

---

# **Intelligent Edge-Embedded Technologies for Digitising Industry**

---

## **RIVER PUBLISHERS SERIES IN COMMUNICATIONS AND NETWORKING**

---

*Series Editors*

**ABBAS JAMALIPOUR**

*The University of Sydney  
Australia*

**MARINA RUGGIERI**

*University of Rome Tor Vergata  
Italy*

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
- Wifi network protocols

For a list of other books in this series, visit [www.riverpublishers.com](http://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 Editor(s) (if applicable) and The Author(s) 2022. This book is published open access.

### **Open Access**

This book is distributed under the terms of the Creative Commons Attribution-Non-Commercial 4.0 International License, CC-BY-NC 4.0) (<http://creativecommons.org/licenses/by/4.0/>), which permits use, duplication, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, a link is provided to the Creative Commons license and any changes made are indicated. The images or other third party material in this book are included in the work's Creative Commons license, unless indicated otherwise in the credit line; if such material is not included in the work's Creative Commons license and the respective action is not permitted by statutory regulation, users will need to obtain permission from the license holder to duplicate, adapt, or reproduce the material.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

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

---

<b>Preface</b>	<b>xv</b>
<b>List of Figures</b>	<b>xvii</b>
<b>List of Tables</b>	<b>xxiii</b>
<b>List of Contributors</b>	<b>xxv</b>
<b>List of Abbreviations</b>	<b>xxix</b>
<b>1 Industrial AI Technologies for Next-Generation Autonomous Operations with Sustainable Performance</b>	<b>1</b>
<i>Ovidiu Vermesan, Frédéric Pétrot, Marcello Coppola, Mathias Schneider, Alfred Höß</i>	
1.1 Industrial AI . . . . .	2
1.1.1 Challenges of Industrial AI versus Consumer AI . . . . .	4
1.1.2 Sustainable AI . . . . .	6
1.2 Capabilities Spectrum of Industrial AI . . . . .	8
1.3 The Industrial AI Spectrum . . . . .	11
1.3.1 Narrow AI vs. General AI . . . . .	11
1.3.2 Weak AI vs. Strong AI . . . . .	12
1.3.3 Basic AI vs. Super AI . . . . .	13
1.3.4 Red AI vs. Green AI . . . . .	13
1.4 AI Problem Solving Domains . . . . .	14
1.4.1 Expert Systems . . . . .	14
1.4.2 Machine Vision . . . . .	17
1.4.3 Robotics . . . . .	18
1.4.4 Biomimicry . . . . .	20
1.4.5 Genetic and Evolutionary Algorithms . . . . .	22
1.4.6 Generative AI . . . . .	24
1.4.7 Artificial Swarm Intelligence . . . . .	27

1.4.8	Natural Language Processing . . . . .	28
1.4.9	Machine learning . . . . .	29
1.4.10	Neural Networks . . . . .	30
1.4.11	Automated Planning and Plan Recognition . . . . .	32
1.4.12	AI for the Metaverse . . . . .	34
1.5	Edge AI continuum . . . . .	34
1.6	Symbolic AI – ML Continuum . . . . .	38
1.7	Logic-based AI: Knowledge Representation and Reasoning . . . . .	39
1.8	Hardware/Software Technology Stack . . . . .	42
1.8.1	ML Methods and Techniques . . . . .	44
1.8.2	Neural Networks Architectures . . . . .	50
1.8.3	Industrial Embedded AI/ML . . . . .	52
1.8.4	On-device ML Applications Enabling True Edge Computing . . . . .	55
1.8.5	Machine Learning on Embedded Devices . . . . .	57
1.8.6	Embedded ML Development Flow in Industrial Setting . . . . .	61
1.9	Summary . . . . .	67
	References . . . . .	69

**2 Technology and Hardware for Neuromorphic Computing 73**

*Björn Debaillie, Ilja Ocket, and Peter Debacker*

2.1	Mobile Devices Call for Efficient Neuromorphic Computing . . . . .	74
2.2	Neuromorphic Hardware Enables Next Generation AI . . . . .	74
2.3	Building Neuromorphic Hardware . . . . .	76
2.3.1	Approach to Realise the Emerging Technologies . . . . .	77
2.3.2	Approach to Derive the Hardware Architectures and Designs . . . . .	78
2.3.3	Approach Related to Neuromorphic Algorithms and Applications . . . . .	78
2.4	Positioning Within the Neuromorphic Computing Landscape . . . . .	79
2.5	Targeted Use Cases and Application Domains . . . . .	81
2.5.1	Food – Food Classification . . . . .	82
2.5.2	Automotive – Object Recognition and Sound Localization . . . . .	82
2.5.3	Digital Industry – Pattern Recognition (Keyword Spotting) . . . . .	82
2.5.4	Consumer – Coaching Biomechanical Assistance (Running) . . . . .	84



2.5.5	Medical Health – Medical Image Denoising . . . . .	84
2.6	Neuromorphic Hardware Technologies Being Developed . .	84
2.7	Conclusion . . . . .	86
	References . . . . .	87

**3 Tools and Methodologies for Training, Profiling, and Mapping a Neural Network on a Hardware Target 89**

*Alexandre Valentian, Simon Narduzzi, Muhammad Arsalan, Kay Bierzynski, Stefano Traferro, Preetha Vijayan, Amirreza Yousefzadeh, Manolis Sifalakis, Rene Van Leuken, Dylan Muir, Rashid Ali Maen Mallah, Bijoy Kundu, Loreto Mateu, and Mario Diaz Nava*

3.1	Introduction . . . . .	90
3.1.1	Edge Computing Benefices and Challenges . . . . .	90
3.1.2	Artificial Neural Networks (ANNs) and Spiking Neural Networks (SNNs) . . . . .	92
3.2	State-of-the-art of key aspects of Neural Networks . . . . .	94
3.2.1	ANN and SNN Hardware Aware Design . . . . .	94
3.2.2	Sparsity . . . . .	95
3.2.3	ANN-to-SNN Conversion . . . . .	96
3.2.4	Surrogate Gradient Descent . . . . .	97
3.2.5	Neural Engineering Object (Nengo) Simulator . . .	97
3.3	NN Transformation: Temporal Delta Layer . . . . .	100
3.3.1	Temporal Delta Layer: Training Towards Brain Inspired Temporal Sparsity for Energy Efficient Deep Neural Networks . . . . .	100
3.3.2	Related Works . . . . .	102
3.3.3	Methodology . . . . .	103
3.3.3.1	Delta inference . . . . .	104
3.3.3.2	Activation quantization to induce sparsity .	105
3.3.3.3	Fixed point quantization . . . . .	105
3.3.3.4	Learned step-size quantization . . . . .	107
3.3.3.5	Sparsity penalty . . . . .	109
3.3.3.6	Proposed algorithms . . . . .	110
3.3.4	Experiments and Results . . . . .	110
3.3.4.1	Baseline . . . . .	110
3.3.4.2	Experiments . . . . .	110
3.3.4.3	Accuracy v/s Activation sparsity . . . . .	113
3.4	NN Compiler for Dedicated Inference Accelerator Hardware	115

3.4.1	Compiler Components . . . . .	116
3.4.2	ONNX Parser . . . . .	117
3.4.3	Hardware Architecture Representation . . . . .	118
3.4.4	Mapper . . . . .	119
3.4.5	Mapping Strategy . . . . .	120
3.4.6	Mapping of Deep Spiking NN Architectures to Digital SNN Inference Devices . . . . .	121
3.5	Simulator/Profiler . . . . .	123
3.6	Conclusions . . . . .	127
3.6.1	On NN Model Transformation . . . . .	127
3.6.2	On NN Compiler for Dedicated Inference Accelerator Hardware with Analog In-Memory Computing Conclusion . . . . .	128
3.6.3	Simulator/Profiler . . . . .	128
	References . . . . .	129

**4 Using FeFETs as Resistive Synapses in Crossbar-based Analog MAC Accelerating Units 137**

*Lei Zhang, David Borggreve, Frank Vanselow, Ralf Brederlow*

4.1	Introduction and Background . . . . .	138
4.2	Requirements of Crossbar Structure on eNVMs . . . . .	139
4.3	Synapse Design . . . . .	144
4.3.1	Conventional Design . . . . .	144
4.3.2	Gate-Cascaded FeFETs . . . . .	146
4.3.3	Exploration Results . . . . .	148
4.4	Conclusion . . . . .	150
	References . . . . .	151

**5 Emerging In-memory Computing for Neural Networks 153**

*Nellie Laleni, Taha Soliman, Alptekin Vardar, and Thomas Kämpfe*

5.1	Memory Technologies . . . . .	154
5.2	In-Memory Architectures . . . . .	154
5.2.1	Volatile Memories . . . . .	154
5.2.2	Non Volatile Memories . . . . .	155
5.2.3	Computational Domain . . . . .	156
5.2.3.1	Mixed signal approach . . . . .	156
5.2.3.2	Digital approach . . . . .	159
5.2.4	Target Network Quantization . . . . .	160

5.2.4.1	Floating point architectures . . . . .	160
5.2.4.2	Fixed-point architectures . . . . .	161
5.2.4.3	Binarized architectures . . . . .	161
5.2.4.4	Flexible precision architectures . . . . .	161
	References . . . . .	162
<b>6</b>	<b>Artificial Intelligence Advancements for Digitising Industry</b>	<b>167</b>
	<i>Ovidiu Vermesan, and Reiner John</i>	
6.1	AI at the Edge in Industrial Processes . . . . .	168
6.2	A pan-European AI Framework for Manufacturing and Process Technology . . . . .	170
6.3	AI Technologies . . . . .	180
6.4	AI Application Areas . . . . .	182
6.4.1	Automotive . . . . .	183
6.4.2	Semiconductor . . . . .	185
6.4.3	Industrial Machinery . . . . .	186
6.4.4	Food and Beverage . . . . .	188
6.4.5	Transportation . . . . .	190
6.5	AI Technology Roadmap for Digitising Industry . . . . .	191
6.6	Conclusion . . . . .	192
	References . . . . .	193
<b>7</b>	<b>Impact of AI and Digital Twins on IIoT</b>	<b>195</b>
	<i>Bin Han, Björn Richerzhagen, Hans Schotten, Davide Calandra, and Fabrizio Lamberti</i>	
7.1	Introduction to the Hexa-X Project . . . . .	195
7.2	An Ecosystem Concept for Digital Twins in IIoT . . . . .	196
7.3	Digital Twins for Emergent Intelligence . . . . .	197
7.4	Network-aware Digital Twins for Local Insight Generation .	200
7.5	AI at the Intersection between DTs and HMI in Industrial IoT . . . . .	201
7.6	Conclusion . . . . .	203
	References . . . . .	203
<b>8</b>	<b>Lesson Learnt and Future of AI Applied to Manufacturing</b>	<b>207</b>
	<i>Valerio Frascolla, Matthias Hummert, Tobias Monsees, Dirk Wübben, Armin Dekorsy, Nicola Michailow, Volkmar Döricht, Christoph Niedermeier, Joachim Kaiser, Arne Bröring, Michael Villnow, Daniel Wessel, Florian Geiser, Matthias Wissel,</i>	

*Alberto Viseras, Bin Han, Björn Richerzhagen, Hans Schotten, Davide Calandra, and Fabrizio Lamberti*

8.1	Introduction . . . . .	208
8.2	IoT Enabled by Machine Learning . . . . .	209
8.3	Machine Learning at the Edge . . . . .	210
8.3.1	Applications of <i>EdgeML</i> in Industrial IoT . . . . .	211
8.3.2	Challenges in <i>EdgeML</i> . . . . .	212
8.4	Federated Learning – A Solution to Train ML Models . . . . .	213
8.4.1	Applications for Federated Learning in Industrial IoT . . . . .	214
8.4.2	Federated Learning Scenarios . . . . .	215
8.4.3	Challenges in Federated Learning . . . . .	217
8.4.4	Frameworks and products for leveraging Federated Learning . . . . .	218
8.4.5	Reducing Complexity of RX Processing . . . . .	220
8.4.6	Enhancing Reliability by Multi-Connectivity in the Uplink . . . . .	223
8.5	Communications in an “Embodied Artificial Intelligence” Future . . . . .	227
8.6	Embodied Artificial Intelligence . . . . .	228
8.7	High Integration as a Central Technological Driver . . . . .	230
8.8	Conclusion . . . . .	233
	References . . . . .	233

**9 Ethical Considerations and Trustworthy Industrial AI Systems 241**

*Ovidiu Vermesan, Cristina De Luca, Reiner John, Marcello Coppola, Björn Debaillie, and Giulio Urlini*

9.1	Introduction . . . . .	242
9.2	Ethics and Responsible AI in Industrial Environments . . . . .	245
9.3	Requirements for Industry-Grade AI . . . . .	246
9.4	Industrial AI Challenges . . . . .	250
9.4.1	Complexity . . . . .	251
9.4.2	Use of Natural Resources . . . . .	251
9.4.3	Pollution and Waste . . . . .	252
9.4.4	Energy . . . . .	252
9.5	Ethical Considerations for Digitising Industry . . . . .	253
9.5.1	AI Trustworthiness . . . . .	253
9.5.2	Bias and Fairness . . . . .	254
9.5.3	Transparency . . . . .	255

9.5.4	Accountability . . . . .	255
9.5.5	Explainability . . . . .	256
9.5.6	Control . . . . .	257
9.5.7	Human-Machine Interaction and Manipulation of Behaviour . . . . .	257
9.5.8	Autonomous Industrial Systems . . . . .	258
9.5.9	Machine Ethics . . . . .	260
9.5.10	Automation and Employment . . . . .	260
9.6	AI and the Future Digitising Industry . . . . .	261
9.7	Ethical Guidelines for AI in Industrial Environments . . . . .	262
9.8	Recommendations for Ethical AI in Industrial Environments . . . . .	262
9.9	Conclusion . . . . .	265
	References . . . . .	266

**10 Current Challenges of AI Standardisation in the Digitising Industry 271**

*Ovidiu Vermesan, Marcello Coppola, Reiner John,  
Cristina De Luca, Roy Bahr, and Giulio Urlini*

10.1	Introduction . . . . .	272
10.2	International Principles . . . . .	273
10.3	Role of AI Standardisation in Digitising Industry . . . . .	274
10.4	Challenges Associated with AI Deployments in Industrial Environments . . . . .	275
10.5	AI Standardisation Needs in Industrial Automation . . . . .	276
10.6	Standardisation of Security and Safety in AI Systems . . . . .	278
10.7	The Global AI Standards Landscape and Standardisation Activities . . . . .	280
10.7.1	CEN-CENELEC . . . . .	282
10.7.2	ETSI . . . . .	282
10.7.3	IEC . . . . .	283
10.7.4	ISO . . . . .	284
10.7.5	IEEE . . . . .	288
10.7.6	IETF . . . . .	289
10.7.7	ITU-T . . . . .	290
10.8	AI Certification . . . . .	291
10.9	Recommendations for an AI Standardisation Roadmap . . . . .	293
10.10	Conclusion . . . . .	296
	References . . . . .	297

xiv *Contents*

**Index** 301

**About the Editors** 307

---

# 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 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

---

<b>Figure 1.1</b>	AI systems capabilities. . . . .	9
<b>Figure 1.2</b>	Narrow AI vs General AI. . . . .	12
<b>Figure 1.3</b>	AI problem solving domains. . . . .	15
<b>Figure 1.4</b>	Typical expert system architecture. . . . .	15
<b>Figure 1.5</b>	Typical CNN-based machine (left) and workflow (right). . . . .	18
<b>Figure 1.6</b>	Self-driving vehicles: Training and inference (generate steering commands). . . . .	19
<b>Figure 1.7</b>	Life's principles. . . . .	21
<b>Figure 1.8</b>	Using Genetic Algorithms in the iterative process of fine-tuning NN hyperparameters. . . . .	24
<b>Figure 1.9</b>	Discriminative (left) vs Generative (right) Models in ML. . . . .	25
<b>Figure 1.10</b>	Network architecture of generator and discriminator based on deep convolutional GAN. . . . .	26
<b>Figure 1.11</b>	Swarm intelligence visualized: population of agents searching for a destination (left) and search space represented by a nonlinear regression generated surface (right). . . . .	28
<b>Figure 1.12</b>	Perceptron illustration. . . . .	31
<b>Figure 1.13</b>	Automated planning, states, and actions. . . . .	32
<b>Figure 1.14</b>	Application of Metaverse. . . . .	34
<b>Figure 1.15</b>	AI across the edge continuum. . . . .	36
<b>Figure 1.16</b>	Knowledge representation. . . . .	40
<b>Figure 1.17</b>	Type of knowledge. . . . .	41
<b>Figure 1.18</b>	Five-layer (with sublayers) AI technology stack. . . . .	43
<b>Figure 1.19</b>	ML taxonomy. . . . .	45
<b>Figure 1.20</b>	Regression visualized 2D (left), 3D (right). . . . .	45
<b>Figure 1.21</b>	Normal(blue) - abnormal(red) (left). Predicted values using logistic regression (right). . . . .	46
<b>Figure 1.22</b>	Classification (left) vs Regression (right) . . . . .	47

<b>Figure 1.23</b>	Cluster (Gaussian mix) 4 clusters (left) vs 2 clusters (right). . . . .	48
<b>Figure 1.24</b>	Principal component analysis. Intuitive visualisation, select variables that capture the largest variability in data. . . . .	48
<b>Figure 1.25</b>	Q-Learning vs Deep Q-Learning. . . . .	49
<b>Figure 1.26</b>	Typical CNN architecture. . . . .	51
<b>Figure 1.27</b>	The repeating module underlying RNN architecture. . . . .	52
<b>Figure 1.28</b>	Example of an architecture useful for fault diagnosis.	52
<b>Figure 1.29</b>	Embedded ML design and development ecosystem view. . . . .	55
<b>Figure 1.30</b>	Embedded ML optimisation. . . . .	58
<b>Figure 1.31</b>	ML hardware options for various AI tasks. . . . .	59
<b>Figure 1.32</b>	Comparison of the von Neumann architecture with the neuromorphic architecture. . . . .	61
<b>Figure 1.33</b>	The high-level embedded ML development flow. . . . .	62
<b>Figure 1.34</b>	Temporal and frequency plots as input to motor classification. . . . .	63
<b>Figure 1.35</b>	Hyperparameters (outside the model) vs parameters (inside the model). . . . .	64
<b>Figure 1.36</b>	Categories of datasets and where they are used. . . . .	64
<b>Figure 1.37</b>	Confusion matrix. . . . .	66
<b>Figure 2.1</b>	TEMPO spreads over three action areas. . . . .	76
<b>Figure 2.2</b>	TEMPO positioned in the greater landscape of neuromorphic computing. . . . .	80
<b>Figure 2.3</b>	Possible inputs for the western food classification DNN. . . . .	83
<b>Figure 2.4</b>	3D landscape, ordering of 3D technologies according to the system-level wiring hierarchy. . . . .	85
<b>Figure 3.1</b>	Networks to hardware workflow. . . . .	95
<b>Figure 3.2</b>	Spiking neuron models. . . . .	99
<b>Figure 3.3</b>	(a) Standard DNN and (b) DNN with temporal delta layer. . . . .	101
<b>Figure 3.4</b>	Sparsity in $\Delta x$ can save multiplications between $\Delta x$ and columns of $W$ that correspond to zero. . . . .	103
<b>Figure 3.5</b>	Demonstration of two consecutive activation maps leading to near zero deltas. . . . .	106

**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. . . . . 108

**Figure 3.7** Methodology flow of temporal delta layer with fixed point quantization. . . . . 111

**Figure 3.8** Methodology flow of temporal delta layer with learned step size quantization. . . . . 112

**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. . . . . 115

**Figure 3.10** Overview of Compiler Tool. . . . . 117

**Figure 3.11** ONNX Parser diagram of parsing and fusing the input ONNX model into a list of Nodes and Fused Nodes. . . . . 118

**Figure 3.12** Mapping flow of the Compiler. . . . . 119

**Figure 3.13** Mapping of layers 1 and 2 on processing core 1. . . 121

**Figure 3.14** A HW architecture for SNN inference. . . . . 122

**Figure 3.15** An example of a deep spiking network that will be mapped to a HW architecture. . . . . 123

**Figure 3.16** MobileNet V1 parameters per layer. . . . . 124

**Figure 3.17** MobileNet V1 data volume per layer, normalized to input data volume. . . . . 124

**Figure 3.18** MobileNet V1 bandwidth (Gb/s) at each layer. . . . 125

**Figure 3.19** N2D2: Neural Network Design & Deployment. . . 126

**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. . . . . 126

**Figure 4.1** (a) shows FeFETs' abstract structure, 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. . . . . 139

**Figure 4.2** Implementations of analogue MAC accelerating units using single-ended (a) and pseudo-differential (b) structures are shown. . . . . 140

**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. . . . . 143

**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 scarfying the number of available states. . . . . 144

**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. . . . . 146

**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. . . . . 149

**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. . . . . 150

<b>Figure 5.1</b>	Conventional volatile memory cells a) 6T SRAM cell and b) DRAM cell. . . . .	154
<b>Figure 5.2</b>	General concept of mixed-signal in-memory cross-bar (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. . . . .	157
<b>Figure 5.3</b>	Eliminated the DACs and instead serialize the activation by applying only a single bit at each cycle. . . . .	158
<b>Figure 5.4</b>	(a) Several row activation approach such as Ambit’s TRA. (b) Changing subarray unit cell structure whether with extra transistors or operation mode as in DRISA 3T1C. (c) Activating only one row at a time and use the row activation as an operand as in FlexPim. . . . .	159
<b>Figure 5.5</b>	The relation between the energy cost for digital and analog MAC operations versus bit precision. . . . .	160
<b>Figure 6.1</b>	AI4DI Objectives. . . . .	172
<b>Figure 6.2</b>	AI4DI Key Targets. . . . .	173
<b>Figure 6.3</b>	Silicon-born AI effect on Moore’s Law beyond the current silicon technology developments. . . . .	180
<b>Figure 7.1</b>	The ecosystem of 6G human-centric industrial DTs, with the arrows indicating the direction of the information flow. . . . .	197
<b>Figure 7.2</b>	Comparing the conventional AI solutions based on centralized AI (left) and FL (middle) to EI (right). . . . .	198
<b>Figure 7.3</b>	Illustration of collaborating DTs in IIoT. . . . .	201
<b>Figure 8.1</b>	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. . . . .	213
<b>Figure 8.2</b>	Visualization of the FL process. The four steps are executed consecutively and are repeated following the same process until the global model converges. . . . .	214

<b>Figure 8.3</b>	FL scenarios according to how the data is split across clients. (a) Horizontal FL. (b) Vertical FL. . . . .	216
<b>Figure 8.4</b>	Efficiency $\eta$ over SNR for standard ARQ scheme in comparison to E-ARQ with NN-FoC forecasting and a Genie forecaster for different decoder delays $\kappa$ . . . . .	222
<b>Figure 8.5</b>	Distributed communication system with $\mathbf{J}$ access points forwarding compressed messages to the DU. . . . .	224
<b>Figure 8.6</b>	BER performance for 16-QAM with RAPs applying SNR-adapted 6-bit quantizer per AP and REMC in DU for $\mathbf{J} \geq 1$ . . . . .	226
<b>Figure 8.7</b>	The cognitive cycle of an embodied intelligence agent. . . . .	229
<b>Figure 8.8</b>	Overview of mmW frequencies. 5G bands expand up to 50 GHz, 6G is expected to reach 1 THz and also include visible light communications. . . . .	231
<b>Figure 8.9</b>	Overview of the functions of mmW wireless technology. . . . .	232
<b>Figure 9.1</b>	A framework for trustworthy industrial AI systems. . . . .	245
<b>Figure 9.2</b>	Complexity of applicability of ethical considerations resulting from the interaction of subsystems. . . . .	247
<b>Figure 9.3</b>	Requirements for industry-grade AI. . . . .	247
<b>Figure 9.4</b>	Reference architecture for AI-based autonomous systems in industrial environments. . . . .	259
<b>Figure 10.1</b>	NIST focus areas for standards development. . . . .	273
<b>Figure 10.2</b>	Three-layer AI topics structure: generic, horizontal, and relevant industrial application areas. . . . .	277
<b>Figure 10.3</b>	Industrial AI standards system framework. . . . .	281
<b>Figure 10.4</b>	Classification scheme along with criticality, AI methods and capabilities. . . . .	293

---

## List of Tables

---

<b>Table 2.1</b>	Edge AI use cases addresses in TEMPO covers five application domains . . . . .	82
<b>Table 3.1</b>	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. . . . .	113
<b>Table 3.2</b>	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. . . . .	114
<b>Table 3.3</b>	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. . . . .	114
<b>Table 3.4</b>	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 . . . . .	127
<b>Table 4.1</b>	Comparison between single-ended and pseudo-differential structures . . . . .	144
<b>Table 4.2</b>	Relative Performance Comparison . . . . .	148





---

## 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*
- Dörich, 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*
- 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*

**Nava, Mario Diaz**, *STMicroelectronics, France*

**Niedermeier, Christoph**, *Siemens AG, Germany*

**Ocket, Ilja**, *imec, Belgium*

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

**Richerzhagen, Björn**, *Siemens Technology, 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

