021). Automotive Intelligence for Connected Shared Mobility. Available at: https://ai4csm.automotive.oth-aw.de/

10

Current Challenges of AI Standardisation in the Digitising Industry

Ovidiu Vermesan[1], Marcello Coppola[2], Reiner John[3], Cristina De Luca[4], Roy Bahr[1], and Giulio Urlini[5]

[1]SINTEF AS, Norway
[2]STMicroelectronics, France
[3]AVL List GmbH, Austria
[4]Silicon Austria Labs GmbH, Austria
[5]STMicroelectronics, Italy

Abstract

The digital transformation of industrial sectors is highly dynamic, and standardisation plays an essential role in achieving the objectives set for this transformation. In this context, AI standardisation efforts and industry AI efforts are intertwined. Industrial AI applications rely on standardisation to build and sustain trust in industrial AI. Conversely, standardisation relies on industrial AI applications to play an important role in forming emerging AI standards. This article provides an overview of the role of AI standardisation in digitising industry and the related objectives, while presenting several requirements and challenges impacting standardisation. Furthermore, it summarises the AI standards landscape and activities within Standards Development Organisations (SDOs), outlines industrial stakeholders' approaches, and provides recommendations for an AI standardisation roadmap (in which the industry should focus on AI standards work). Its concluding remarks relate to AI standardisation activities, priorities in industrial environments, and certification efforts to conceptualise new approaches to conformance and acceptance criteria.

Keywords: Artificial intelligence, standardisation, machine learning (ML), interoperability, trustworthiness, digitising industry, AI certification, ecosystem of excellence, AI standardisation roadmap, verification, validation and testing (VV&T), industrial internet of things (IIoT), autonomous systems, safety, and security

10.1 Introduction

The development of AI technologies and applications for industrial environments requires standards that create common building blocks establishing foundations for product differentiation, technological innovation, and frameworks for industrial stakeholders in enabling reliable, responsible, safe, and secure AI solutions.

In North America, organisations such as NIST [2] have actively supported the development of AI standards and stated, "AI standards that articulate requirements, specifications, guidelines, or characteristics can help to ensure that AI technologies and systems meet critical objectives for functionality, interoperability, and trustworthiness - and that they perform accurately, reliably, and safely."

NIST developed a roadmap on AI standards to guide the development of technical standards and related tools to support reliable, robust, and trustworthy systems that use AI technologies. NIST focus areas for standards development are outlined in Figure 10.1. While progressing in developing the roadmap, the industry responded with submissions, some of which emphasised the importance of the standards being created by ISO/IEC JTC 1/SC 42. The roadmap [22]:

- Identifies areas of strategic focus for standardisation (Figure 10.1).
- Outlines the importance of co-ordination concerning standards-setting.
- Calls for strategic engagement with international parties to 'advance AI.

In Europe, the overall strategy on AI proposes an ecosystem of excellence and trust for AI [12]. The concept of an ecosystem of excellence in Europe refers to measures supporting research, fostering collaboration between the Member States, and increasing investment into AI development and deployment [15]. The trust ecosystem is based on EU values and fundamental rights and foresees robust requirements that would give citizens the confidence to embrace AI-based solutions while encouraging businesses to develop them [14]. The European approach for AI "aims to promote Europe's innovation capacity in AI while supporting the development and uptake of ethical

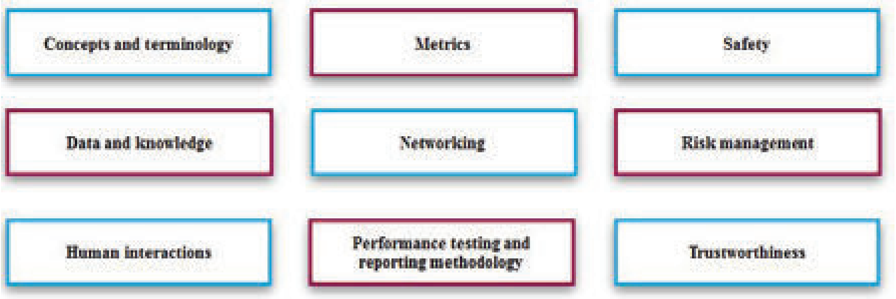


Figure 10.1 NIST focus areas for standards development.

and trustworthy AI across the EU economy. AI should work for people and be a force for good in society” [12][13].

This article presents several issues related to AI standardisation drawn from the experience gained in the ECSEL JU AI4DI [1], ArchitectECA2030 [24] and AI4CSM [25] projects that addresses the challenges of digitising industry, automation of vehicles and the integration of AI-based components, techniques, methods, and applications to various industrial sectors. The projects provide new reference architecture concepts, methodologies, new silicon-born-AI components supporting the development of AI- born embedded systems and integrating AI-born industrial systems, design languages, application generators, design automation and respective standardisation to accelerate the transfer of these technologies into industrial applications.

10.2 International Principles

The Organization for Economic Co-operation and Development (OECD) [4] provided a set of principles and encouraged governments to “promote the development of multi-stakeholder, consensus-driven global technical standards for interoperable and trustworthy AI” [4]. The principles proposed by OECD incorporate actionable measures to promote a framework for the “responsible stewardship of trustworthy AI”, including design, development, and deployment of AI internationally. OECD’s high-level value-based principles are summarised below:

- Inclusive growth, sustainable development, and well-being.
- Human-centred values and fairness require that AI-based systems are designed to respect the rule of law, defined values and diversity, and include appropriate safeguards, allowing human intervention where necessary.

- Transparency and responsible AI-based systems ensure that users understand AI-based outcomes and can challenge them.
- Robustness, security, and safety are embedded in AI-based systems throughout their life cycles by continually assessing and managing potential risks related to AI systems, including privacy, digital security, safety, and bias. AI actors should assure traceability concerning datasets, processes and decisions made during the AI system lifecycle to facilitate an analysis of the AI system's outcomes and responses to inquiry, suitable to the context and consistent with state-of-the-art.
- Accountability applies to organisations and individuals developing, deploying, or operating AI systems for the proper functioning of these systems in line with the above principles, based on their roles, the context, and consistent with the state-of-art.

The implementation of these principles is reflected in the developments of AI technology, regulations/legislation, and standards. The development of AI standards is done through SDOs that function mainly on a consensus basis.

The World Economic Forum (WEF) has strengthened the activities around the governance of AI, focusing on developing high-level principles-based guidance, frameworks, and workbooks to support decision-making and based on these activities, create partnerships with different national governments. These partnerships represent an additional valuable role in developing international standards to support the design, development, deployment, and evaluation of responsible AI systems, including within industrial sectors. The forum supports the organisation in implementing the practices and measures suggested in a Model Framework [5] and sharing experiences to inspire other organisations adopting AI to do so in a similarly responsible manner.

10.3 Role of AI Standardisation in Digitising Industry

AI, alongside IIoT, edge computing and intelligent connectivity, has become a core technology across various industries and one of the driving forces in digital transformation, and AI standardisation plays an essential role in shaping its future. AI standards are critical for building trust and confidence in AI technologies.

Standardisation activities ensure industry collaboration on the development of new AI standards, best practices, use cases and terminologies for scaling AI and enabling industries to achieve their full potential.

AI standardisation initiatives bring to industrial stakeholders common vocabularies, agreements on taxonomies and definitions, and new pre-normative activities to address autonomous and semi-autonomous industrial systems.

AI standards form the basis for AI technologies and provide reference points for assessing AI systems' computational approaches and characteristics and studying technologies used by those systems, such as ML algorithms and reasoning, as well as their properties and features.

By analysing existing specialised industrial AI systems, stakeholders involved in standardisation processes can understand and identify the AI systems' underlying computational approaches, architectures, and characteristics.

Using representative use cases collected across application domains as a reference for emerging standards ensures that the standardisation process will reflect the contexts in which AI is being used and thus help to define AI architectural approaches.

Standardisation is expected to be a prominent driving force in the adoption and integration of AI in industrial applications. It is also expected to play a supportive role in mitigating some of the concerns and challenges brought by AI deployments in industrial environments. Moreover, the most essential requirements for AI standardisation can be naturally derived from these challenges.

10.4 Challenges Associated with AI Deployments in Industrial Environments

The challenges of AI deployments in industrial environments are associated with complexity, data acquisition and storage, training, testing, compliance requirements, high cost of failures/changes, and other variables used in the optimisation processes.

The sensors and IIoT-based systems that collect data capture many parameters from various processes, and inevitably also capture noisy information. As such, extensive storage, and computing resources for analytics capabilities are required.

To properly train AI-based systems, adequately large amounts of representative data, including information on expected and unexpected failures and other events, must be collected. This is a challenging task, as the data is available in different systems or platforms, provided in different formats and, in many cases, too scarce to be used for training purposes.

Testing AI-based systems on real-world production lines, manufacturing warehouses and other industrial facilities requires extensive time and resources. For AI applications with a low technology readiness level (TRL), simulation environments are used for training and testing before deployment.

AI-based systems require adaptions in industrial manufacturing processes, and the cost of changes and failures at large-scale industrial facilities is very high.

Testing industrial AI applications is often required for specific deployment contexts in various industrial environments. Testing and certification bodies must depend on and increasingly trust more simulation or virtual testing to perform a conformity assessment (in addition to field testing) of industrial AI applications. AI verification, validation and testing (VV&T) approaches become essential for the safety demonstration of AI features in industrial applications.

Furthermore, since industrial environments must adhere to industry compliance requirements, changes to industrial processes often trigger extensive re-assessment of compliance, which implies a need for comprehensive VV&T of the AI-based systems and automation affected by the changes.

Manufacturing facilities and industrial systems are highly complex, often providing hundreds of parameters and inputs to AI and ML optimising algorithms. This is an enormous challenge for managing the complex AI solution space, both in terms of inference and training and learning.

Considering these challenges, the trustworthiness of organisations, products and services is critical in AI-based industrial environments. Moreover, this need for trust means that new standards for design, manufacturing and business practices must be implemented so that industrial environments can evolve and promote industry innovation and deliver reliable, responsible, safe, and secure industrial AI solutions.

Finally, the requirements and challenges of AI deployments in industrial environments must be captured in the AI standards as part of a pathway to certification for AI-based systems, products, and services. In this way, any gaps that arise between technical and ethical risks and between standardisation and certification efforts can be identified and closed.

10.5 AI Standardisation Needs in Industrial Automation

AI standardisation has a different focus in industrial applications than in consumer AI applications in terms of data quality and privacy, information content and the impact of AI on various stakeholders; therefore, it also has different needs.

In industrial AI, standardisation needs are identified and driven by use cases that are representative for various industrial sectors.

The challenge with AI standardisation lies in harmonising standardisation efforts across industrial sectors and applications to create a common set of AI requirements and standardisation needs. In this context, a differentiation needs to be made explicit between horizontal (related to generic issues across several industrial areas) and vertical (related to more specific issues relevant to a given sector or application area) standardisation tasks.

To facilitate these efforts, it makes sense to categorise the many complex AI topics based on their relevance and use, as illustrated in Figure 10.2. The AI topics are placed in a three-layer structure with generic topics on the top, horizontal topics in the middle and relevant AI4DI application areas at the bottom [1]. The generic topics form the basis for discussions on AI and include terminology, classification, methods, datasets and generic use-cases. The horizontal topics are common across industries and must be considered for the development of guidelines, standards, regulations, and certification to support AI-based systems governance in industrial environments. Ethical aspects and associated topics such as fairness, transparency, accountability, explainability, and control are part of the horizontal topics. AI is relevant for almost all industry sectors, and the application areas are very diverse, such as automotive, semiconductor, industrial machinery, food and beverages and transportation. The relevant industrial application area topics are found in the AI systems, components of AI systems and services, and manufacturing and support processes.

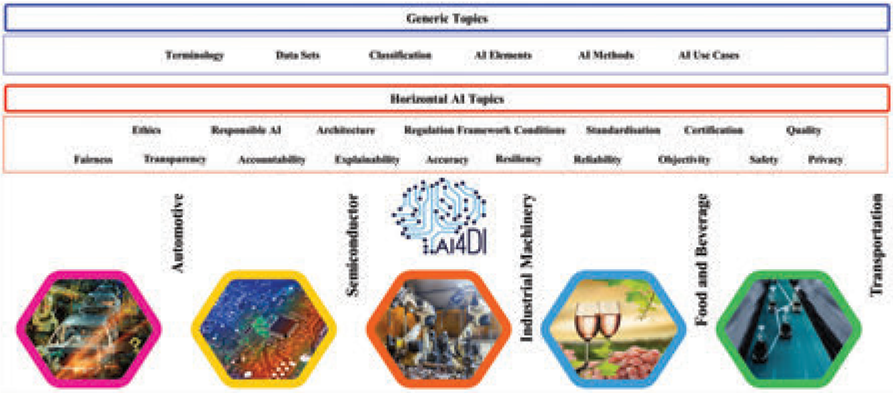


Figure 10.2 Three-layer AI topics structure: generic, horizontal, and relevant industrial application areas.

Finally, to address standardisation gaps and future standardisation activities, an interdisciplinary exchange between expert groups is needed. This exchange should focus on the role of AI in industrial environments, e.g., in the context of IIoT, functional and operational security and safety, given the complexity of AI technologies and applications.

10.6 Standardisation of Security and Safety in AI Systems

As industrial AI and ML become more and more integrated in critical systems, responsible for supporting or making decisions that can impact the security and safety of people, assets, and the environment, new challenges associated with the standardisation of security and safety in AI systems need to be addressed.

Existing safety standards in various industrial sectors are not compatible with AI methods, such as machine learning and computer vision. As such, they do not include criteria for the security and safety of AI systems or means of verification for compliance. Thus, either existing standards need to be adapted or new safety standards must emerge, or both.

Safety and security are intertwined when it comes to autonomous systems and IIoT devices, with differential approaches to address attacks against AI-based systems and services. The end-to-end and by-design principles applied to IIoT systems need to be applied to AI technologies and applications. The by-design model may be most appropriate for addressing additional concerns related to AI, such as security, safety, privacy, and inclusion.

One main challenge is to guarantee that the capabilities of AI systems, such as autonomous industrial systems and driverless vehicles, are tested before being used and monitored during operation. Physical and virtual safety validation ensures the correct and safe operation of a system in an environment. It plays a critical role in AI-based autonomous systems.

Security concerns include the protection of information within AI-based systems from unauthorised tampering, especially considering the different types of users (e.g., persons, systems, software agents, machines, IIoT devices) and levels of permission they hold. The security of AI-based systems, models, and algorithms is characterised by confidentiality, integrity, non-repudiation, accountability, and authenticity. When breached, the authenticity of data used in ML can cause significant deviations in an industrial system's outputs. In this way, accountability and responsibility are challenging to achieve for complex industrial AI-based systems if the dependencies between the system's components are not adequately identified.

AI systems could expose different kinds of security weaknesses throughout their use. However, providing security guidelines and standards based end-to-end security, including addressing the quality of data and trained ML models, could improve the trustworthiness of AI solutions.

Safety in industrial environments is related to the use of AI-based systems and associated risks. Significant safety risks in industrial environments include ML system accidents, which can be defined as unintended or harmful behaviour that may emerge from the inadequate or faulty design or implementation of AI-based systems. Safety is also tightly linked to robustness, since robustness guarantees the proper operation of an AI-based system in each industrial context/environment.

The complexity of AI autonomous safety-critical systems often averts the use of formal verification, and real-world testing can be too complicated and lengthy during development. Simulation-based techniques are developed that consider the system under test as a black box operating in a simulated environment. Safety validation missions include the following:

- Find disturbances in the environment that cause the system to fail (falsification) by discovering previously unknown failure modes and determining regions where the system can operate safely.
- Locate the most-likely failure, based on a probabilistic model of the disturbances.
- Assess the probability of system failures.

Autonomous systems deployed in industrial environments or autonomous vehicles require inherent safety by design that starts with the design specifications, implementation strategy, and virtual validation for providing fail-operational properties and minimising residual risk by increasing the safety margin. Fail-operational safety and redundancy are achieved using redundant sensors and AI-based algorithms for safety-critical functions [23].

AI safety standards are critical for industrial processes, safety-critical applications, and new AI-based applications involving autonomy. AI-based autonomous systems are also evolving throughout their life cycles, learning new behaviours, and introducing unknown safety risks that need to be addressed with standard safety measures.

As a concluding remark, the first step in addressing this challenge is to review the legal and regulatory frameworks for security and safety-critical tasks in the industrial sectors. This will help to assess how AI will impact existing standards, as well as identify gaps. It is expected that most safety standards can be extended to cover AI methods fully or partially, until they become too complex and difficult to use. At that point, new AI security and safety standards will need to be developed. Certification procedures will also need to be adapted. Therefore, it is important that existing and new standards are developed with the involvement of a large group of stakeholders to understand AI technology as well as industrial-specific use cases and integration at the systems level of industrial environments.

10.7 The Global AI Standards Landscape and Standardisation Activities

The development of AI standards in industrial environments requires coordinated efforts led by the industry and implemented by international standards bodies to support the global governance and alignment of AI development in the industrial sectors.

The international standards bodies have the institutional capacity to manage expert consensus and then publish AI standards across industrial sectors.

Standards shape the development and deployment of AI systems through product/service requirements and specifications for reliability, explainability, robustness, and fail-safe, fail-operational design. They influence the broader setting in which AI is researched, developed, and deployed through process and product requirements/specifications. The creation, dissemination, and enforcement of AI standards can build trust among industrial stakeholders, researchers, companies, and users.

AI standards are developed by international standards bodies which have the experience to monitor and enforce standards globally or other organisations that develop standards sponsored by different stakeholders. Examples of such development are the AI open-source software standards (e.g., software libraries TensorFlow, PyTorch, AI datasets, models, etc.,) developed by industry consortia, organisational sponsors, and individual contributors, which convert to standards across the industry over time [8]. Open-source AI enhances transparency by opening the AI black boxes and accelerating the deployment of new AI technologies, but it can bring unknown risks or negative consequences for industrial sectors.

Figure 10.3 illustrates an industrial AI standards system framework that includes the elements required and partly addressed in the existing standards and future standardisation activities.

AI national strategies confirm that several countries draft national standards and use the activities at the national level to leverage with the

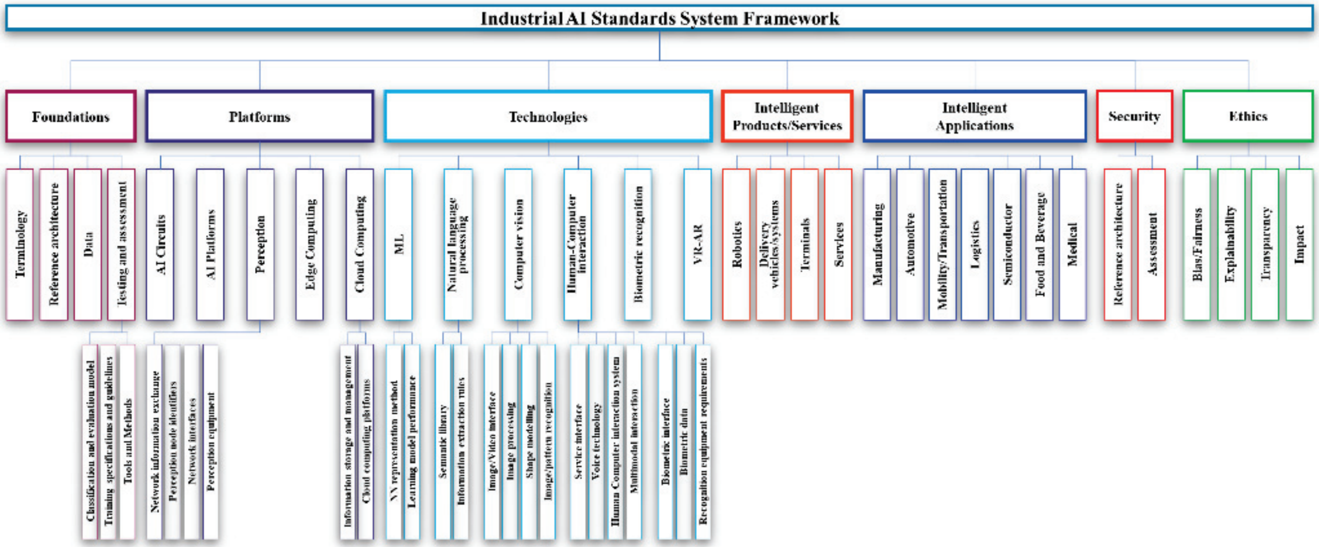


Figure 10.3 Industrial AI standards system framework.

involvement in international technical standards. Considering the market structure in the AI industry, the national standardisation bodies are encouraged and motivated to ensure that the international standards align as closely as possible to the national standards.

The following paragraphs give an overview of the AI standardisation activities covered by international standards bodies.

10.7.1 CEN-CENELEC

CEN and CENELEC continuously analyse whether relevant standards are already being produced at the international level and if European standards covering specific European needs, must be produced.

In the area of AI, CEN -CENELEC Focus Group on Artificial Intelligence has published the "Road Map on Artificial Intelligence (AI)" [10][11] that provided an overview of existing standardisation activities in IEEE, ETSI, ISO/IEC, ITU-T and CEN-CENELEC.

The Focus Group on Artificial Intelligence addresses AI standardisation in Europe through a bottom-up approach (e.g., ISO/IEC JTC 1 SC 42) and a top-down approach (concentrating on a long-term plan for European standardisation). The Focus Group identified the following seven themes that are addressed for European standardisation:

- Mapping of current European and international standardisation initiatives on AI and identifying specific standardisation needs
- Promoting further European participation in the ISO and IEC TCs
- Formulating recommendations on the best way to address AI Ethics in the European context
- Identifying the CEN and CENELEC TCs that AI will impact
- Monitoring potential changes in European legislation
- Liaising with the European High-Level Expert Group on AI and identify synergies
- Acting as the focal point for the CEN and CENELEC TCs

10.7.2 ETSI

The ETSI community focuses on AI as a "tool" in architectural models, enhancing information/data models, redesigning operational processes, increasing solution interoperability, and data management for new ICT standards [9].

The ETSI Industry Specification Group (ISG) on Securing Artificial Intelligence (SAI) focuses on three areas: AI to enhance security, mitigate against attacks that leverage AI, and secure AI itself from attack. ISG SAI cooperates with ENISA and have join activities. ISG SAI outputs are focusing on the following topics:

- The problem statement that guides the work of the group.
- AI threat ontology to align terminology.
- Data supply chain addressing data issues and risks for training AI.
- Mitigation strategy, with guidance to mitigate the impact of AI threats.
- Security testing of AI.
- Hardware in securing artificial intelligence.

Several other ETSI ISGs are working in the domain of ML for defining the specification of functionalities that are used in technology. A list of these ISGs is provided below:

- ISG on Experiential Networked Intelligence (ISG ENI) develops standards that use AI mechanisms to manage and orchestrate the network. The work supports making the deployment of future 5G networks more intelligent and efficient.
- ISG ZSM (Zero-touch network and Service Management) defines the ML enablers in end-to-end service and network management.
- ISG F5G on Fixed 5G defines the application of AI in the evolution towards “fibre to everything” of the fixed network.
- ISG CIM (Context Information Management) publishes specifications for a data interchange format (ETSI CIM GS 009 V1.2.1 NGSI-LD API) and a flexible information model (ETSI CIM GS 006 V1.1.1), which support the exchange of information from, e.g., knowledge graphs and can facilitate modelling of the real world, including relationships between entities.
- ISG ENI (Experiential Networked Intelligence) defines ML functionality that can be used/reused throughout the network, cloud, and end devices.

10.7.3 IEC

IEC addresses the AI through the standardisation evaluation group SEG 10, “Ethics in Autonomous and Artificial Intelligence Applications” which identifies ethical issues and societal concerns related to IEC technical activities and develops guidelines on ethical aspects related to autonomous and/or AI applications [16]. IEC’s SEG 10 is consisting of two working groups:

- Autonomous and AI Applications Societal and Ethical Foundations (WG 1)

- Autonomous and AI Applications Specific Ethical Requirements (WG 2).

SEG 10 outputs are focusing on the following topics:

- Identify relevant ethical issues and societal concerns to IEC technical activities.
- Formulate appropriate recommendations to Standardization Management Board (SMB).
- Develop guidelines applicable for IEC committees on ethical aspects related to autonomous and/or AI applications.
- Assure work consistency across IEC committees and foster cooperation with JTC 1/SC 42.
- Analyse any change needed in the IEC use case template to address ethical issues and societal concerns.

10.7.4 ISO

ISO/IEC JTC 1, a joint technical committee formed between IEC and ISO on IT issues, addresses the activities related to AI terminology.

The principles and rules for drafting documents used by ISO and JTC1 [21] imply specific classifications and styles of normative language that include:

- A requirement, defined as an objectively verifiable criterion that must be met without deviation to claim conformance to the containing standards.
- A recommendation, that suggests a possible choice or course of action without excluding others.
- A permission, which conveys consent or liberty to do something. JTC 1 issued a series of International Standards on AI terminology:
    - ISO/IEC 2382-28:1995, Information technology – Vocabulary – Part 28: Artificial intelligence – Basic concepts and expert systems.
    - ISO/IEC 2382-29:1999, Information technology – Vocabulary – Part 29: Artificial intelligence – Speech recognition and synthesis.
    - ISO/IEC 2382-31:1997, Information technology – Vocabulary – Part 31: Artificial intelligence – Machine learning.
    - ISO/IEC 2382-34:1999, Information technology – Vocabulary – Part 34: Artificial intelligence – Neural networks.

All these parts are merged into the common JTC 1 standard for IT vocabulary: ISO/IEC 2382:2015 [17].

Standardisation in AI is covered by ISO/IEC JTC 1/SC 42-Artificial Intelligence, which focuses on JTC 1's standardisation program on AI and provides guidance to JTC 1, IEC, and ISO committees developing AI applications. ISO/IEC JTC 1/SC 42 topics within the work programme include:

- SC 42/WG 1 - Foundational AI standards.
    - ISO/IEC 22989: Artificial Intelligence Concepts and Terminology.
    - ISO/IEC 23053: Framework for Artificial Intelligence Systems Using Machine Learning.
- SC 42/WG 2 – Big data ecosystem.
    - ISO/IEC 20547-1: Information technology - Big data reference architecture – Part 1: Framework and application process.
    - ISO/IEC 20547-3: Information technology - Big data reference architecture - Part 3: Reference architecture.
    - ISO/IEC 24688: Information technology – Artificial Intelligence – Process management framework for big data analytics.
- SC 42/WG 3 – AI Trustworthiness.
    - ISO/IEC 24027: Information technology - Artificial Intelligence (AI) - Bias in AI systems and AI aided decision making.
    - ISO/IEC 24028: Information technology - Artificial Intelligence (AI) - Overview of trustworthiness in Artificial Intelligence.
    - ISO/IEC 24029: Information technology - Artificial Intelligence (AI) - Assessment of the robustness of neural networks.
    - ISO/IEC 23894 – Information technology - Artificial intelligence – Risk management.
    - ISO/IEC 24368: Information technology - Artificial Intelligence (AI) - Overview of Ethical and Societal Concerns.
- SC 42/WG 4 – AI Use cases and applications.
    - ISO/IEC 24030: Information technology - Artificial Intelligence (AI) – Use cases.
- SC 42/WG 5 – Computational approaches and computational characteristics of AI systems.
    - ISO/IEC 24372: Information technology - Artificial Intelligence (AI) - Overview of computational approaches for AI systems.
- SC 42/JWG 1 - Governance implications of AI.

  - ISO/IEC 38507 - Information technology - Governance of IT – Governance implications of the use of artificial intelligence by organisations.
- ISO/IEC JTC 1/SC 40 IT Service Management and IT Governance.
- SC 40/WG 1 has started work on ISO/IEC 38508 Governance of data — Guidelines for data classification.
- In addition to the above projects, several study topics are assigned to the various working groups that also include topics that cross multiple areas such as ethics, societal concerns and lifecycle that are being considered across the work programme.

The list with standards and/or projects under the direct responsibility of ISO/IEC JTC 1/SC 42 secretariat is given below:

- ISO/IEC WD TS 4213 - Information technology - Artificial Intelligence — Assessment of machine learning classification performance.
- ISO/IEC WD 5259-1 - Data quality for analytics and ML - Part 1: Overview, terminology, and examples.
- ISO/IEC AWI 5259-2 - Data quality for analytics and ML - Part 2: Part 2: Data quality measures.
- ISO/IEC WD 5259-3 - Data quality for analytics and ML - Part 3: Data quality management requirements and guidelines.
- ISO/IEC WD 5259-4 - Data quality for analytics and ML - Part 4: Data quality process framework.
- ISO/IEC WD 5338 - Information technology - Artificial intelligence — AI system life cycle processes.
- ISO/IEC WD 5339 - Information Technology - Artificial Intelligence — Guidelines for AI applications.
- ISO/IEC WD 5392 - Information technology - Artificial intelligence - Reference architecture of knowledge engineering.
- ISO/IEC AWI TR 5469 - Artificial intelligence - Functional safety and AI systems.
- ISO/IEC AWI TS 6254 - Information technology - Artificial intelligence — Objectives and methods for explainability of ML models and AI systems.
- ISO/IEC 20546:2019 - Information technology - Big data - Overview and vocabulary.
- ISO/IEC TR 20547-1:2020 - Information technology - Big data reference architecture — Part 1: Framework and application process.
- ISO/IEC TR 20547-2:2018 - Information technology - Big data reference architecture — Part 2: Use cases and derived requirements.

- ISO/IEC 20547-3:2020 - Information technology - Big data reference architecture — Part 3: Reference architecture.
- ISO/IEC TR 20547-5:2018 - Information technology - Big data reference architecture — Part 5: Standards roadmap.
- ISO/IEC CD 22989.2 - Artificial intelligence - Concepts and terminology.
- ISO/IEC CD 23053.2 - Framework for Artificial Intelligence (AI) Systems Using Machine Learning (ML).
- ISO/IEC CD 23894 - Information Technology - Artificial Intelligence - Risk Management.
- ISO/IEC DTR 24027 - Information technology - Artificial Intelligence (AI) - Bias in AI systems and AI aided decision making.
- ISO/IEC TR 24028:2020 - Information technology - Artificial intelligence - Overview of trustworthiness in artificial intelligence.
- ISO/IEC TR 24029-1 - Artificial Intelligence (AI) - Assessment of the robustness of neural networks - Part 1: Overview.
- ISO/IEC AWI 24029-2 - Artificial intelligence (AI) - Assessment of the robustness of neural networks - Part 2: Methodology for the use of formal methods.
- ISO/IEC PRF TR 24030 - Information technology - Artificial Intelligence (AI) - Use cases.
- ISO/IEC AWI TR 24368 - Information technology - Artificial intelligence - Overview of ethical and societal concerns.
- ISO/IEC DTR 24372 - Information technology - Artificial intelligence (AI) - Overview of computational approaches for AI systems.
- ISO/IEC CD 24668 - Information technology - Artificial intelligence - Process management framework for big data analytics.
- ISO/IEC AWI 25059 - Software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - Quality model for AI-based systems
- ISO/IEC DIS 38507 - Information technology — Governance of IT - Governance implications of the use of artificial intelligence by organizations.
- ISO/IEC AWI 42001 - Information Technology - Artificial intelligence - Management system.

ISO/IEC JTC 1/SC 42 has built more than 30 active liaisons with ISO and IEC committees, SDOs and industry organisations to promote cooperation and creating the industry ecosystem around AI.

10.7.5 IEEE

IEEE Standards Association (SA) has focused on the use and impact of autonomous and intelligent systems (A/IS) as they become pervasive. There is a necessity to establish societal and policy guidelines for such systems to remain human-centric, serving humanity's values and ethical principles. In this context, the IEEE Global Initiative on Ethics of Autonomous and Intelligent Systems was started with a project addressing the "Ethically Aligned Design for Business: A call to action for businesses using AI" [18].

IEEE's AI standards series P7000TM address ethical considerations covering issues regarding autonomous and intelligent systems, including transparency, privacy, algorithmic bias, children's data, employee data, creating an algorithmic agent for individuals, creating an ethical robotic ontological framework, dealing with robotic nudging, creating a uniform fail-safe standard for A/IS, defining well-being metrics relating to A/IS, assessing news sources to keep them accountable and objective in reporting, creating machine-readable privacy terms for all individuals and updating facial recognition systems and databases to avoid bias. A list of the IEEE standardisation projects is presented below:

- IEEE P7000 - Model Process for Addressing Ethical Concerns During System Design.
- IEEE P7001 - Transparency of Autonomous Systems (defining levels of transparency for measurement).
- IEEE P7002 - Data Privacy Process.
- IEEE P7003 - Methodologies to address algorithmic bias in the development of AI systems.
- IEEE P7004 - Certification framework for child/student data governance.
- IEEE P7005 - Certification framework for employer data governance procedures based on GDPR.
- IEEE P7006 - Personalized AI agent specification.
- IEEE P7007 - Ontologies at different levels of abstraction for ethical design.
- IEEE P7008 - Ethically Driven AI Nudging methodologies.
- IEEE P7009 - Fail-Safe design of autonomous and semi-autonomous systems.
- IEEE P7010 - Well-being metrics for ethical AI.
- IEEE P7011 - Process of Identifying and Rating the Trustworthiness of News Sources.

- IEEE P7012 - Machine Readable Personal Privacy Terms.
- IEEE P7013 - Benchmarking Accuracy of Facial Recognition systems.
- IEEE ECPAIS - Certification for products and services in transparency, accountability, and algorithmic bias in systems.

Different other IEEE technical standardisation projects address various aspects of ML and different AI techniques:

- IEEE P2807 - Framework of Knowledge Graphs.
- IEEE P2807.1 - Standard for Technical Requirements and Evaluating Knowledge Graphs.
- IEEE P2830, Standard for Technical Framework and Requirements of Shared Machine Learning.
- IEEE P2841 - Framework and Process for Deep Learning Evaluation.
- IEEE P3652.1 - Guide for Architectural Framework and Application of Federated Machine Learning.

IEEE SA started developing an Ethics Certification Program for Autonomous and Intelligent Systems (ECPAIS), and the development is open to paid member organisations and individuals. ECPAIS seeks to develop three separate processes for certifications related to transparency, accountability, and algorithmic bias.

10.7.6 IETF

The activities related to AI are addressed by the IETF working group on "Autonomic Networking Integrated Model and Approach" [19]. With the development of the networks, it is necessary to introduce artificial intelligence technology to achieve self-adjustment, self-optimisation, and self-recovery of the network by collecting massive network state and machine learning data.

The work in IETF defined the architecture of Network Artificial Intelligence (NAI), including the key components and the critical protocol extension requirements.

IETF working group on "Autonomic Networking Integrated Model and Approach" develops a system of autonomic functions that carry out the intentions of the network operator without the need for detailed low-level management of individual devices.

Autonomic networking refers to the self-managing characteristics (configuration, protection, healing, and optimisation) of distributed network elements, adapting to unpredictable changes while hiding intrinsic complexity from operators and users.

Autonomic Networking, which usually involves closed-loop control, applies to the complete network (functions) lifecycle (e.g., installation, commissioning, operating, etc.). An autonomic function that works in a distributed way across various network elements is a candidate for protocol design. Such functions should allow central guidance and reporting and co-existence with non-autonomic methods of management.

The working group aims to enable the progressive introduction of autonomic functions into operational networks and reusable autonomic network infrastructure to reduce operating expenses.

10.7.7 ITU-T

ITU-T Focus Group on Machine Learning addresses the activities related to AI for future networks, including 5G. The working group has generated several documents covering methods for evaluating the intelligence level of future networks, data handling to enable machine learning in future networks, use cases of ML in future networks and unified architecture for ML in 5G.

A list of ITU-T documents related to AI is presented below:

- Recommendation ITU T Y.3172 - Architectural framework for machine learning in future networks including IMT-2020.
- Recommendations ITU-T Y.3173 - Framework for evaluating intelligence levels of future networks including IMT-2020.
- Y.3174 - Framework for data handling to enable machine learning in future networks including IMT-2020.
- Y.3176 - Machine learning marketplace integration in future networks including IMT-2020.
- Y.3170 - Requirements for machine learning-based quality of service assurance for the IMT-2020 network.
- Y.3175 - Functional architecture of machine learning-based quality of service assurance for the IMT-2020 network.
- Y.3531 - Cloud computing - Functional requirements for machine learning as a service.
- Y.ML-IMT2020-NA-RAFR - Architecture framework of AI-based network automation for resource adaptation and failure recovery in future networks including IMT-2020.
- Y.ML-IMT2020-serv-prov - Architecture framework of user-oriented network service provisioning for future networks including IMT-2020.

ITU-T plans to release a document on “Artificial Intelligence Standard Roadmap” [20] to assist in developing AI standards in the IT fields by providing information about existing and under developing standards in key SDOs. In addition, it describes the overviews of AI itself and AI-related technical areas from a standards perspective, AI-related activities in SDOs, and gap analysis.

10.8 AI Certification

Certification is the process of issuing a certificate to indicate conformance with a standard, a set of guidelines or some similar norms.

Certification must have value to be accepted, successfully deployed, approved and promoted by industry.

A certification framework for AI-based systems in industrial environments can have value and provide support for the assessment and benchmark of AI-based products, services, models, algorithms for key requirements.

Producers can choose to have their AI-based products certified because they believe it will make the product more competitive.

Producers themselves may declare that their AI-based products conform to specified standards and issue accordingly a certificate referred to as self-certification or first-party certification. In other cases, a person, or an organisation with interest as a product user may require that products be submitted for certification by an independent body; this is referred to as requested third-party certification. Third-party certification is, therefore, when a body, independent of both the producer and the user, carries out the certification process.

The situation is slightly different in industrial sectors. Industrial stakeholders will not invest resources in a certification that does not achieve a goal. In other words, for certification of AI-based systems, for example, to be successful, its effect must match the stated purpose of the industrial sector.

In other cases, manufacturers of safety-critical systems may need AI-based systems certification because this is a regulatory requirement. Many industries have a regulatory authority that oversees all projects. The industry's regulations may specify that an independent third party demonstrate the conformity of a product. In this case, certification is mandatory, as opposed to the above-mentioned requested certification. This is referred to as a mandatory third-party certification.

The vast majority of AI4DI project partners agree that the standardisation goal must be to improve the efficiency of manufacturing processes and the quality of the resulting products to stay highly competitive in the global market. Furthermore, the quality embodies not only compliance with functional requirements but also non-functional requirements. An AI-based product, system or process that failed the safety or ethical certification has not achieved its goal.

Based on the above and regardless of whether the certification is requested or mandatory, first-, second, or third-party, a common AI certification framework for AI-based systems in industrial sectors is needed. Furthermore, this AI certification framework should have the following two roles:

- To function as a quality and efficiency assessment framework during development.
- To serve as a conformity assessment framework during certification.

The AI certification framework's purpose should be to automate the procedures that support development and certification by offering standardised inspection, testing, calibration, verification and validation tools and methods. This AI certification framework would allow for many inferences using the AI algorithm under test on standardised input datasets. The results would be valuable inputs for designers and developers as well as certifiers.

In addition, the AI certification framework should have a comprehensive set of best-fit use cases for experimentation relevant to most industrial sectors (with minor adjustments) and specialised for one or several sectors.

Moreover, the AI certification must ensure that certified processes and products are more efficient and have improved quality. For instance, in the case of prediction AI systems, there must be an assurance that the prediction is as accurate as it is claimed to be.

Furthermore, virtual validation will be an essential tool, especially in autonomous systems where regulatory controls impose further qualifications for AI-based systems.

The standardised tools, AI methods, datasets, use-cases must ensure repeatability of the assessment results carried out by the same body and reproducibility of the results from assessment by different bodies.

The extent and scope of certification efforts largely depend on the AI system in question. Therefore, the AI certification framework should also include a classification scheme, allowing AI systems to be classified in desired dimensions. One such classification scheme is illustrated in Figure 10.4 and used as reference in several ECSEL JU projects such as AI4DI, ArchitectECA2030 and AI4CSM [1][24][25].

The criteria for evaluating AI systems reflect their suitability and can be uni- or multidimensional, technical, legal, or ethical, depending on the application and the application domain.

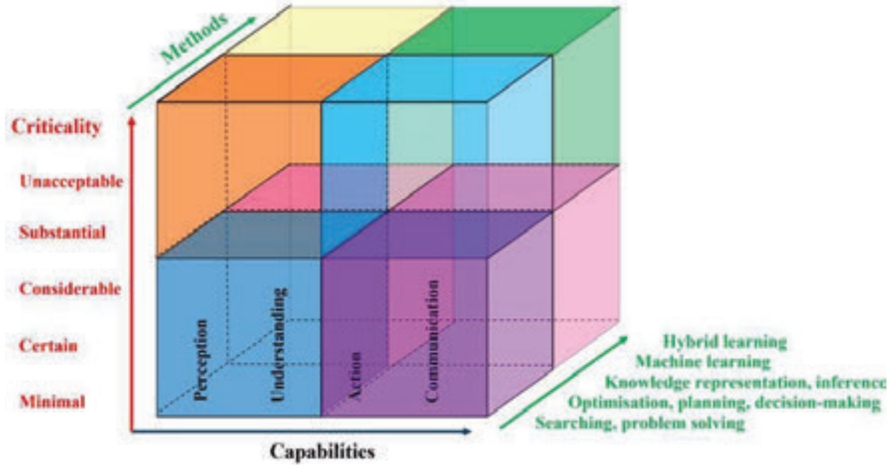


Figure 10.4 Classification scheme along with criticality, AI methods and capabilities. Adapted from [26].

One typical dimension is the potential for harm, which is commonly agreed to play a critical role in the acceptance of AI. The potential for damage can vary from minimal to unacceptable and is often related to the degree of autonomy. Other aspects, such as privacy and integrity, can be reflected through this critical dimension.

Given the wide range of capabilities of AI (from perception and understanding to communication and action), capabilities is another dimension, as the more capable the system, the greater is the risk for harm. AI methods are the third dimension, ranging from simple searching and optimisation to machine and hybrid learning. AI methods are used to achieve various AI capabilities. The more sophisticated the methods, the greater the risk.

Industrial sectors may embrace AI standardisation and certification at their own pace. But even if the ultimate goal is not to obtain a certificate, starting the design with certification in mind and using this framework towards efficient processes and high-quality AI-based products, systems, processes means the standardisation has achieved its goal.

10.9 Recommendations for an AI Standardisation Roadmap for Industrial Environments

The AI standards developments for industrial environments need to address responsible AI through standards development activities and voluntary use.

For applications in the industrial sector, AI researchers and projects that address the development of AI technologies and applications need to be involved in ongoing standardisation processes and create links with standards committees to contribute to and track outcomes. Identifying gaps in the AI standardisation landscape can benefit the development of pre-normative activities and standards with views from independent experts that provide and transfer their findings and standardisation proposal to international standards bodies under existing procedures.

In industrial environments, it is recommended that the standardisation and regulatory work concerning AI technologies and applications is progressed through multi-stakeholder discussions, allowing approaches to risk management to be tested to provide fit-for-purpose, scalability, and foster innovation.

The AI standards in industrial sectors are used to increase knowledge of reliability, trustworthiness, safety, security, and responsibility among AI developers and support the adoption of AI in different manufacturing processes.

Regulatory interventions in industrial sectors require to be proportionate to the possible and recognised harm(s) posed by AI in specific settings of the industrial sectors and identified areas of heightened vulnerability.

Different forms of certification models for AI are proposed, which involves industrial stakeholders developing the outlines of what could be recognised as responsible AI [3][6][7]. This is challenging as many large companies developed their principles for AI, which display elements of both more common values and more specific guidance elements through complementary resources.

The AI-based applications in industrial environments involve industry stakeholders and ecosystems that need best practices, standardised solutions, industry-grade benchmarking and reference data sets for training and learning. Further research is needed on industrial AI standards from technical and industrial perspectives. Technical standards desiderata can inform new standardisation efforts, and industrial strategies can develop paths for AI standards to spread in practice in different industrial sectors.

To evaluate the performance of AI-based algorithms, guidelines and reference datasets must be developed that can be used by various industrial actors in implementing AI solutions. The datasets depend on the industrial application area, and special requirements are placed on them together with guidelines that evaluate the datasets quantity/quality for training, validation, and testing.

AI and ML allow for vulnerabilities and misconfigurations, and as the manufacturing facilities are using more AI-based solutions, the more concerned they are about security risks. Open-source code is susceptible to attackers who can inject malicious code or has vulnerabilities or vulnerable dependencies.

Protecting the information in industrial environments is a crucial pillar for the performance and competitiveness of each manufacturing facility with data protection standards applied to AI systems, including training data.

All AI-based systems must integrate security by design built-in and developed around core data security principles, including encryption, logging, monitoring, authentication, and access controls. These policies must be applied even stricter considering the heterogeneous nature of AI- based solutions, including HW/SW, models, algorithms, IIoT devices and systems using open-source algorithms, commercial "black box" AI systems, or built-in AI models.

The results and outcomes from research and innovation projects with the involvement of the AI community should be aligned and provide input to the standards under development to further accelerate the advancements in AI for digitising industry. European AI projects and initiatives should dedicate efforts to understanding and engaging in standardisation processes through liaisons or partnerships with specific third-party organisations.

It is recommended that efforts be made to propose standardised AI virtual testing environments for industrial applications. These actions should include the development of standards for AI virtual testing facilities, for interoperability between AI-based digital twins and standardised AI virtual testing environments and standards for AI physical simulations/modelling (sensors, actuators, etc.).

Within industrial organisations, closer cooperation between product development units with experience in standards, industrial processes, and AI research teams can increase the efficient use of AI standards, identify the gaps, and enhance or create new standards.

Adopting AI standards under development and the involvement in activities for shaping future standards can further support the collaboration between AI research groups and the industry.

AI researchers should engage in ongoing standardisation processes. Projects addressing industrial AI should consider becoming liaisons with standards committees to contribute to and track developments. Different standards may benefit from independent development initially and then be transferred to an international standards body under existing procedures. The involvement in AI standardisation activities support the work to create a roadmap for global AI standardisation and identify the gaps and the needs for further standardisation efforts. A roadmap is a tool for individual researchers, organisations, industrial consortia, or larger groups to evaluate the existing activities and initiate standardisation efforts in more AI-based technology and applications with priorities coming from both industry and the AI research community.

The acceleration of the digital transformation of the industry requires further research on AI standards from both technical and industrial enterprise perspectives. Technical AI standards requirements can generate new standardisation efforts, and industrial enterprise strategies can develop paths across industries in practice.

10.10 Conclusion

Building and sustaining trust in industrial AI requires developing ecosystems of industrial stakeholders that work together to define the functional and non-functional requirements for AI-based hardware, software, models, and systems; and to provide and promote reference designs and use cases employed across various industrial sectors.

In different industrial sectors, market incentives drive companies to develop product and service standards in relation to the use of AI technologies. Standards are a foundation for coordination and ensure that AI-based products and services produced across an industrial sector or different sectors are interoperable.

Standards constitute a common language and practice of communication among industry stakeholders that build guardrails that help support positive AI research and development outcomes.

The requirements for AI in industrial environments have a different focus and weight compared to those of AI in consumer and general business applications. Reliability, maintainability, explainability, safety, and security privacy are in many cases the primary concerns. Privacy, inclusion, and fairness are the specific issues addressed.

Industrial companies working with AI solutions are taking measures to protect personal information and personally identifiable information connected with deployments in the manufacturing processes.

This article presented the AI standardisation role and needs in industrial environments, derived from requirements and challenges defined and agreed upon by industrial stakeholders, provided an overview of ongoing AI standardisation efforts, and offered recommendations for an AI standardisation roadmap for industrial environments.

The aim of this article is to encourage support for standardisation efforts in the form of improved and new representative use cases from various industry sectors and possibly spark new research topics related to AI standardisation.

Acknowledgements

Part of the work presented in this chapter was supported by the European Commission within the European Union's Horizon 2020 research and innovation programme funding, ECSEL Joint Undertaking project AI4DI under Grant Agreement No. 826060, ECSEL Joint Undertaking project ArchitectECA2030 under Grant Agreement No. 877539, ECSEL and ECSEL Joint Undertaking project AI4CSM under Grant Agreement No. 101007326.

References

[1] AI4DI (2019). Artificial Intelligence for Digitising Industry. Available at: https://ai4di.eu/

[2] NIST (2019). US Leadership in AI: A Plan for Federal Engagement in Developing Technical Standards and Related Tools. Washington: NIST (US Department of Commerce), 8. Available online at: https://www.nist.gov/system/files/documents/2019/08/10/ai_standards_fedengagement_plan_9aug2019.pdf

[3] A. Somani (2019). "AI needs a certification process, not legislation". Available online at: https://venturebeat.com/2019/06/09/ai-needs-a-certification-process-not-legislation/

[4] Organization for Economic Co-operation and Development (2019). Principles on Artificial Intelligence. Paris: OECD. Available online at: https://legalinstruments.oecd.org/en/instruments/OECD-LEGAL-0449

[5] World Economic Forum (WEF). Model Artificial Intelligence Governance Framework and Assessment Guide. Available online at: https://www.weforum.org/projects/model-ai-governance-framework

[6] A. Finkel (2018). "What will it take for us to trust AI?". Available online at: weforum.org

[7] G. Banavar (2016). "What it will take for us to trust AI?". Available online at: https://hbr.org/2016/11/what-it-will-take-for-us-to-trust-ai

[8] A. Theben, L. Gunderson, L. López-Forés, G. Misuraca, F. Lupiáñez Villanueva (2021). Challenges and limits of an open-source approach to Artificial Intelligence, study for the Special Committee on Artificial Intelligence in a Digital Age (AIDA), Policy Department for Economic, Scientific and Quality of Life Policies, European Parliament, Luxembourg. Available online at: https://www.europarl.europa.eu/RegData/etudes/STUD/2021/662908/IPOL_STU(2021)662908_EN.pdf

[9] ETSI White Paper No. #34 (2020). Artificial Intelligence and future directions for ETSI, First Edition, ISBN No. 979-10-92620-30-1. Available online at: https://www.etsi.org/images/files/ETSIWhitePapers/etsi_wp34_Artificial_Intellignce_and_future_directions_for_ETSI.pdf

[10] CEN-CENELEC (2020). CEN-CENELEC response to the EC White Paper on AI. Available online at: https://ftp.cencenelec.eu/EN/News/PolicyOpinions/2020/CEN-CLC_AI_FG_White-Paper-Response_Final-Version_June-2020.pdf

[11] CEN-CENELEC (2020). CEN-CENELEC Focus Group Report: RoadMap on Artificial Intelligence (AI). Available online at: https://ftp.cenc enelec.eu/EN/EuropeanStandardization/Sectors/AI/CEN-CLC_FGR_RoadMapAI.pdf

[12] European Commission (2020). On Artificial Intelligence - A European approach to excellence and trust. White Paper. Available online at: https://ec.europa.eu/info/sites/default/files/commission-white-paper-artificial-intelligence-feb2020_en.pdf

[13] European Commission - High-Level Expert Group on Artificial Intelligence (2019). Ethics guidelines for trustworthy AI. Available online at: https://digital-strategy.ec.europa.eu/en/library/ethics-guidelines-trustworthy-ai

[14] European Commission (2021). Commission staff working document. Impact assessment. Accompanying the proposal for a regulation of the European Parliament and of the Council. Laying down harmonised rules on artificial intelligence (Artificial Intelligence Act) and amending certain union legislative act. Available online at: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex%3A52021SC0084

[15] European Commission (2021). Rolling Plan for ICT standardization. Available online at: https://joinup.ec.europa.eu/collection/rolling-plan-ict-standardisation/rolling-plan-2021

[16] IEC. SEG 10 Ethics in Autonomous and Artificial Intelligence Applications. Available online at: https://www.iec.ch/ords/f?p=103:186:310164989157292::::FSP_ORG_ID,FSP_LANG_ID:22827,34

[17] ISO/IEC 2382:2015, Information technology – Vocabulary. Available online at: https://www.iso.org/standard/63598.html

[18] IEEE (2019). Ethically Aligned Design –: A Vision for Prioritizing Human Well-being with Autonomous and Intelligent Systems (A/IS). Version II. Available online at: https://standards.ieee.org/content/dam/ieee-standards/standards/web/documents/other/ead_v2.pdf

[19] IETF. Autonomic Networking Integrated Model and Approach. Available online at: https://datatracker.ietf.org/wg/anima/about/

[20] ITU-T. Focus Group on Machine Learning for Future Networks including 5G. Available online at: https://www.itu.int/en/ITU-T/focusgroups/ml5g/Pages/default.aspx

[21] ISO/IEC Directives, Part 2: "Principles and rules for drafting and structuring of ISO and IEC documents". https://www.iso.org/sites/directives/current/part2/index.xhtml

[22] NIST (2019). 'U.S. Leadership in AI'. Available online at: https://www.nist.gov/system/files/documents/2019/08/10/ai_standards_fedengagement_plan_9aug2019.pdf

[23] O. Vermesan, et al., "Automotive Intelligence Embedded in Electric Connected Autonomous and Shared Vehicles Technology for Sustainable Green Mobility. Frontiers in Future Transportation, Vol. 2 2021. ISSN=2673-5210. https://www.frontiersin.org/article/10.3389/ffutr.2021.688482

[24] ArchitectECA2030 (2020). Trustable Architectures with Acceptable Residual Risk for the Electric, Connected and Automated Cars. Available at: https://autoc3rt.automotive.oth-aw.de/

[25] AI4CSM (2021). Automotive Intelligence for Connected Shared Mobility. Available at: https://ai4csm.automotive.oth-aw.de/

[26] The German Artificial Intelligence (AI) Standardization Roadmap (2020). https://www.din.de/resource/blob/772610/e96c34dd6b12900ea75b460538805349/normungsroadmap-en-data.pdf

Index

About the Editors

Ovidiu Vermesan holds a PhD degree in microelectronics and a Master of International Business (MIB) degree. He is Chief Scientist at SINTEF Digital, Oslo, Norway. His research interests are in smart systems integration, mixed-signal embedded electronics, analogue neural networks, edge artificial intelligence and cognitive communication systems. Dr. Vermesan received SINTEF's 2003 award for research excellence for his work on the implementation of a biometric sensor system. He is currently working on projects addressing nanoelectronics, integrated sensor/actuator systems, communication, cyber–physical systems (CPSs) and Industrial Internet of Things (IIoT), with applications in green mobility, energy, autonomous systems, and smart cities. He has authored or co-authored over 100 technical articles and conference papers. He is actively involved in the activities of European partnership for Key Digital Technologies (KDT). He has coordinated and managed various national, EU and other international projects related to smart sensor systems, integrated electronics, electromobility and intelligent autonomous systems such as E^3Car, POLLUX, CASTOR, IoE, MIRANDELA, IoF2020, AUTOPILOT, AutoDrive, ArchitectECA2030, AI4DI, AI4CSM. Dr. Vermesan actively participates in national, Horizon Europe and other international initiatives by coordinating and managing various projects. He is the coordinator of the IoT European Research Cluster (IERC) and a member of the board of the Alliance for Internet of Things Innovation (AIOTI). He is currently the technical co-coordinator of the Artificial Intelligence for Digitising Industry (AI4DI) project.

Mario Diaz Nava has a Ph.D, and M.S. both in computer science, from Institut National Polytechnique de Grenoble, France, and B.S. in communications and electronics engineering from Instituto Politecnico National, Mexico. He has worked in STMicroelectronics since 1990. He has occupied different positions (Designer, Architect, Design Manager, Project Leader, Program Manager) in various STMicroelectronics research and development organisations. His selected project experience is related to the specifications and design of communication circuits (ATM, VDSL, Ultra-wideband), digital and analogue design methodologies, system architecture and program management. He currently has the position of ST Grenoble R&D Cooperative Programs Manager, and he has actively participated, for the last five years, in several H2020 IoT projects (ACTIVATE, IoF2020, Brain-IoT), working in key areas such as Security and Privacy, Smart Farming, IoT System modelling, and edge computing. He is currently leading the ANDANTE project devoted to developing neuromorphic ASICS for efficient AI/ML solutions at the Edge. He has published more than 35 articles in these areas. He is currently a member of the Technical Expert Group of the PENTA/Xecs European Eureka cluster and a Chapter chair member of the ECSEL/KDT Strategic Research Innovation Agenda. He is an IEEE member. He participated in the standardisation of several communication technologies in the ATM Forum, ETSI, ANSI and ITU-T standardisation bodies.

Intelligent Edge-Embedded Technologies for Digitising Industry

Ovidiu Vermesan, Mario Diaz Nava

This book explores new developments, ideas, and concepts in intelligent edge-embedded technologies for the digitising industry. The work is based on recent research results and activities in edge industrial computing, artificial intelligence (AI), the Industrial Internet of Things (IIoT) and digital twin technologies. Each chapter builds on the research, developments and innovative ideas generated by AI4DI, ANDANTE and TEMPO ECSEL JU, as well as other European research projects.

The evolution towards an environmentally friendly Industry 5.0 brings more industrial edge devices that include embedded intelligence, with enough computing power and sufficiently advanced programming to make operational decisions based on local data and independent of external advice.

These new intelligent edge-embedded devices act on the data they capture or generate about evolving conditions in the industrial process and change the machine's behaviour to optimise the operation and its functionalities. Connected, these intelligent edge IIoT devices can act on information generated across multiple intelligent edge devices and even across various types of intelligent edge devices to create more intelligent behaviours for industrial manufacturing processes.

The embedded intelligence technologies implement AI capabilities into the edge device itself, so the device can learn, analyse, and act autonomously.

Intelligent edge systems implemented on-premises improve industrial manufacturing facilities' outcomes by making instantaneous, autonomous, or semi-autonomous decisions independent of external cloud computing capabilities.

The specific interest here lies in the advancement of the convergence of edge computing and AI technologies in edge industrial application areas. This is examined by introducing the concepts of sustainable industrial-edge AI technologies and industrial-edge AI for sustainability.

Insights from recent research on key AI technologies that support the development of industrial-edge AI applications for the digitising industry are presented. The concept of AI at the edge is introduced, and the edge continuum components and their distribution across the micro-, deep- and meta-edge continuum are explained.

Moreover, the book discusses how to build AI models (e.g., model training and inference) on edge and provides insights into this new interdisciplinary field of intelligent industrial edge from a broader perspective.

The authors examine the technologies and hardware for neuromorphic computing, highlighting emerging in-memory computing techniques, the implementation of resistive synapses for neural networks, neuromorphic reference architectures, and the tools and methodologies for training and mapping neural networks on hardware targets.

Furthermore, the book reviews the core concepts for edge AI advancements in the digitising industry and the impact of AI and digital twins on IIoT. Finally, the book addresses ethical considerations and the trustworthiness of industrial AI systems' core concepts, as well as the current challenges of AI standardisation in the digitising industry.

This book's target audience includes academics, research scholars, industrial experts, scientists, and postgraduate students working in industrial edge-embedded intelligence hardware, software, and algorithms to add machine learning and deep learning to enhance the industrial edge processing capabilities.

ISBN 978-87-7022-611-0

9 788770 226110

River Publishers