Distributed Machine Learning in Edge Computing

Subjects: Computer Science, Artificial Intelligence

Contributor: Carlos Poncinelli Filho , Elias Marques Junior , Victor Chang , Leonardo dos Santos , Flavia Bernardini , Paulo F. Pires , Luiz Satoru Ochi , Flavia Coimbra Delicato

Distributed edge intelligence is a disruptive research area that enables the execution of machine learning and deep learning (ML/DL) algorithms close to where data are generated. Since edge devices are more limited and heterogeneous than typical cloud devices, many hindrances have to be overcome to fully extract the potential benefits of such an approach (such as data-in-motion analytics).

machine learning | artificial intelligence | distributed | edge intelligence | fog intelligence | Internet of Things

1. Introduction

Nowadays, with the rise of the Internet of Things (IoT), a large number of smart applications are being built, taking advantage of connecting several types of devices to the internet. These applications will generate a massive amount of data that need to be processed promptly to generate valuable and actionable information. Edge intelligence (EI) refers to the ability to bring about the execution of machine learning tasks from the remote cloud closer to the IoT/Edge devices, either partially or entirely. Examples of edge devices are smartphones, access points, gateways, smart routers and switches, new generation base stations, and micro data centers.

Some edge devices have considerable computing capabilities (although always much smaller than cloud processing centers), but most are characterized by very limited capabilities. Currently, with the increasing development in the area of MEMS (Micro–Electro–Mechanical Systems) devices, there is a tendency to carry out part of the processing within the data producing devices themselves (sensors) [1][2][3][4]. There are certainly several challenges involved in performing processing on resource-limited devices, including the need to adapt complex algorithms and divide the processing among several nodes.

Therefore, in Edge Intelligence, it is essential to promote collaboration between devices to compensate for their lower computing capacity. Some synonyms of this concept found in the literature are: distributed learning, edge/fog learning, distributed intelligence, edge/fog intelligence and mobile intelligence [5][6][7].

The leverage of edge intelligence reduces some drawbacks of running ML tasks entirely in the cloud, such as:

- High latency [8]: offloading intelligence tasks to the edge enables achievement of faster inference, decreasing the inherent delay in data transmission through the network backbone;
- Security and privacy issues [9][10]: it is possible to train and infer on sensitive data fully at the edge, preventing their risky propagation throughout the network, where they are susceptible to attacks. Moreover, edge intelligence can derive non-sensitive information that could then be submitted to the cloud without further processing;
- The need for continuous internet connection: in locations where connectivity is poor or intermittent, the ML/DL could still be carried out;
- Bandwidth degradation: edge computing can perform part of processing tasks on raw data and transmit the produced data to the cloud (filtered/aggregated/pre-processed), thus saving network bandwidth. Transmitting large amounts of data to the cloud burdens the network and impacts the overall Quality of Service (QoS) [11];
- Power waste [12]: unnecessary raw data being transmitted through the internet demands power, decreasing energy efficiency on a large scale.

The steps for data processing in ML vary according to the specific technique in use, but generally occur in a well-defined life cycle, which can be represented by a workflow. Model building is at the heart of any ML technique, but the complete life cycle of a learning process involves a series of steps, from data acquisition and preparation to model deployment into a production environment. When adopting the Edge intelligence paradigm, it is necessary to carefully analyze which steps in the ML life cycle can be successfully executed at the edge of the network. Typical steps that have been investigated for execution at the edge are data collection, pre-processing, training and inference.

2. Related Work

Some surveys have been published that address the edge intelligence subject recently. However, they adopt different perspectives from the one adopted in this SLR. Al-Rakhami et al. [13] propose and analyze a framework based on the distributed edge/cloud paradigm using docker technology which provides a very lightweight and effective virtualization solution. This solution can be utilized to manage, deploy and distribute applications onto clusters (e.g., small board devices such as Raspberry PI). It is able to provide an advantageous combination of various benefits and lower costs of data processing performed at the edge instead of central servers. However, the authors base their proposal on experiments to support the proposal of a new framework. The research does not mention any of the nine groups of techniques the researchers present in the work.

Wang et al. [14] survey is centered on the connection between Deep Learning and the edge, either to apply DL in optimizing the edge or to use the edge to run DL algorithms. The study is divided into five fronts: DL applications on edge; DL inference in edge; edge computing for DL; DL training at the edge; DL for optimizing the edge. The paper discusses hardware and virtualization aspects. Concerning the (groups of) techniques and strategies, it is more restricted to Federated Learning and the optimization of the edge with DL.

Xu et al. [10] approach edge intelligence under the perspectives of edge caching, edge training, edge inference, and edge offloading in a very comprehensive way. The researchers discuss all these aspects in the work but explore additional techniques, and strategies related to pre-processing, federated learning, and scheduling. One intersection of this paper with the researchers' research is the overlap of three groups of techniques the researchers present (Federated Learning, Edge Pre-processing and Scheduling). However, the researchers deepened the discussion into more groups of techniques.

The work presented by Zhou et al. [15] covers artificial intelligence to edge AI, showing a generalized representation of application architecture used in the lifecycle management of ML. In the edge layer: sensors/actuators; edge analytics; logging and monitoring. In the fog layer: visualization; live streaming engines; batch processing; data ingestion; storage and ML model development platforms and libraries. The researchers' research approaches several more domains in which edge intelligence is used, which are not present in this survey. Compared to these other surveys, the researchers analyze the literature more comprehensively, including a discussion on application domains of edge intelligence and their correlation with identified techniques.

Verbraeken et al. [16] provide an extensive overview of the current state-of-the-art in terms of outlining the challenges and opportunities of distributed machine learning over conventional machine learning, discussing the techniques used for distributed machine learning. The paper follows the same line of research of Wang et al. [14], with a focus on machine learning applied to the distributed environment. To this end, it makes inroads into the various types of algorithms to solve problems using ML.

Table 1 shows the comparison between the researchers' work and the other surveys mentioned in this section. In summary, the main gaps of the analyzed works are focused on aspects such as “Techniques and Strategies” on the edge. The table also shows the aspects of “Challenges”and “Different Application Domains”, where edge intelligence can be used.

Table 1. Comparison of existing surveys.

	Scope		
Paper	Challenges	Group of Techniques	Different Application Domains
Al-Rakhami et al. [13]	0/6	2/8	1/6
Wang et al. [14]	1/6	4/8	4/6
Verbraeken et al. [16]	1/6	0/8	0/6

Paper	Challenges	Scope	
		Group of Techniques	Different Application Domains
Zhou et al. [15]	2/6	4/8	0/6
Dianlei Xu et al. [10]	6/6	3/8	0/6
The researchers' work	6/6	8/8	6/6

3. Answering the RQs

3.1. RQ1—Research Challenges in Edge Intelligence (EI)

In this section, the researchers summarize the challenges faced by the Edge Intelligence (EI) paradigm that the analyzed studies either mentioned or aimed to tackle. The discussion presented in this section aims to provide answers to RQ1: What are the main challenges and open issues in the distributed learning field?

As mentioned earlier, performing ML techniques at the edge of the network promises to bring several benefits, but it raises several challenges. As this field is still in its beginning, solutions to each challenge are still being investigated. The surveyed studies tackle several challenges, which can be broadly grouped into six categories, displayed in **Table 2** and described in what follows.

Table 2. Challenges in distributed machine learning in edge computing.

	Challenges
CH1	Running ML/DL on devices with limited resources
CH2	Ensuring energy efficiency without compromising the accuracy
CH3	Communication efficiency

https://www.semi.org/en/blogs/technology-trends/ai-and-mems-sensors (accessed on 1 March 2022).
2. Dion, G.; Mejaouri, S.; Sylvestre, J. Reservoir computing with a single delay-coupled non-linear mechanical oscillator. J. Appl. Phys. 2018, 124, 152132.
3. Rafaie, M.; Hasan, M.H.; Alsaleem, F.M. Neuromorphic MEMS sensor network. Appl. Phys. Lett. 2019, 114, 163501.
4. Hasan, M.H.; Al-Ramini, A.; Abdel-Rahman, E.; Jafari, R.; Alsaleem, F. Colocalized Sensing and Intelligent Computing in Micro-Sensors. Sensors 2020, 20, 6346.
5. Wang, X.; Han, Y.; Wang, C.; Zhao, Q.; Chen, X.; Chen, M. In-edge ai: Intelligentizing mobile edge computing, caching and communication by federated learning. IEEE Netw. 2019, 33, 156–165.
6. Li, E.; Zhou, Z.; Chen, X. Edge intelligence: On-demand deep learning model co-inference with device-edge synergy. In Proceedings of the 2018 Workshop on Mobile Edge Communications, Budapest, Hungary, 20 August 2018; pp. 31–36.

Challenges	
CH4	Ensuring data privacy and security
CH5	Handling failure in edge devices
CH6	Heterogeneity and low quality of data

… have little processing capacity, mainly when compared to the powerful data centers at the cloud. On the other hand, many ML applications require high computational power that outweighs the possibilities of resource-constrained IoT and edge devices. Limited resources also include memory and storage capacities. NN and ML algorithms generally require storing of and access to a handful of parameters that describe the model architecture and weight values forming the classification model. With limited storage, it may not be possible to have continued access to the original training data, or the data may have been removed altogether to free up space. Therefore, a significant challenge is reducing memory access and storing the data locally to avoid costly reading and writing to external memory modules.

CH2 consists of ensuring the energy efficiency of edge devices without compromising the accuracy of the system. In general, the higher the complexity of the required processing, the more energy is consumed. Edge devices can be battery-powered. In these cases, the energy consumption of algorithms must be minimized to reach energy efficiency. However, this should be done with care so as not to compromise the quality of the data generated and the decisions/inferences made. So, there is an important trade-off to be managed.

CH3 concerns communication issues, where edge intelligence models must consider that the devices might face poor connectivity. In such cases, the model update time in training tasks may be delayed. Valerio, Passarella and Conti [17] claim that the inference is highly sensitive to the available bandwidth in communication. Challenges in communication include network traffic, fluctuations in the bandwidth, intermittent or unavailable connectivity.

CH4 is related to data privacy and security. Several applications in edge intelligence handle sensitive data, such as healthcare. Thus, distributed ML algorithms must be able to preserve user privacy and information security when data are transferred throughout the devices. Distributed Edge-Intelligence (EI) has multiple points of vulnerability to possible malicious attacks or leakage of confidential or important data in the ML workflow.

CH5 is the challenge posed by failures in edge devices. Since devices might fail at some point, the distributed algorithm must consider ways to overcome this situation. Lastly, heterogeneity and lack of quality in available data rise challenge CH6. For most ML algorithms, especially in supervised machine learning, high accuracy depends on the high quality of training data. However, this often does not apply in edge intelligence scenarios, where the collected data are sparse and unlabelled [10]. Distributed edge intelligence can handle data from different sources

11. … mance … tional … 2407–
12. … service … 2019, 15, 4216–4224.
13. Al-Rakhami, M.; Alsahli, M.; Hassan, M.M.; Alamri, A.; Guerrieri, A.; Fortino, G. Cost efficient edge intelligence framework using docker containers. In Proceedings of the 2018 IEEE 16th International Conference on Dependable, Autonomic and Secure Computing Congress (DASC/PiCom/DataCom/CyberSciTech), Athens, Greece, 12–15 August 2018; pp. 800–807.
14. Wang, X.; Han, Y.; Leung, V.C.; Niyato, D.; Yan, X.; Chen, X. Convergence of edge computing and deep learning: A comprehensive survey. IEEE Commun. Surv. Tutorials 2020, 22, 869–904.
15. Zhou, Z.; Chen, X.; Li, E.; Zeng, L.; Luo, K.; Zhang, J. Edge intelligence: Paving the last mile of artificial intelligence with edge computing. Proc. IEEE 2019, 107, 1738–1762.
16. Verbraeken, J.; Wolting, M.; Katzy, J.; Kloppenburg, J.; Verbelen, T.; Rellermeyer, J.S. A survey on distributed machine learning. ACM Comput. Surv. (CSUR) 2020, 53, 1–33.
17. Valerio, L.; Passarella, A.; Conti, M. Accuracy vs. traffic trade-off of learning iot data patterns at the edge with hypothesis transfer learning. In Proceedings of the 2016 IEEE 2nd International Forum on Research and Technologies for Society and Industry Leveraging a better tomorrow (RTSI), Bologna, Italy, 7–9 September 2016; pp. 1–6.
18. Zou, Z.; Jin, Y.; Nevalainen, P.; Huan, Y.; Heikkonen, J.; Westerlund, T. Edge and fog computing enabled AI for IoT-an overview. In Proceedings of the 2019 IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS), Hsinchu, Taiwan, 18–20 March 2019; pp. 51–56.
19. Hossain, M.S.; Muhammad, G.; Amin, S.U. Improving consumer satisfaction in smart cities using edge computing and caching: A case study of date fruits classification. Future Gener. Comput. Syst. 2018, 88, 333–341.
20. Sharma, A.; Sabitha, A.S.; Bansal, A. Edge analytics for building automation systems: A review. In Proceedings of the 2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN), Greater Noida, India, 12–13 October 2018; pp. 585–590.
21. Wan, S.; Gu, Z.; Ni, Q. Cognitive computing and wireless communications on the edge for healthcare service robots. Comput. Commun. 2020, 149, 99–106.

22. Elbamby, M.S.; Perfecto, C.; Liu, C.F.; Park, J.; Samarakoon, S.; Chen, X.; Bennis, M. Wireless edge computing with latency and reliability guarantees. Proc. IEEE 2019, 107, 1717–1737.

23. Rausch, T.; Dustdar, S. Edge intelligence: The convergence of humans, things, and ai. In Proceedings of the 2019 IEEE International Conference on Cloud Engineering (IC2E), Prague, Czech Republic, 24–27 June 2019; pp. 86–96.

24. Fasciano, C.; Vitabile, F. Artificial Intelligence on Edge Computing: A Healthcare Scenario in Ambient Assisted Living. In Proceedings of the Artificial Intelligence for Ambient Assisted Living (AI*AAL.it 2019), Rende, Italy, 20–23 November 2019.

used in distributed ML applications. The table also correlates each framework with the

Table 3. References to the challenges of Edge Intelligence.

	References	Works That Tackle the Challenges
CH1	[10][15][18][19][20][21][22][23][24][25][26][27]	[14][24][27][28][29][30][31][32][33][34][35][36][37][38][39][40][41][42][43][44][45][46][47][48][49][50][51][52][53][54][55][56][57][58][59][60][61][62]
CH2	[10][15][19][22][24][26][37]	[8][20][24][27][29][32][33][34][45][48][52][55][57][61][62][63]
CH3	[10][20][24][25][42][62]	[16][17][19][29][30][31][32][35][39][42][52][64][65]
CH4	[10][20][23][32][40][66]	[8][9][10][20][40][47][67][68][69]
CH5	[10][23]	–
CH6	[10][20][40][70][71]	[10][34][66]

inference via edge computing. IEEE Trans. Wirel. Commun. 2019, 19, 447–457.

31. Li, H.; Ota, K.; Dong, M. Learning IoT in edge: Deep learning for the Internet of Things with edge computing. IEEE Netw. 2018, 32, 96–101.

32. Hassan, M.A.; Xiao, M.; Wei, Q.; Chen, S. Help your mobile applications with fog computing. In Proceedings of the 2015 12th Annual IEEE International Conference on Sensing, Communication, and Networking-Workshops (SECON Workshops), Seattle, WA, USA, 22–25 June 2015; pp. 1–6.

33. Liu, C.; Cao, Y.; Luo, Y.; Chen, G.; Vokkarane, V.; Yunsheng, M.; Chen, S.; Hou, P. A new deep learning-based food recognition system for dietary assessment on an edge computing service infrastructure. IEEE Trans. Serv. Comput. 2017, 11, 249–261.

34. Liu, Y.; Yang, C.; Jiang, L.; Xie, S.; Zhang, Y. Intelligent edge computing for IoT-based energy management in smart cities. IEEE Netw. 2019, 33, 111–117.

3.2. RQ2—Techniques and Strategies

Here, the researchers focus on three main aspects, namely: (i) the system architecture, (ii) how the ML tasks are distributed among the devices, and (iii) the underlying adopted techniques. The researchers classify the several approaches used in distributed learning based on these three aspects. The researchers identified nine groups of techniques and strategies, described in what follows: Federated learning; Model partitioning; Right-sizing; Edge pre-processing; Scheduling; Cloud pre-training; Edge only; Model Compression; and Other techniques.

3.3. RQ3—Frameworks for Edge Intelligence

This section describes the studies that provided answers to the RQ3 of this survey. **Table 4** lists the main frameworks currently used in distributed ML applications. The table also correlates each framework with the

35. Nishio, T.; Yonetani, R. Client selection for federated learning with heterogeneous resources in mobile edge. In Proceedings of the ICC 2019-2019 IEEE International Conference on Communications (ICC), Shanghai, China, 20–24 May 2019; pp. 1–7.

Corresponding Techniques or Strategies of E frameworks.

Framework	Groups of Techniques or Strategies	Comments
Neurosurgeon [72]	Model Partitioning	Lightweight scheduler to automatically partition DNN computation between edge devices and cloud at the granularity of NN layers
JointDNN [73]	Model Partitioning	JointDNN provides an energy- and performance-efficient method of querying some layers on the mobile device and some layers on the cloud server.
H. Li et al. [31]	Model Partitioning	They divide the NN layers and deploy the part with the lower ones (closer to the input) into edge servers and the part with higher layers (closer to the output) into the cloud for offloading processing. They also propose an offline and an online algorithm that schedules tasks in Edge servers.
Musical chair [74]	Model Partitioning	Musical Chair aims at alleviating the compute cost and overcoming the resource barrier by distributing their computation: data parallelism and model parallelism.
AAIoT [75]	Model Partitioning	Accurate segmenting NNs under multi-layer IoT architectures
MobileNet [42]	Model Compression Model Selector	Presented by Google Inc., the two hyperparameters introduced allow the model builder to choose the right sized model for the specific application.

46. Guo, R.; Xiang, Y.; Mao, Z.; Yi, Z.; Zhao, X.; Shi, D. Artificial Intelligence Enabled Online Non-intrusive Load Monitoring Embedded in Smart Plugs. In Proceedings of the International Symposium on Signal Processing and Intelligent Recognition Systems, Trivandrum, India, 18–21 December 2019; Springer: Singapore, 2019; pp. 23–36.

Framework	Groups of Techniques or Strategies	Comments
Squeezenet	Model Compression	It is a reduced DNN that achieves AlexNet-level accuracy with 50 times fewer parameters
Tiny-YOLO	Model Compression	Tiny Yolo is a very lite NN and is hence suitable for running on edge devices. It has an accuracy that is comparable to the standard AlexNet for small class numbers but is much faster.
BranchyNet	Right sizing	Open source DNN training framework that supports the early-exit mechanism.
TeamNet [76]	Model Compression Transfer Learning	TeamNet trains shallower models using the similar but downsized architecture of a given SOTA (state of the art) deep model. The master node compares its uncertainty with the worker's and selects the one with the least uncertainty as to the final result.
OpenEI [42]	Model Compression Data Quantization Model Selector	The algorithms are optimized by compressing the size of the model, quantizing the weight. The model selector will choose the most suitable model based on the developer's requirement (the default is accuracy) and the current computing resource.
TensorFlow Lite [77]	Data Quantization	TensorFlow's lightweight solution, which is designed for mobile and edge devices. It leverages many optimization techniques, including quantized kernels, to reduce the latency.

57. Gonzalez-Guerrero, P.; Tracy II, T.; Guo, X.; Stan, M.R. Towards low-power random forest using asynchronous computing with streams. In Proceedings of the 2019 Tenth International Green and Sustainable Computing Conference (IGSC), Alexandria, VA, USA, 21–24 October 2019; pp. 1–5.
58. Sanchez, J.; Soltani, N.; Chamarthi, R.; Sawant, A.; Tabkhi, H. A novel 1d-convolution accelerator for low-power real-time cnn processing on the edge. In Proceedings of the 2018 IEEE High

Framework	Groups of Techniques or Strategies	Comments
QNNPACK (Quantized Neural Networks PACKage) [78]	Data Quantization	Developed by Facebook, is a mobile-optimized library for high-performance NN inference. It provides an implementation of common NN operators on quantized 8-bit tensors.
ProtoNN [79]	Model Compression	Inspired by k-Nearest Neighbor (KNN) and could be deployed on the edges with limited storage and computational power.
EMI-RNN [80]	Right Sizing	It requires 72 times less computation than standard Long Short term Memory Networks (LSTM) and improves its accuracy by 1%.
CoreML [81]	Model Compression Data Quantization	Published by Apple, it is a deep learning package optimized for on-device performance to minimize memory footprint and power consumption. Users are allowed to integrate the trained machine learning model into Apple products, such as Siri, Camera, and QuickType.
DroNet [33]	Model Compression Data Quantization	The DroNet topology was inspired by residual networks and was reduced in size to minimize the bare image processing time (inference). The numerical representation of weights and activations reduces from the native one, 32-bit floating-point (Float32), down to a 16-bit fixed point one (Fixed16).
Stratum [82]	Model Selector Dynamic Scheduling	Stratum can select the best model by evaluating a series of user-built models. A resource monitoring framework within Stratum keeps track of resource utilization and is responsible for triggering actions to elastically scale resources and migrate tasks, as needed, to meet the ML workflow's Quality of Services (QoS). ML modules can be placed on the edge of the Cloud layer, depending on user requirements and capacity analysis.

3681–3692.

70. Zhou, Z.; Liao, H.; Gu, B.; Huq, K.M.S.; Mumtaz, S.; Rodriguez, J. Robust mobile crowd sensing: When deep learning meets edge computing. IEEE Netw. 2018, 32, 54–60.

Framework	Groups of Techniques or Strategies	Comments
Efficient distributed deep learning (EDDL) [53]	Model Compression Model Partitioning Right-Sizing	A systematic and structured scheme based on balanced incomplete block design (BIBD) used in situations where the dataflows in DNNs are sparse. Vertical and horizontal model partition and grouped convolution techniques are used to reduce computation and memory. To speed up the inference, BranchyNet is utilized.
In-Edge AI [5]	Federated Learning	Utilizes the collaboration among devices and edge nodes to exchange the learning parameters for better training and inference of the models.
Edgence [83]	Blockchain	Edgence (EDGe + intelligENCE) is proposed to serve as a blockchain-enabled edge-computing platform to intelligently manage massive decentralized applications in IoT use cases.
FederatedAveraging (FedAvg) [84]	Federated Learning	Combines local stochastic gradient descent (SGD) on each client with a server that performs model averaging.
SSGD [85]	Federated Learning	System that enables multiple parties to jointly learn an accurate neural network model for a given objective without sharing their input datasets.
BlockFL [86]	Blockchain Federated Learning	Mobile devices' local model updates are exchanged and verified by leveraging blockchain.
Edgent [6]	Model Partitioning	Adaptively partitions DNN computation between the device and edge, in order to leverage hybrid computation resources in

83. Xu, J.; Wang, S.; Zhou, A.; Yang, F. Edgence: A blockchain-enabled edge-computing platform for intelligent IoT-based dApps. China Commun. 2020, 17, 78–87.

Framework	Groups of Techniques or Strategies	Comments
	Right-Sizing	proximity for real-time DNN inference. DNN right-sizing accelerates DNN inference through the early exit at a proper intermediate DNN layer to further reduce the computation latency.
PipeDream [87]	Model Partitioning	PipeDream keeps all available GPUs productive by systematically partitioning DNN layers among them to balance work and minimize communication.
GoSGD [88]	Gossip Averaging	Method to share information between different threads based on gossip algorithms and showing good consensus convergence properties.
Gossiping SGD [89]	Gossip Averaging	Asynchronous method that replaces the all-reduce collective operation of synchronous training with a gossip aggregation algorithm.
GossipGraD [90]	Gossip Averaging	Asynchronous communication of gradients for further reducing the communication cost.
INCEPTIONN [91]	Data Quantization	Lossy-compression algorithm for floating-point gradients. The framework reduces the communication time by 70.9 80.7% and offers 2.2 3.1× speedup over the conventional training system while achieving the same level of accuracy.
Minerva [92]	Data Quantization Model compression	Quantization analysis minimizes bit widths without exceeding a strict prediction error bound. Compared to a 16-bit fixed-point baseline, Minerva reduces power consumption by 1.5×. Minerva identifies operands that are close to zero and removes them from the prediction computation such that model accuracy is not

Cloud Computing, Carlsbad, CA, USA, 11–13 October 2018; pp. 401–411.
95. Sun, W.; Liu, J.; Yue, Y. AI-enhanced offloading in edge computing: When machine learning meets industrial IoT. IEEE Netw. 2019, 33, 68–74.

Framework	Groups of Techniques or Strategies	Comments
		affected. Selective pruning further reduces power consumption by 2.0× on top of bit width quantization.
AdaDeep [93]	Model Compression	Automatically selects a combination of compression techniques for a given DNN that will lead to an optimal balance between user-specified performance goals and resource constraints. AdaDeep enables up to 9.8× latency reduction, 4.3× energy efficiency improvement, and 38× storage reduction in DNNs while incurring negligible accuracy loss.
JALAD [94]	Data Quantization Model Partitioning	Data compression by jointly considering compression rate and model accuracy. A latency-aware deep decoupling strategy to minimize the overall execution latency is employed. Decouples a deep NN to run a part of it at edge devices and the other part inside the conventional cloud.

Figure 2 illustrates this taxonomy up to a third level. **Table 5** shows the works that tackle these domains. **Figure 3** summarizes the statistics of the six domains of the publishing by field.

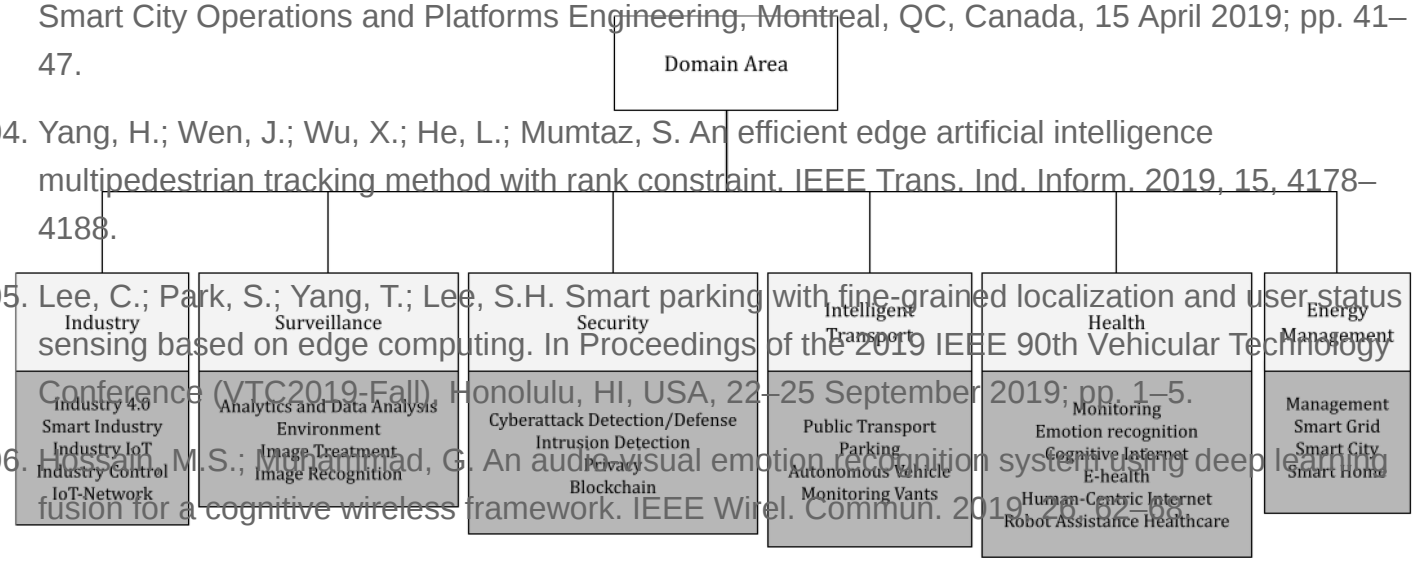

Figure 2. Application domains.

103. Liu, L.; Zhang, X.; Zhang, Q.; Weinert, A.; Wang, Y.; Shi, W. AutoVAPS: An IoT-enabled public safety service on vehicles. In Proceedings of the Fourth Workshop on International Science of Smart City Operations and Platforms Engineering, Montreal, QC, Canada, 15 April 2019; pp. 41–47.
104. Yang, H.; Wen, J.; Wu, X.; He, L.; Mumtaz, S. An efficient edge artificial intelligence multipedestrian tracking method with rank constraint. IEEE Trans. Ind. Inform. 2019, 15, 4178–4188.
105. Lee, C.; Park, S.; Yang, T.; Lee, S.H. Smart parking with fine-grained localization and user status sensing based on edge computing. In Proceedings of the 2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall), Honolulu, HI, USA, 22–25 September 2019; pp. 1–5.
106. Hossain, M.S.; Muhammad, G. An audio-visual emotion recognition system using deep learning fusion for a cognitive wireless framework. IEEE Wirel. Commun. 2019, 26, 62–68.
107. Cao, Y.; Hou, P.; Brown, D.; Wang, J.; Chen, S. Distributed analytics and edge intelligence: Pervasive health monitoring at the era of fog computing. In Proceedings of the 2015 Workshop on Mobile Big Data, Hangzhou, China, 22–25 June 2015; pp. 43–48.

108. Al-Rakhami, M.; Gumaei, A.; Alsahli, M.; Hassan, M.M.; Alamri, A.; Guerrieri, A.; Fortino, G. A lightweight and cost effective edge intelligence architecture based on containerization technology. World Wide Web 2020, 23, 1341–1360.

109. Rincon, J.A.; Julian, V.; Carrascosa, C. Towards the Edge Intelligence: Robot Assistant for the Detection and Classification of Human Emotions. In Proceedings of the International Conference on Practical Applications of Agents and Multi-Agent Systems, L'Aquila, Italy, 13–15 July 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 31–41.

110. Maitra, A.; Kuntagod, N. A novel mobile application to assist maternal health workers in rural India. In Proceedings of the 2013 5th International Workshop on Software Engineering in Health Care (SEHC), San Francisco, CA, USA, 20–21 May 2013; pp. 75–78.

111. Gupta, A.; Mukherjee, N. A cloudlet platform with virtual sensors for smart edge computing. IEEE Internet Things J. 2019, 6, 8455–8462.

112. Elif, Y.H. Novel smart home system architecture facilitated with distributed and embedded flexible edge analytics in demand-side management. Int. Trans. Electr. Energy Syst. 2019, 29, e12014.

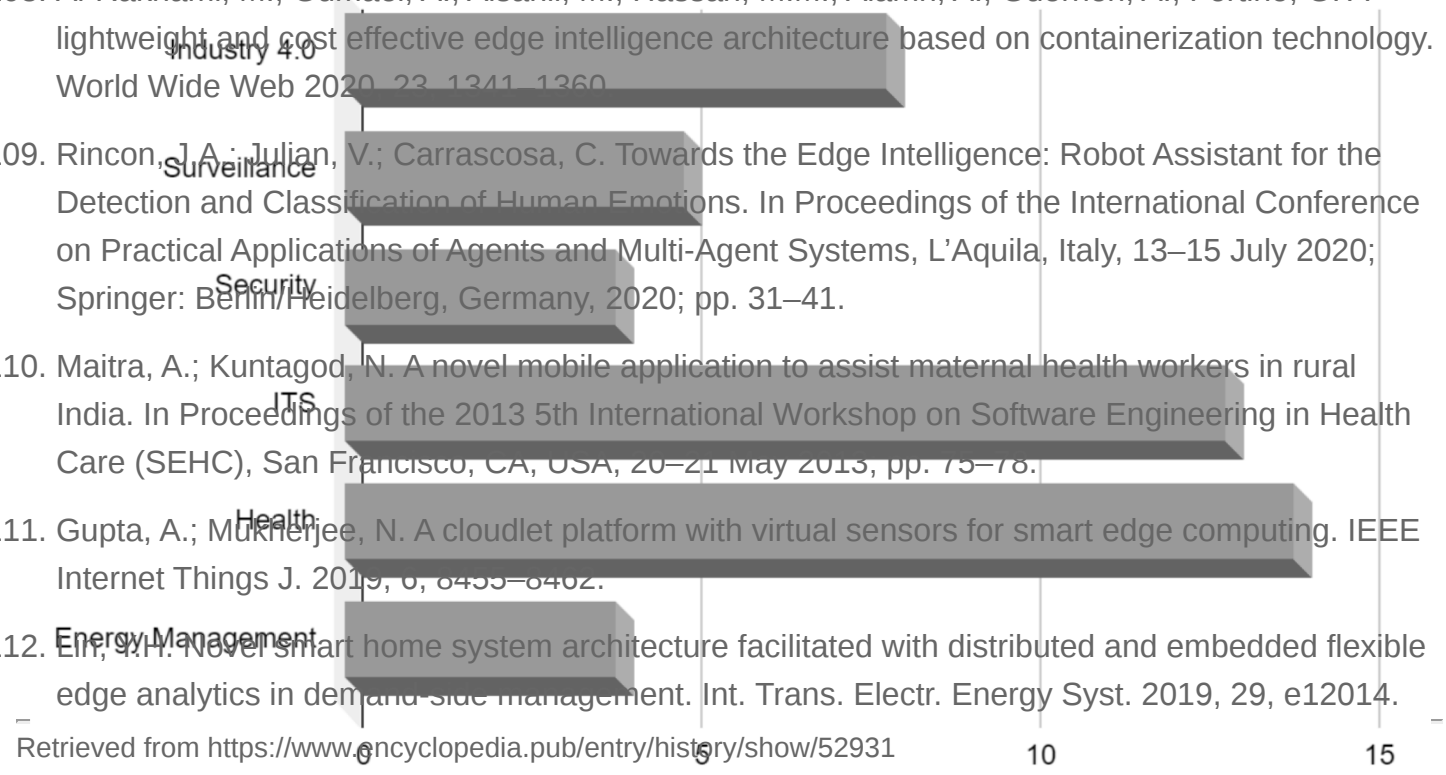

Figure 3. Publications by domain application.

Table 5. Application domains and corresponding works.

Domains	Works That Approach the Theme
Industry (8)	[8][27][47][49][64][95][96][97]
Surveillance (5)	[38][65][98][99][100]
Security (4)	[9][25][67][101]
Intelligent Transport Systems (ITS) (13)	[22][36][39][41][47][48][52][54][71][102][103][104][105],
Health (14)	[13][21][24][43][44][47][63][66][106][107][108][109][110][111]
Energy Management (4)	[34][46][47][112]