Artificial Intelligence in the IoT Era: A Review of Edge AI Hardware and Software

Tuomo Sipola, Janne Alatalo, Tero Kokkonen, Mika Rantonen
JAMK University of Applied Sciences
Jyväskylä, Finland
{tuomo.sipola, janne.alatalo, tero.kokkonen, mika.rantonen}@jamk.fi

***Abstract*—The modern trend of moving artificial intelligence computation near to the origin of data sources has increased the demand for new hardware and software suitable for such environments. We carried out a scoping study to find the current resources used when developing Edge AI applications. Due to the nature of the topic, the research combined scientific sources with product information and software project sources. The paper is structured as follows. In the first part, Edge AI applications are briefly discussed followed by hardware options and finally, the software used to develop AI models is described. There are various hardware products available, and we found as many as possible for this research to identify the best-known manufacturers. We describe the devices in the following categories: artificial intelligence accelerators and processors, field-programmable gate arrays, system-on-a-chip devices, system-on-modules, and full computers from development boards to servers. There seem to be three trends in Edge AI software development: neural network optimization, mobile device software and microcontroller software. We discussed these emerging fields and how the special challenges of low power consumption and machine learning computation are being taken into account. Our findings suggest that the Edge AI ecosystem is currently developing, and it has its own challenges to which vendors and developers are responding.**

I. Introduction

In the last ten years, Artificial Intelligence (AI) solutions have become common in several application areas. In particular, Machine Learning (ML) based solutions are applied to solving a wide range of real-life problems. This variety extends from analysing diseases based on healthcare imaging to predicting energy consumption or detecting anomalous intrusions in network traffic. These AI-based solutions commonly require a large amount of computational capability, which is usually achieved using cloud-based solutions relying on High Performance Computing (HPC) clusters.

However, the rapidly increasing number of Internet of Things (IoT) applications has also raised the number of devices and applications that are producing, collecting and analysing data on the edge of the network. This has naturally increased the interest in applying AI computation on the edge. This concept of using AI near the devices that are producing data is called Edge AI. One of the earliest publications about Edge AI [1] notices two requirements promoting the use of Edge AI: (i) connection robustness and its latency and (ii) privacy issues when uploading data to cloud-based servers. Lee et al. present following examples of those issues: the AI calculation of self-driving cars must be immediate and cannot be in the cloud behind the network latencies or bad network connection, similarly uploading of the video recording with personal data to the cloud based servers raises privacy considerations [1]. The concept of fog computing, i.e., placing computation nodes below the cloud, between edge and the cloud [2], covers similar technologies in the same problem space.

However, even if the concept of Edge AI seems to be coherent, there are some known issues with it. As stated by Shi et al. [3], deploying complete AI models (such as deep neural networks) to the Edge device is generally impracticable because of the hardware boundaries; the size of the model is too large or the computational requirements are too high. A potential implementation is to accomplish collaboration between different Edge AI devices and solutions [3]. Bharwaj et al. [4] identify three challenges for Edge AI:

1) *Computation-aware learning on IoT*. Most of the IoT devices are power and/or memory constrained and in that sense the computation-aware compression of AI models is required.
2) *Data-independent model compression for learning from small data*. The original private data sets of big-data models cannot be used for model compression.
3) *Communication-aware deployment of deep learning models on multiple IoT devices*. Distributing computation with IoT devices could be difficult because of limited communication resources.

As can be seen, IoT-based Edge AI devices produce distinct data. The analysed information must be exchanged between collaborating Edge AI devices in order to achieve sufficient overall capability. Lin et al. [5] introduce blockchain-based architecture for a knowledge market for trading the knowledge of Edge AI devices. Security and privacy issues of Edge AI should be considered thoroughly, as with any data processing systems. Sachdev presented security and privacy issues of Edge AI in digital marketing and concluded that one of the main challenges is how Edge AI can extensively be implemented in that context [6]. Kumar et al. proved by using the classical k-means algorithm in Edge AI concept the feasibility of maintaining privacy preservation of data with Edge AI processing [7].

There have been earlier reviews about Edge AI focusing on different aspects of the emerging field. Wang et al. [8] present

a survey of technologies related to Edge AI emphasising *edge intelligence* and *intelligent edge*. Edge intelligence is the concept of deploying machine learning models to the devices using those models on the edge. This is done in order to lower latency and make the applications more reliable. The concept of intelligent edge focuses on maintenance and management of edge devices. The intelligence via machine learning is used to adaptively control the shared edge resources. The survey also introduces various applicable scenarios for both technologies. Similar classification of the types of Edge AI is introduced by Deng et al. [9], but instead of using the terms *edge intelligence* and *intelligent edge* they are using the terms *AI on Edge* and *AI for Edge*. In addition of making the distinction between the classes, the paper also reviews the state of the art and grand challenges in both categories. Reuther et al. [10] have surveyed machine learning accelerators. They present and categorise close to hundred chips and systems covering everything from low power solutions to data center systems. Li and Liewig have done a similar review [11]. The paper also lists some future trends that AI accelerators might implement in the future. Crespo [12] has collected a list of hardware, software and other resources that are related to Edge AI to a GitHub project where community members can contribute to share knowledge of the topic. Merenda et al. [13] have carried out a literature review on the topic of running Edge AI on resource constrained devices. They review different algorithms, hardware, infrastructure architectures, wireless standards, privacy issues and solutions, and edge training solutions that can be used with these devices. Furthermore, the authors performed a test deployment of a convolutional neural network model to a real world microcontroller system. Ray [14] has carried out an extensive review of machine learning state-of-the-art and prospects on embedded devices (TinyML).

This scoping study aims to create an overview of the Edge AI ecosystem and provide answers to the following questions:

1) What application areas can be identified for Edge AI?
2) What Edge AI hardware platforms exist?
3) What Edge AI software packages exist?

The first question is meant to be a cursory glance at the possibilities and latest trends in applications. The hardware and software sections provide a fresh look at the tools available for Edge AI development.

II. Methodology

This research is structured as a scoping study that describes and summarises an emerging field [15]. We use the stages of scoping study described by Arksey and O'Malley [16]:

1) Identify the question,
2) Identify relevant studies and product descriptions,
3) Select relevant studies and product descriptions,
4) Chart the data,
5) Collate, summarize, report the results.

Since the topic we are interested in heavily depends on hardware products and software packages, we have changed the process to include vendor marketing material in addition to studies. As queries about the three research questions revealed information about use cases, hardware and software, the found articles and resources were included in the three categories accordingly. For Edge AI applications, we queried Google Scholar and IEEE Xplore with the search phrases "Edge AI" and "Edge AI application". Our search resulted in several articles about hardware platforms and those were included in the hardware platforms chapter. The goal here was to gain a general overview about different applications, so the most relevant and diverse ones were chosen.

AI hardware platforms were searched in Google with the phrase "edge AI hardware platform". The first 50 results were evaluated. We excluded offerings that focused on providing services or projects centered around data. This method is useful because this is the way most users would search for products. However, we must be aware that search engine optimization for marketing purposes and Google's own ranking can skew the results. New devices and manufacturers were discovered when going through the found product descriptions and results from the software and applications searches. Those were also included in the study where needed. Especially Crespo's list was found useful [12].

For the Edge AI software section we settled on three distinct categories that clearly can be placed under the Edge AI software term. These categories are *neural network model optimization*, *Edge AI on mobile devices* and *Edge AI on embedded devices*. For the neural network optimization category, we studied recent review publications about the different optimization methods and then reviewed the two most popular neural network frameworks, TensorFlow and PyTorch for support to these methods. For Edge AI on mobile devices section, we reviewed the two most popular mobile device operating systems, Android and iOS, for machine learning support. The final category, Edge AI on embedded devices, turned out to be problematic. Finding the software products that belong to this category was a challenging task. There does not seem to be a single search term that reliably finds these projects. The problem might be that the terminology and workflows have not yet properly settled for the Edge AI field of study. We tried to use search terms such as *TinyML software*, *IoT AI software*, *embedded devices AI software* and *microcontrollers AI software*, but the results were saturated by blogspam like content. Finally, the best sources for actual software projects that belong to this category were community collected lists on open forums [12], [17]. After finding projects that way, some additional projects were discovered by searching survey and benchmarking publications and blog posts with those project names. Often the projects were compared to some other projects that were not listed yet. After the projects were found, their features were evaluated using their publicly available documentation and compared with each other.

III. Edge AI applications

As with traditional AI solutions, there exists a wide range of applications in the Edge AI world. Because the Edge AI concept is relatively new, the published research papers started

appearing in 2018 and most of them after 2019. The research of published studies on Edge AI applications identifies five main categories of applications: Security, Mobile networks, Healthcare, Voice and Image Analysis and Frameworks.

AntiConcealer is an Edge AI approach for detecting adversary concealed behaviors in the IoT [18]. For the security solutions, Edge AI is also used for anomaly detection in the advanced metering infrastructures [19], while Nawaz et al. introduce Ethereum blockchain based solution for analysing the data and tracking the parties accessing that analysis data [20].

Examples of Edge AI solutions for mobile communication and networking are the paper about learning method to support mobile target tracking in the edge platform [21] and another one introducing a resource allocation scheme for 6G [22].

In the healthcare domain Edge AI is for example applied for detecting diabetic retinopathy [23]. Queralta et al. proposed an architecture for health monitoring [24]. Edge AI is used for predicting diseases such as respiratory diseases [25] and chronic obstructive pulmonary disease [26].

As in the traditional AI applications, Edge AI applications are heavily used for voice and image detection and analysis. Shen et al. [27] introduce Edge AI based human head detection algorithm, and Gamanayake et al. propose an Edge AI based method for image pruning [28]. Edge AI based solution is implemented for acoustic classification to be deployed in the autonomous cars [29]. Miyata et al. [30] illustrate Edge AI based mobile robot including voice and object recognition. Application for tangible real-world problem solving is the An Edge AI based apple detection solution has been created to count apples and estimate their sizes [31].

There are also different frameworks published for Edge AI solutions, for example NeuroPilot, a cross-platform framework for Edge AI [32] and an Edge AI framework for telemetry collection and utilization, evaluating both graphics card (GPU) and field-programmable gate array (FPGA) platforms. [33]

IV. Edge AI hardware platforms

The concept of Edge AI is tied to the idea of placing computing power physically near the data source. Any desktop or server rack computer could serve as an edge device. However, many environments are not optimal for such devices. Their size and power consumption are also a major concern. For these reasons, specific Edge AI devices have been designed. Their size and wireless connectivity make them easily attachable to industrial environments. Limited power consumption is also essential when many devices are deployed at once. Moreover, the need for specific mathematical capabilities has given rise to the AI accelerator modules.

Developments in the Edge AI ecosystem drive the devices to be more efficient. Benchmarking hardware platforms has also interested researchers from computing and power consumption points of view. Baller et al. measured five edge devices and give their recommendations for best performance in continuous and sporadic scenarios [34]. Operating AI inference in industrial conditions could be made more robust by using magnetoresistive random access memory (MRAM) [35], [36] Energy efficiency is a constant concern with Edge AI, and there are developments in this area, such as Levisse et al. with their functionality enhanced memories [37]. Liu et al. propose hybrid parallelism, which makes hierarchical training of AI models for Edge AI situations efficient [38].

A. AI acceleration units

Special-purpose acceleration units can either be used as additional processors in any electronic device or as machine learning co-processors in devices that are designed to include such capabilities in addition to traditional processing power and connectivity. AI acceleration units are fast at executing vector and tensor computation and have optimal pipelines for machine learning operations, usually neural network methods. Unfortunately, it is difficult to find meaningful details about many of these devices. Intel Neural Compute Engine is an accelerator for deep neural networks. It supports native FP 16 floating point and 8-bit fixed point data types and can be used to deploy neural networks in Caffe and TensorFlow formats [39]. MediaTek's AI Processing Unit (APU) is an AI accelerator with multimedia features. APU lists TensorFlow, TensorFlow Lite, Caffe and others as supported neural network formats. APU can perform 8-bit and 16-bit integer and 16-bit floating point calculations. It supports Android Neural Networks API (NNAPI) and a custom API [40], [41]. Google Edge TPU is an application specific integrated circuit (ASIC) designed to run TensorFlow Lite models. [42]. The NVIDIA Deep Learning Accelerator (NVDLA) is built specifically for neural network operations. Its processors map to the corresponding mathematical operations used during deep learning. It supports a wide range of data types [43]. The Gyrfalcon Matrix Processing Unit (MPE) is built to compute matrix operations related to neural networks [44]. Mythic has created an analog matrix processor called M1076 Mythic AMP, which uses the Mythic Analog Compute Engine (ACE). Supported data formats are 4-bit, 8-bit and 16-bit integers, and PyTorch, Caffe and TensorFlow models can be used [45]. Syntiant has created a product line of Neural Decision Processors in order to create faster possibilities for neural network solutions, including speech recognition [46], [47], sensor applications [48], [49] and vision [50]. Hailo offers an AI processor that supports 8- and 16-bit numeric presentations and TensorFlow and ONNX for software [51].

B. Field-programmable gate arrays

One trend in Edge AI devices is to employ a field-programmable gate array (FPGA) to build a processor suitable for the specific task of using machine learning methods. Because FPGAs allow great flexibility in what the processor does, they are very useful in building AI accelerators. Intel has produced FPGAs whose applications cover Edge AI: MAX V CPLD [52], Cyclone 10 LP FPGA [53] and Cyclone 10 GX FPGA [54]. For example, a CPU intended for IoT and Edge AI has been developed using the MAX 10 FPGA [55]. There have also been, e.g., frameworks using a FPGA for accelerating machine learning in edge environments [56]

C. System-on-a-chip and system-on-module devices

Intel's Movidius Myriad X Vision Processing Unit is a video processor with neural network inference capabilities. It has 16 cores and a dedicated on-chip Neural Compute Engine and can be used with up to 8 high-definition cameras [39]. Intel has also produced a USB device based on the Movidius Myriad X unit [57] and vision accelerators for edge applications [58]. Systems such as UP Squared 6000 use Movidius Myriad X as an optional visual processing unit [59], Luxonis DepthAI [60] and Luxonis megaAI [61]. HiSilicon's Kirin 970 is a processor for AI computing. It has a dedicated NPU for AI and features aimed at solving computer vision and audio tasks. It also has connectivity in the cellular network using an LTE modem [62]. Qualcomm's Snapdragon 855+/860 is aimed at photography and gaming. However, the on-device AI engine can perform vector and tensor acceleration. It has an LTE modem for cellular connectivity along with Wi-Fi, Bluetooth and near field communication (NFC) [63], MediaTek's Helio P90 is also geared towards imaging, photography, and gaming and features cellular connectivity (LTE), Wi-Fi and Bluetooth. The AI system is marketed for image processing [64]. MediaTek also has AIoT Chipset Platforms specifically for IoT and Edge AI casesincluding displays [65], voice recognition [66], audio and video processing [67] and AI vision [68], although only the last two have a dedicated AI processor. Other devices using the APU units include Helio P95 [69], the Dimensity 1000 series [70] and Dimensity 9000 [71]. MediaTek has also released a short paper about their Edge AI solutions [72]. Nowadays, Rock Chip offers two processor models for Edge AI. These processors are aimed at image and voice processing, especially for mobile devices [73], [74] Kendryte K210 is a chip designed for face recognition. It uses the TinyYOLO object detection neural network [75]. JeVois-A33 [76] is an open-source camera with computer vision AI capabilities. JeVois-Pro [77] has in internal neural processing unit but can also be updated with Coral and Movidius Myriad X units.

D. Coral

Google's Coral Accelerator Module [42] is a solderable module that contains the tensor processing unit Edge TPU. It is also offered using various connectors: Coral USB Accelerator [78], Coral M.2 Accelerator [79], Coral M.2 Accelerator with Dual Edge TPU [80] and Coral Mini PCIe Accelerator [81]. The Coral Dev Board Mini [82] can be used to develop and test applications to be used with the accelerator itself. Coral System-on-Module [83] is an integrated system that includes the Edge TPU accelerator. The module is meant for deployment into production environments. It has a development board counterpart called Coral Dev Board [84]. There are also camera [85] and sensor add-ons available [86].

E. Jetson

NVIDIA has produced devices for edge computing using graphical processing units, which can be used for the vector calculations needed in machine learning. These Jetson models with additional connectivity include Jetson Nano [87], Jetson TX2 NX [88], Jetson TX2 4GB [89], Jetson TX2 [90] and Jetson TX2i [91]. As discussed earlier, NVIDIA's NVDLA is a deep learning accelerator. The use of a separate processing unit for neural network calculations releases the GPU for multimedia tasks. There are various Jetson models that use this technology: Jetson Xavier NX 16GB [92], Jetson Xavier NX [93], Jetson AGX Xavier 64GB [94], Jetson AGX Xavier [95], Jetson AGX Xavier Industrial [96], Jetson Orin NX [97] and Jetson AGX Orin [98]. Both the GPU-based and NVDLA-based devices have developer kits available: Jetson Nano Developer Kit [99], Jetson Nano 2GB Developer Kit [100], Jetson Nano Xavier NX Developer Kit [101], Jetson AGX Xavier Developer Kit [102] and Jetson AGX Orin Developer Kit [103]. These kits can be used for prototyping and testing before moving on to the production versions.

F. Gyrfalcon MPE

Gyrfalcon produces its MPE-based devices for Edge AI. Their products cover a wide area of hardware from MPE processors to servers that use that technology. Lightspeeur 2801S Neural Accelerator can be deployed as a USB dongle or as an embedded device. It supports TensorFlow, Caffe and PyTorch [44]. Lightspeeur 5801S Neural Accelerator is the more efficient (operations/Watt) model for consumer edge devices [104]. Lightspeeur 2803S Neural Accelerator provides even more computational power [105]. Lacelli Edge Inferencing Server AI Acceleration Subsystem uses Lightspeeur 2803S chips on M.2 cards [106]. Gainboard 2801 provides MPE capabilities via the PCIe connector [107]. Gainboard 2803 does the same for the other neural accelerator [108]. Janux G31 AI Server is an AI server with 32 MPE cards [109].

G. Mythic

Mythic's unique perspective is using an analog engine to run its M1076 processor [45]. MP10304 Quad-AMP PCIe Card has four processors [110]. MM1076 M.2 M key card makes one processor usable via the M.2 bus [111]. ME1076 M.2 A+E key card offers it in smaller size and bandwidth [112]. There is also an evaluation system MNS1076 AMP [113].

H. Development boards

Beagle Bone AI is an open-source device featuring TI C66x digital signal processor (DSP) cores and TI embedded vision engines (EVE). It is marketed as focusing on everyday automation, including industrial applications. It has USB and Ethernet connectivity, along with Wi-Fi and Bluetooth [114]. OpenMV Cam is a microcontroller board for machine vision. It has a 480p resolution camera and a USB connection. This small device can run TensorFlow Lite models in addition to multiple basic machine vision tasks [115]. SparkFun Edge Development Board Apollo3 Blue is a low-power board that can run TensorFlow Lite models [116]. Syntiant's Tiny Machine Learning Development Board uses their NDP101 Neural Decision Processor. [117] STMicroelectronics' STM32 microcontroller units can be used for Edge AI solutions [118], [119], e.g., the STM32L4 Discovery kit IoT node provides a

development board for IoT [120]. Hailo offers its AI processor via M.2 and PCIe bus. There are also two evaluation boards available [51]. Other possible Edge AI hardware vendors include Adlink [121], Blaize [122], Aetina [123] and ARM with the Ethos-U65 [124]. Another popular platform is the Raspberry Pi, for example the newest model Raspberry Pi 4 [125].

I. Device tables

Tables I and II list the devices and their basic specifications: central processing unit (CPU) and possible graphics processing unit (GPU), neural processing unit (NPU), memory and type of the device. The NPU can also be a digital signal processing (DSP) unit. Maximum indicated RAM is also reported. Not all details were relevant for the device, available, or they were too ambiguous, so this information is indicated by a dash (–). We follow the hardware taxonomy proposed by Li and Liewig [11], but we have extended the system-on-module to indicate the connection type. We have also marked server devices as a separate category. Thus, the categories are: system-on-a-chip (SoC), system-on-module (SOM), single-board computer (SBC) and server. SOM connection types include external universal serial bus device (USB), external M.2 card slot device (M.2), PCIe slot device (PCIe).

V. Edge AI software

This section lists and explains software projects and tools that are useful in the context of Edge AI. The section is subdivided in the following way. Subsection V-A lists the software that is generally useful for preparing neural network models for running on resource constrained devices. Subsection V-B lists the software that is useful when deploying machine learning models to modern smart phones and other powerful mobile devices. Subsection V-C lists the software that is useful when the target is a microcontroller.

A. Neural network model optimization

When speaking about deep learning, the most popular frameworks are TensorFlow [126], Keras [127] (high level TensorFlow API) and PyTorch [128] according to Kaggle 2021 survey [129]. These frameworks are optimized at running on GPUs and other specialised hardware that accelerate the model training process. After the model has been trained, inference is not as computationally expensive operation, but even that requires moderate amounts of memory and computation power. Devices running on the edge are often resource constrained on that front. There exist techniques that can be used on deep learning models that reduce the model complexity, memory and computation requirements with little or no affect to the model accuracy [130]. Both, TensorFlow and PyTorch, have some of these methods built in to the frameworks that can be used to optimize the model performance on low power devices. TensorFlow has collected the tools and documentation about this topic under TensorFlow Lite subproject [131]. PyTorch has a somewhat similar situation with PyTorch Mobile [132], except that PyTorch Mobile is more of a workflow than a proper subproject and the tooling is included in the main PyTorch API. PyTorch Mobile can also target only mobile devices running Android and iOS, while TensorFlow Lite can also target embedded systems.

The study by Alqahtani et al. [130] concluded that quantization is the most effective method for model optimization. In quantization optimization, the 32-bit floating point model parameters are converted to lower precision integers, which allows the devices to use more computationally efficient integer math operations and the model storage requirements are reduced. Both, TensorFlow and PyTorch, have documentation and easy to use support for model quantization, although PyTorch API is still in beta.

In addition to quantization, TensorFlow also supports model pruning and weight clustering. In model pruning, the model weights are sparsified so that the model compresses better. TensorFlow pruning method is explained in detail in [133]. Weight clustering has the same goal with better model compression. The short explanation of weight clustering is that the layer weights are clustered to N clusters and only the centroid of every cluster is saved. Weight clustering is explained in [134]. PyTorch also supports pruning, but it does not currently have built in support for weight clustering.

B. Edge AI software on mobile devices

Mobile devices that have specialised hardware for neural network acceleration expose the hardware for developers through application programming interfaces (API). On devices that use Android operating system, the API is called Android Neural Networks API (NNAPI) [135]. On Apple operating systems the API is called Core ML [136]. The Android NNAPI is not designed to be used directly by the developers. Instead, it is intented to be used through some higher-level API like TensorFlow Lite. The TensorFlow Lite neural network models are executed on the specialised hardware with TensorFlow Lite NNAPI delegate [137]. PyTorch also has NNAPI support, but it is currently still in beta and not very well documented [138]. NNAPI only supports inference on the device, so it cannot be used for on-device learning. The Apple Core ML framework also supports both TensorFlow and Pytorch through its Unified Conversion API [139], and in addition to that, Apple also has the Create ML framework [140]. It is an easy-to-use interface for developers to create machine learning models that work with Core ML. In comparison to NNAPI another difference is also that Core ML supports on-device training that can be used to personalise a model to user's needs on-device.

In addition to Android NNAPI hardware acceleration API, many vendors have their own software development kits (SDK) for running hardware accelerated models on their systems. Qualcomm has the Qualcomm Neural Processing SDK for AI product [141], Huawei has the HUAWEI HiAI foundation product [142], Mediatek has the Mediatek Neuropilot product [143], and Samsung has the Samsung Neural SDK product [144] although Samsung does no longer provide the SDK to third party developers. Ignatov et al. have a good section about the vendor specific SDKs in [145], [146]. The problem with vendor specific SDKs is that the model created

TABLE I
HARDWARE DEVICE SPECIFICATIONS.

Device	C/GPU	NPU	memory	type
Movidius Myriad X [39]	16-core CPU 700 MHz	Neural Compute Engine	2.5 MB	SoC
Neural Compute Stick 2 [57]	16-core CPU 700 MHz	Neural Compute Engine	2.5 MB	USB
Vision Accelerator [58]	CPU	Neural Compute Engine	4 GB	M.2/PCIe
UP Squared 6000 [59]	4-core CPU 2.0 GHz, GPU	Neural Compute Engine	64 GB	SBC
Kirin 970 [62]	8-core CPU, 12-core GPU	Dedicated NPU	–	SoC
Snapdragon 855+/860 [63]	8-core CPU 2.96 GHz, GPU	DSP	16 GB	SoC
RK1808 [73]	2-core CPU 1.6 GHz	NPU	(DDR)	SoC
RK3399Pro [74]	6-core CPU	NPU	(DDR)	SoC
Helio P90 [64]	8-core CPU 2.2 GHz, GPU	APU 2.0	8 GB	SoC
Helio P95 [69]	8-core CPU, GPU	APU 2.0	8 GB	SoC
i300a [65]	4-core CPU 1.5 GHz, GPU	–	(DDR)	SoC
i300b [66]	4-core CPU 1.3 GHz	–	3 GB	SoC
i350 [67]	4-core CPU 2.0 GHz, GPU	APU 1.0	(DDR)	SoC
i500 [68]	8-core CPU 2.0 GHz, GPU	APU 2-core 500 MHz	(DDR)	SoC
Dimensity 1000 [70]	8-core CPU, GPU	APU 3.0	16 GB	SoC
Dimensity 9000 [71]	8-core CPU, GPU	APU 590	(DDR)	SoC
Beagle Bone AI [114]	2-core CPU 1.5 GHz, GPU	2x DSP, 2x EVE	1 GB	SBC
Coral Accelerator Module [42]	–	Edge TPU	–	SoC
Coral USB Accelerator [78]	–	Edge TPU	–	USB
Coral M.2 Accelerator [79]	–	Edge TPU	–	M.2
Coral M.2 Accelerator with Dual Edge TPU [80]	–	2x Edge TPUs	–	M.2
Coral Mini PCIe Accelerator [81]	–	Edge TPU	–	PCIe
Coral Dev Board Mini [82]	4-core CPU 1.5 GHz, GPU	Edge TPU	2 GB	SBC
Coral System-on-Module [83]	4-core CPU 1.5 GHz, GPU	Edge TPU	4 GB	SOM
Coral Dev Board [84]	4-core CPU 1.5 GHz, GPU	Edge TPU	4 GB	SBC
JeVois-A33 [76]	4-core CPU 1.34 GHz, GPU	–	256 MB	SOM
JeVois Pro [77]	6-core CPU, GPU	Neural Processing Unit	4 GB	SOM
Lightspeeur 2801S Neural accelerator [44]	100 MHz	MPE	–	SoC
Lightspeeur 5801S Neural accelerator [104]	200 MHz	MPE	–	SoC
Lightspeeur 2803S Neural accelerator [105]	250 MHz	MPE	–	SoC
Lacelli Edge Inferencing Server [106]	32-core CPU	4x MPE	32x 8 GB	server
Gainboard 2801 [107]	–	MPE	–	PCIe
Gainboard 2803 [108]	–	MPE	–	PCIe
Janux G31 AI Server [109]	16-core CPU	32x MPE	–	server
M1076 [45]	–	ACE	–	SoC
MP10304 Quad-AMP PCIe Card [110]	–	4x ACE	–	PCIe
MM1076 M.2 M [111]	–	ACE	–	M.2
ME1076 M.2 A+E [112]	–	ACE	–	M.2
MNS1076 AMP [113]	–	ACE	–	SBC
Kendryte K210 [75]	2-core	–	–	SoC
OpenMV Cam [115]	CPU 480 MHz	–	1 MB	SOM
SparkFun Edge Dev Apollo3 Blue [116]	CPU 48 MHz	–	384 kB	SBC
Syntiant Dev Board [117]	CPU 48 MHz	NDP101	32 kB	SBC
STM32L4 [120]	CPU 80 MHz	–	128 kB	SBC
Hailo-8 [51]	–	Hailo-8	–	SoC
Hailo-8 M.2 [51]	–	Hailo-8	–	SOM
Hailo-8 Mini PCIe [51]	–	Hailo-8	–	SOM
Hailo-8 Century Evaluation Platform [51]	–	Hailo-8	–	PCIe
Hailo-8 Evaluation Board [51]	–	Hailo-8	–	SOM
Raspberry Pi 4 [125]	4-core CPU 1.5 GHz	–	8 GB	SBC

TABLE II
JETSON HARDWARE DEVICE SPECIFICATIONS.

device	C/GPU	NPU	memory	type
Jetson Nano [87]	4-core CPU, GPU	–	4 GB	SOM
Jetson TX2 NX [88]	6-core CPUs, GPU	–	4 GB	SOM
Jetson TX2 4GB [89]	6-core CPUs, GPU	–	4 GB	SOM
Jetson TX2 [90]	6-core CPUs, GPU	–	8 GB	SOM
Jetson TX2i [91]	6-core CPUs, GPU	–	8 GB	SOM
Jetson Xavier NX 16GB [92]	6-core CPU, GPU	2x NVDLA v1, 2x PVA v1	16 GB	PCIe
Jetson Xavier NX [93]	6-core CPU, GPU	2x NVDLA v1, 2x PVA v1	8 GB	PCIe
Jetson AGX Xavier 64GB [94]	8-core CPU, GPU	2x NVDLA v1, 2x PVA v1	64 GB	SOM
Jetson AGX Xavier [95]	8-core CPU, GPU	2x NVDLA v1, 2x PVA v1	32 GB	SOM
Jetson AGX Xavier Industrial [96]	8-core CPU, GPU	2x NVDLA v1, 2x PVA v1	32 GB	SOM
Jetson Orin NX [97]	8-core CPU 2.0 GHz, GPU	2x NVDLA v2, PVA v2	12 GB	SOM
Jetson AGX Orin [98]	12-core CPU 2.0 GHz, GPU	2x NVDLA v2, PVA v2	32 GB	SOM
Jetson Nano Developer Kit [99]	4-core CPU 1.42 GHz, GPU	–	4 GB	SBC
Jetson Nano 2GB Developer Kit [100]	4-core CPU1.43 GHz, GPU	–	2 GB	SBC
Jetson Nano Xavier NX Developer Kit [101]	6-core CPU, GPU	2x NVDLA, PVA	8 GB	SBC
Jetson AGX Xavier Developer Kit [102]	8-core CPU, GPU	2x NVDLA, PVA	32 GB	SBC
Jetson AGX Orin Developer Kit [103]	12-core CPU, GPU	2x NVDLA v2.0, PVA 2.0	32 GB	SBC

with one SDK can run only in devices that the vendor specific SDK supports. For that reason it is better to use the more generic NNAPI interface if possible.

C. Edge AI software for microcontrollers

Even when a device does not have specialized hardware for model execution acceleration, the model can always be executed on CPU. This means that neural network model inference can be performed on-device even on the tiniest microcontrollers if the model size is small enough to fit into memory. The problem with microcontrollers is that very often they are not running the operating system that projects such as TensorFlow Lite depend on. This can be solved with TensorFlow Lite for Microcontrollers library [147]. The project was created by merging the uTensor project into the main TensorFlow project [148]. The library is written in C++11 and works on any 32-bit platform. The model data is stored as a C array to read-only program memory on the device where the library can read it. Thus, the library does not need an operating system or a file system for model creation and inference.

Similar to TensorFlow Lite for microcontrollers is the deepC project [149]. The project has the same goal of getting neural network models to work on microcontrollers, but the approach is very different. The project includes a compiler that compiles neural network models directly to C++ code that can then be included in the actual project that uses the model. All neural network models that can be stored in the basic neural network variant of the Open Neural Network Exchange (ONNX) format [150] can be compiled with the deepC compiler. The ONNX format has good support for every major deep learning framework, so the project can be used with a variety of different models and is not restricted to models created by TensorFlow Lite like the TensorFlow Lite for microcontrollers is. The project does not support the ONNX-ML extension of the format that has support for other machine learning algorithms not based on neural networks.

Somewhat similar to deepC project are the deep learning compiler projects Glow [151], ONNC [152], TVM [153] and openVINO [154]. None of these tools are specifically made for compiling code to microcontroller targets, but many of them support microcontroller chips. Sponner et al. have done a good review and a benchmark about these tools targeting embedded platforms in [155]. From these tools the TVM project is probably the most interesting. It does not only contain a compiler for compiling the models to target platforms, but it also contains an auto-tuning feature that tests different compilation optimizations on the target platform to find more optimal compilation results. The project also includes microTVM subproject specially made for compiling models to bare metal microcontroller targets, although the documentation includes a disclaimer that the project is still under heavy development.

Probably not an exhaustive list, but other libraries and toolkits for converting neural network models to microcontrollers are Neural Network on Microcontroller (NNoM) [156], X-CUBE-AI [157], e-AI [158], eIQ [159], nncase [160], NNCG [161] and Embedded Learning Library (ELL) [162]. From these, the NNoM project is the most similar when compared to TensorFlow Lite for Microcontrollers library and the deepC project. It is vendor independent, but it supports only models that are created using Keras. The project includes a compiler that compiles the Keras code to pure C code. If the target platform is ARM Cortex-M processor, the compiler can generate optimized code by utilizing ARM CMSIS NN Software Library [163], [164]. The ARM CMSIS NN library includes optimized versions of the functions that are often used

in neural network models, but it does not include the automatic conversion tool from other deep learning frameworks, so the conversion step would be manual without a tool like NNoM.

The X-CUBE-AI project is an extension package from STMicroelectronics to their STM32CubeMX product. STM32CubeMX is a graphical user interface that allows users to create configuration and initialization code to STM32 microcontrollers [165]. The extension package supports pretrained machine learning models that are made with TensorFlow Lite, or that are exported to the ONNX standard from some other framework. It outputs an optimized code library that works on STM32 microcontrollers. The e-AI project from Renesas is similar to this. Instead of targeting STM32, the tool generates code for Renesas own microcontroller families. It supports deep learning models made with TensorFlow, Pytorch or TensorFlow Lite. The third tool in the same class of vendor specific tools is eIQ by NXP Semiconductors, supporting TensorFlow and ONNX input formats. The compilation target is more modular as the tool supports more inference engines. It can use TensorFlow Lite, Glow, ARM CMSIS-NN or DeepViewRT [166] to run the model on the target platform. The last vendor specific tool is nncase. The generated code targets Kendryte K210 or K510 chips. It supports TensorFlow Lite and ONNX formats.

From the last two neural network conversion tools listed, NNCG is more of a research project and the authors discourage using the tool in production. The project is very similar when compared to NNoM. It converts Keras models to C code. The last listed tool, the Embedded Learning Library (ELL) project is made by Microsoft. It is work in progress, and the authors warn about unexpected API changes. The project documentation is also lacking with only few tutorials about deploying machine learning models to Raspberry Pi single board computers. The project repository commit history shows that the project has received only few updates in recent years, so the project might be obsolete.

Table III summarizes the vendor neutral open-source deep learning model compilers and converters in a table format for easier comparison.

The previously listed tools are made for getting neural network inference to work with microcontrollers. In addition to them, the more traditional machine learning models can also be used to do inference on the edge devices. Often the traditional models are not computationally as demanding as neural networks, but the problem is that very often the models are created using some Python based framework like scikit-learn [167]. It is possible to run Python code on microcontrollers with a project like MicroPython [168], but this creates unnecessary overhead for code execution. It is better to convert the model to more efficient machine code to save as much as possible of the limited resources that the microcontrollers have. This is probably not an exhaustive list, but some of the existing projects to do this conversion are: sklearn-porter [169], emlearn [170], m2cgen [171], EmbML [172], micromlgen [173], Micro-LM [174], micro-learn [175] and weka-porter [176]. Except for Micro-LM and weka-porter, the other conversion projects convert scikit-learn models to C or C++ code. Support for different models varies. sklearn-porter and m2cgen can also convert the model to some other programming language such as Javascript or Java. The weka-porter project supports only WEKA [177] decision tree conversions, and the Micro-LM project supports only models trained with the Desk-LM module [174], although the Desk-LM module in turn depends on and uses scikit-learn library.

VI. Conclusion

The Edge AI ecosystem is still in its infancy. Various products and services are offered, but many of them are placed under the umbrella term for marketing purposes. However, the development of Edge AI as a discipline of its own is evident.

Hardware ranges from special-purpose processors and AI accelerators to full servers. For many users, the various system-on-chip solutions can be useful for the final product. On the other hand, the various development boards and single-board computers provide a good starting point and prototyping possibilities. In addition, the USB, M.2 and PCIe bus devices bring the power of AI acceleration to other devices.

Both of the most popular deep learning frameworks, TensorFlow and PyTorch, can be used to do Edge AI. Between them, TensorFlow is more suitable for Edge AI purposes. The framework includes better documentation and more out of the box methods for model optimization than PyTorch. TensorFlow also supports microcontroller targets with the TensorFlow Lite for microcontrollers subproject while PyTorch only supports mobile device operating system targets.

Between Android and Apple mobile device operating systems support for AI acceleration on hardware, Apple maybe has a better edge by supporting on-device training and having the Create ML framework for creating AI models in addition of supporting all of the most popular AI frameworks.

Edge AI for microcontrollers comes with the most software offerings. The workflow of getting AI models running on microcontroller hardware has not yet found a best practice that everyone uses. There seem to be three competing approaches:

1) Using a runtime that loads the model data from read-only device memory at runtime
2) Using transcompiler that compiles model to C or C++ code that then can be used in the project
3) Using a compiler that compiles the model to a library that is statically or dynamically linked to the project

The good thing is that many of these projects support the ONNX model format, which could mean that benchmarking the different projects with the same model might be easier.

In the future, a standardized software API to access hardware acceleration could offer a more productive development experience. Standardized software workflows or, at least, commonly accepted reference specifications would be highly useful. Software terminology needs unification across research and vendors. Furthermore, security considerations should be studied further, as many of the Edge AI solutions could suffer from the same vulnerabilities as common IoT systems.

TABLE III
OPEN-SOURCE DEEP LEARNING MODEL COMPILERS.

Project	License	Supported models	Tool output	Platform support requirements
TensorFlow lite for microctonrollers [147]	Apache-2.0	TensorFlow	TensorFlow lite flat buffer	C++ compiler
deepC [149]	Apache-2.0	ONNX	C++ code	C++ compiler
Glow [151]*	Apache-2.0	ONNX, Caffe2, TensorFlow Lite	Compiled library bundle (object, header and weight files)	LLVM support
ONNC [152]	BSD-3-Clause	ONNX	C code and binary weight files	C compiler
TVM [153] (microTVM)	Apache-2.0	TensorFlow, TensorFlow Lite, Keras, PyTorch, ONNX, Core ML, caffe2, mxnet, PaddlePaddle	C code or compiled object file, Graph JSON file and Parameter file	C compiler and standard library
NNoM [156]	Apache-2.0	Keras	C code	C compiler

* Ahead-of-time compilation mode

ACKNOWLEDGMENT

This research was funded by the Regional Council of Central Finland/Council of Tampere Region and European Regional Development Fund as part of the *Data for Utilisation – Leveraging digitalisation through modern artificial intelligence solutions and cybersecurity* and *coADDVA - ADDing VAlue by Computing in Manufacturing* projects of JAMK University of Applied Sciences.

The authors would like to thank Ms. Tuula Kotikoski for proofreading the manuscript.

REFERENCES

[1] Y.-L. Lee, P.-K. Tsung, and M. Wu, "Techology trend of edge AI," in *2018 International Symposium on VLSI Design, Automation and Test (VLSI-DAT)*, 2018, pp. 1–2.

[2] S. Yi, Z. Hao, Z. Qin, and Q. Li, "Fog computing: Platform and applications," in *2015 Third IEEE Workshop on Hot Topics in Web Systems and Technologies (HotWeb)*, 2015, pp. 73–78.

[3] Y. Shi, K. Yang, T. Jiang, J. Zhang, and K. B. Letaief, "Communication-efficient edgeAI: Algorithms and systems," *IEEE Communications Surveys Tutorials*, vol. 22, no. 4, pp. 2167–2191, 2020.

[4] K. Bhardwaj, N. Suda, and R. Marculescu, "EdgeAI: A vision for deep learning in the IoT era," *IEEE Design Test*, vol. 38, no. 4, pp. 37–43, 2021.

[5] X. Lin, J. Li, J. Wu, H. Liang, and W. Yang, "Making knowledge tradable in edge-ai enabled iot: A consortium blockchain-based efficient and incentive approach," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 12, pp. 6367–6378, 2019.

[6] R. Sachdev, "Towards security and privacy for edge AI in IoT/IoE based digital marketing environments," in *2020 Fifth International Conference on Fog and Mobile Edge Computing (FMEC)*, 2020, pp. 341–346.

[7] H. H. Kumar, K. V R, and M. K. Nair, "Federated k-means clustering: A novel edge AI based approach for privacy preservation," in *2020 IEEE International Conference on Cloud Computing in Emerging Markets (CCEM)*, 2020, pp. 52–56.

[8] X. Wang, Y. Han, V. C. M. Leung, D. Niyato, X. Yan, and X. Chen, "Convergence of edge computing and deep learning: A comprehensive survey," *IEEE Communications Surveys Tutorials*, vol. 22, no. 2, pp. 869–904, 2020.

[9] S. Deng, H. Zhao, W. Fang, J. Yin, S. Dustdar, and A. Y. Zomaya, "Edge intelligence: The confluence of edge computing and artificial intelligence," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 7457–7469, 2020.

[10] A. Reuther, P. Michaleas, M. Jones, V. Gadepally, S. Samsi, and J. Kepner, "Survey of machine learning accelerators," in *2020 IEEE High Performance Extreme Computing Conference (HPEC)*, 2020, pp. 1–12.

[11] W. Li and M. Liewig, "A survey of AI accelerators for edge environment," in *Trends and Innovations in Information Systems and Technologies. WorldCIST 2020*, ser. Advances in Intelligent Systems and Computing, Á. Rocha, H. Adeli, L. Reis, S. Costanzo, I. Orovic, and F. Moreira, Eds. Cham: Springer, 2020, vol. 1160, pp. 35–44.

[12] X. Crespo. (2022) AI at the edge. [Online]. Available: https://github.com/crespum/edge-ai

[13] M. Merenda, C. Porcaro, and D. Iero, "Edge machine learning for ai-enabled iot devices: A review," *Sensors*, vol. 20, no. 9, 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/9/2533

[14] P. P. Ray, "A review on tinyml: State-of-the-art and prospects," *Journal of King Saud University - Computer and Information Sciences*, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1319157821003335

[15] D. Levac, H. Colquhoun, and K. K. O'Brien, "Scoping studies: advancing the methodology," *Implementation Science*, vol. 5, 2010.

[16] H. Arksey and L. O'Malley, "Scoping studies: towards a methodological framework," *International Journal of Social Research Methodology*, vol. 8, pp. 19–32, 2005.

[17] Hattori, J. Doucette, R. Puntaier, and cppRohit. (2021) How to embed/deploy an arbitrary machine learning model on microcontrollers? [Online]. Available: https://ai.stackexchange.com/questions/25775/how-to-embed-deploy-an-arbitrary-machine-learning-model-on-microcontrollers

[18] J. Zhang, M. Z. A. Bhuiyan, X. Yang, T. Wang, X. Xu, T. Hayajneh, and F. Khan, "Anticoncealer: Reliable detection of adversary concealed behaviors in edgeai assisted iot," *IEEE Internet of Things Journal*, pp. 1–1, 2021.

[19] R. E. Ogu, C. I. Ikerionwu, and I. I. Ayogu, "Leveraging artificial intelligence of things for anomaly detection in advanced metering infrastructures," in *2020 IEEE 2nd International Conference on Cyberspac (CYBER NIGERIA)*, 2021, pp. 16–20.

[20] A. Nawaz, T. N. Gia, J. P. Queralta, and T. Westerlund, "EdgeAI and blockchain for privacy-critical and data-sensitive applications," in *2019 Twelfth International Conference on Mobile Computing and Ubiquitous Network (ICMU)*, 2019, pp. 1–2.

[21] J. Zhang, M. Z. A. Bhuiyan, X. Yang, A. K. Singh, D. F. Hsu, and E. Luo, "Trustworthy target tracking with collaborative deep reinforcement learning in edgeai-aided iot," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 2, pp. 1301–1309, 2022.

[22] J. Sanghvi, P. Bhattacharya, S. Tanwar, R. Gupta, N. Kumar, and M. Guizani, "Res6edge: An edge-ai enabled resource sharing scheme

for c-v2x communications towards 6g," in *2021 International Wireless Communications and Mobile Computing (IWCMC)*, 2021, pp. 149–154.
[23] G. Mathew, S. Sindhu Ramachandran, and S. V.S., "Edgeai: Diabetic retinopathy detection in intel architecture," in *2020 IEEE / ITU International Conference on Artificial Intelligence for Good (AI4G)*, 2020, pp. 75–80.
[24] J. P. Queralta, T. N. Gia, H. Tenhunen, and T. Westerlund, "Edge-AI in LoRa-based health monitoring: Fall detection system with fog computing and LSTM recurrent neural networks," in *2019 42nd International Conference on Telecommunications and Signal Processing (TSP)*, 2019, pp. 601–604.
[25] S. O. Ooko, D. Mukanyiligira, J. P. Munyampundu, and J. Nsenga, "Edge AI-based respiratory disease recognition from exhaled breath signatures," in *2021 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT)*, 2021, pp. 89–94.
[26] ——, "Synthetic exhaled breath data-based edgeAI model for the prediction of chronic obstructive pulmonary disease," in *2021 International Conference on Computing and Communications Applications and Technologies (I3CAT)*, 2021, pp. 1–6.
[27] F.-J. Shen, J.-H. Chen, W.-Y. Wang, D.-L. Tsai, L.-C. Shen, and C.-T. Tseng, "A CNN-based human head detection algorithm implemented on EdgeAI chip," in *2020 International Conference on System Science and Engineering (ICSSE)*, 2020, pp. 1–5.
[28] C. Gamanayake, L. Jayasinghe, B. K. K. Ng, and C. Yuen, "Cluster pruning: An efficient filter pruning method for edgeAI vision applications," *IEEE Journal of Selected Topics in Signal Processing*, vol. 14, no. 4, pp. 802–816, 2020.
[29] R. Rawat, S. Gupta, S. Mohapatra, S. P. Mishra, and S. Rajagopal, "Intelligent acoustic module for autonomous vehicles using fast gated recurrent approach," in *2021 4th International Conference on Recent Developments in Control, Automation Power Engineering (RDCAPE)*, 2021, pp. 345–350.
[30] R. Miyata, O. Fukuda, N. Yamaguchi, and H. Okumura, "Object search using Edge-AI based mobile robot," in *2021 6th International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS)*, vol. 6, 2021, pp. 198–203.
[31] V. Mazzia, A. Khaliq, F. Salvetti, and M. Chiaberge, "Real-time apple detection system using embedded systems with hardware accelerators: An edge AI application," *IEEE Access*, vol. 8, pp. 9102–9114, 2020.
[32] T.-C. Chen, W.-T. Wang, K. Kao, C.-L. Yu, C. Lin, S.-H. Chang, and P.-K. Tsung, "Neuropilot: A cross-platform framework for edge-ai," in *2019 IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, 2019, pp. 167–170.
[33] H. Rexha and S. Lafond, "Data collection and utilization framework for edge AI applications," in *1st IEEE/ACM Workshop on AI Engineering - Software Engineering for AI, WAIN@ICSE 2021, Madrid, Spain, May 30-31, 2021*. IEEE, 2021, pp. 105–108.
[34] S. P. Baller, A. Jindal, M. Chadha, and M. Gerndt, "DeepEdgeBench: Benchmarking deep neural networks on edge devices," in *2021 IEEE International Conference on Cloud Engineering (IC2E)*, 2021, pp. 20–30.
[35] M. Suri, A. Gupta, V. Parmar, and K. H. Lee, "Performance enhancement of edge-AI-inference using commodity MRAM: IoT case study," in *2019 IEEE 11th International Memory Workshop (IMW)*, 2019, pp. 1–4.
[36] V. Parmar, M. Suri, K. Yamane, T. Lee, N. L. Chung, and V. B. Naik, "MRAM-based BER resilient quantized edge-AI networks for harsh industrial conditions," in *2021 IEEE 3rd International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, 2021, pp. 1–4.
[37] A. Levisse, M. Rios, W.-A. Simon, P.-E. Gaillardon, and D. Atienza, "Functionality enhanced memories for edge-AI embedded systems," in *2019 19th Non-Volatile Memory Technology Symposium (NVMTS)*, 2019, pp. 1–4.
[38] D. Liu, X. Chen, Z. Zhou, and Q. Ling, "HierTrain: Fast hierarchical edge AI learning with hybrid parallelism in mobile-edge-cloud computing," *IEEE Open Journal of the Communications Society*, vol. 1, pp. 634–645, 2020.
[39] Intel Corporation. (2022) Intel® Movidius™ Myriad™ X Vision Processing Unit. [Online]. Available: https://www.intel.com/content/www/us/en/products/details/processors/movidius-vpu/movidius-myriad-x.html
[40] MediaTek Inc. (2022) Artificial intelligence. [Online]. Available: https://www.mediatek.com/innovations/artificial-intelligence
[41] ——. (2022) AI for smartphones. [Online]. Available: https://www.mediatek.com/innovations/ai-for-smartphones
[42] Google LLC. (2020) Accelerator Module. [Online]. Available: https://coral.ai/docs/module/datasheet/
[43] NVIDIA Corporation. (2022) NVDLA Primer. [Online]. Available: http://nvdla.org/primer.html
[44] Gyrfalcon Technology Inc. (2022) Lightspeeur 2801s neural accelerator. [Online]. Available: https://www.gyrfalcontech.ai/solutions/2801s
[45] Mythic. (2022) M1076 Analog Matrix Processor. [Online]. Available: https://www.mythic-ai.com/product/m1076-analog-matrix-processor/
[46] Syntiant Corp. (2022) Ndp100 neural decision processor. [Online]. Available: https://www.syntiant.com/ndp100
[47] ——. (2022) Ndp101 neural decision processor. [Online]. Available: https://www.syntiant.com/ndp101
[48] ——. (2022) Ndp102 neural decision processor. [Online]. Available: https://www.syntiant.com/ndp102
[49] ——. (2022) Ndp120 neural decision processor. [Online]. Available: https://www.syntiant.com/ndp120
[50] ——. (2022) Ndp200 neural decision processor. [Online]. Available: https://www.syntiant.com/ndp200
[51] Hailo. (2022) The world's top performing AI processor for edge devices. [Online]. Available: https://hailo.ai/
[52] Intel Corporation. (2022) MAX® V CPLDs. [Online]. Available: https://www.intel.com/content/www/us/en/products/details/fpga/max/v.html
[53] ——. (2022) Intel® Cyclone® 10 LP FPGA. [Online]. Available: https://www.intel.com/content/www/us/en/products/details/fpga/cyclone/10/lp.html
[54] ——. (2022) Intel® Cyclone® 10 GX FPGA. [Online]. Available: https://www.intel.com/content/www/us/en/products/details/fpga/cyclone/10/gx.html
[55] K. Hagiwara, T. Hayashi, S. Kawasaki, F. Arakawa, O. Endo, H. Nomura, A. Tsukamoto, D. Nguyen, B. Nguyen, A. Tran, H. Hyunh, I. Kudoh, and C.-K. Pham, "A two-stage-pipeline CPU of SH-2 architecture implemented on FPGA and SoC for IoT, edge AI and robotic applications," in *2018 IEEE Symposium in Low-Power and High-Speed Chips (COOL CHIPS)*, 2018, pp. 1–3.
[56] K. Karras, E. Pallis, G. Mastorakis, Y. Nikoloudakis, J. M. Batalla, C. X. Mavromoustakis, and E. K. Markakis, "A hardware acceleration platform for AI-based inference at the edge," *Circuits Syst. Signal Process.*, vol. 39, no. 2, pp. 1059–1070, 2020.
[57] Intel Corporation. (2022) Intel Neural Compute Stick 2 (Intel NCS2). [Online]. Available: https://www.intel.com/content/www/us/en/developer/tools/neural-compute-stick/overview.html
[58] ——. (2022) Intel vision accelerator. [Online]. Available: https://www.intel.com/content/www/us/en/developer/topic-technology/edge-5g/hardware/vision-accelerator-movidius-vpu.html
[59] Aaeon Europe BV. (2022) UP Squared 6000 Edge Computing Kit. [Online]. Available: https://up-board.org/up-squared-6000/
[60] Luxonis. (2022) DepthAI. [Online]. Available: https://www.crowdsupply.com/luxonis/depthai
[61] ——. (2022) megaAI. [Online]. Available: https://www.crowdsupply.com/luxonis/megaai
[62] HiSilicon (Shanghai) Technologies Co., Ltd. (2022) Kirin 970. [Online]. Available: https://www.hisilicon.com/en/products/Kirin/Kirin-flagship-chips/Kirin-970
[63] Qualcomm Technologies, Inc. (2022) Snapdragon 855+/860 mobile platform. [Online]. Available: https://www.qualcomm.com/products/snapdragon-855-plus-and-860-mobile-platform
[64] MediaTek Inc. (2022) MediaTek Helio P90. [Online]. Available: https://www.mediatek.com/products/mediatek-helio-p90
[65] ——. (2022) i300a (mt8362a). [Online]. Available: https://www.mediatek.com/products/products/aiot/mt8362a
[66] ——. (2022) i300b (mt8362b). [Online]. Available: https://www.mediatek.com/products/products/aiot/mt8362b
[67] ——. (2022) i350. [Online]. Available: https://www.mediatek.com/products/products/aiot/i350-mt8365
[68] ——. (2022) i500 (mt8385). [Online]. Available: https://www.mediatek.com/products/products/aiot/i500
[69] ——. (2022) MediaTek Helio P95. [Online]. Available: https://www.mediatek.com/products/products/smartphones-2/mediatek-helio-p95
[70] ——. (2022) MediaTek Dimensity 1000 Series. [Online]. Available: https://www.mediatek.com/products/products/smartphones-2/dimensity-1000-series

[71] ——. (2022) MediaTek Dimensity 9000. [Online]. Available: https://www.mediatek.com/products/products/smartphones-2/mediatek-dimensity-9000

[72] P.-K. Tsung, T.-C. Chen, C.-H. Lin, C.-Y. Chang, and J.-M. Hsu, "Heterogeneous computing for edge AI," in *2019 International Symposium on VLSI Design, Automation and Test (VLSI-DAT)*, 2019, pp. 1–2.

[73] Rockchip Electronics Co. (2022) RK1808. [Online]. Available: https://www.rock-chips.com/a/en/products/RK18_Series/2019/0529/989.html

[74] ——. (2022) RK3399Pro. [Online]. Available: https://www.rock-chips.com/a/en/products/RK33_Series/2018/0130/874.html

[75] Canaan Inc. (2022) Kendryte k210. [Online]. Available: https://canaan.io/product/kendryteai

[76] JeVois Smart Machine Vision. (2022) Jevois-a33. [Online]. Available: https://www.jevoisinc.com/pages/hardware

[77] ——. (2022) Jevois-pro. [Online]. Available: https://www.jevoisinc.com/products/jevois-pro-deep-learning-smart-camera

[78] Google LLC. (2019) USB Accelerator datasheet. [Online]. Available: https://coral.ai/docs/accelerator/datasheet/

[79] ——. (2019) M.2 Accelerator. [Online]. Available: https://coral.ai/docs/m2/datasheet/

[80] ——. (2020) M.2 Accelerator with Dual Edge TPU. [Online]. Available: https://coral.ai/docs/m2-dual-edgetpu/datasheet/

[81] ——. (2019) Mini PCIe Accelerator. [Online]. Available: https://coral.ai/docs/mini-pcie/datasheet/

[82] ——. (2020) Dev Board Mini datasheet. [Online]. Available: https://coral.ai/docs/dev-board-mini/datasheet/

[83] ——. (2019) System-on-Module. [Online]. Available: https://coral.ai/docs/som/datasheet/

[84] ——. (2020) Dev Board datasheet. [Online]. Available: https://coral.ai/docs/dev-board/datasheet/

[85] ——. (2020) Camera. [Online]. Available: https://coral.ai/docs/camera/datasheet/

[86] ——. (2019) Environmental sensor board. [Online]. Available: https://coral.ai/docs/enviro-board/datasheet/

[87] NVIDIA Corporation. (2022) Jetson Nano. [Online]. Available: https://developer.nvidia.com/embedded/jetson-nano

[88] ——. (2022) Jetson TX2 NX Module. [Online]. Available: https://developer.nvidia.com/embedded/jetson-tx2-nx

[89] ——. (2022) Jetson TX2 4GB Module. [Online]. Available: https://developer.nvidia.com/embedded/jetson-tx2-4gb

[90] ——. (2022) Jetson TX2 Module. [Online]. Available: https://developer.nvidia.com/embedded/jetson-tx2

[91] ——. (2022) Jetson TX2i Module. [Online]. Available: https://developer.nvidia.com/embedded/jetson-tx2i

[92] ——. (2022) Jetson Xavier NX 16GB. [Online]. Available: https://developer.nvidia.com/embedded/jetson-xavier-nx-16gb

[93] ——. (2022) Jetson Xavier NX. [Online]. Available: https://developer.nvidia.com/embedded/jetson-xavier-nx

[94] ——. (2022) Jetson AGX Xavier 64GB. [Online]. Available: https://developer.nvidia.com/embedded/jetson-agx-xavier-64gb

[95] ——. (2022) Jetson AGX Xavier. [Online]. Available: https://developer.nvidia.com/embedded/jetson-agx-xavier

[96] ——. (2022) Jetson AGX Xavier Industrial. [Online]. Available: https://developer.nvidia.com/embedded/jetson-agx-xavier-i

[97] ——. (2022) Jetson Jetson Orin NX. [Online]. Available: https://developer.nvidia.com/embedded/jetson-orin-nx

[98] ——. (2022) Jetson Jetson AGX Orin. [Online]. Available: https://developer.nvidia.com/embedded/jetson-agx-ori

[99] ——. (2022) Jetson Nano Developer Kit. [Online]. Available: https://developer.nvidia.com/embedded/jetson-nano-developer-kit

[100] ——. (2022) Jetson Nano 2GB Developer Kit. [Online]. Available: https://developer.nvidia.com/embedded/jetson-nano-2gb-developer-kit

[101] ——. (2022) Jetson Xavier NX Developer Kit. [Online]. Available: https://developer.nvidia.com/embedded/jetson-xavier-nx-devkit

[102] ——. (2022) Jetson AGX Xavier Developer Kit. [Online]. Available: https://developer.nvidia.com/embedded/jetson-agx-xavier-developer-kit

[103] ——. (2022) Jetson AGX Orin Developer Kit. [Online]. Available: https://developer.nvidia.com/embedded/jetson-agx-orin-developer-kit

[104] Gyrfalcon Technology Inc. (2022) Lightspeeur 5801s neural accelerator. [Online]. Available: https://www.gyrfalcontech.ai/solutions/lightspeeur-5801

[105] ——. (2022) Lightspeeur 2803s neural accelerator. [Online]. Available: https://www.gyrfalcontech.ai/solutions/2803s

[106] ——. (2022) Lacelli edge inferencing server AI acceleration subsystem. [Online]. Available: https://www.gyrfalcontech.ai/solutions/lacelli-edge-inferencing-server/

[107] ——. (2022) Gainboard 2801 AI for the data center, private & public cloud. [Online]. Available: https://www.gyrfalcontech.ai/solutions/gainboard-2801s/

[108] ——. (2022) Gainboard 2803. [Online]. Available: https://www.gyrfalcontech.ai/solutions/gainboard-2803s/

[109] ——. (2022) Janux G31 AI server. [Online]. Available: https://www.gyrfalcontech.ai/solutions/janux-inference-server/

[110] Mythic. (2022) MP10304 Quad-AMP PCIe Card. [Online]. Available: https://www.mythic-ai.com/product/mp10304-quad-amp-pcie-card/

[111] ——. (2022) Mm1076 m.2 key card. [Online]. Available: https://www.mythic-ai.com/product/mm1076/

[112] ——. (2022) Me1076 m.2 a+e key card. [Online]. Available: https://www.mythic-ai.com/product/me1076/

[113] ——. (2022) Mns1076 amp evaluation system. [Online]. Available: https://www.mythic-ai.com/product/evaluation-system/

[114] BeagleBoard.org Foundation. (2022) Beaglebone® AI. [Online]. Available: https://beagleboard.org/ai

[115] OpenMV, LLC. (2022) OpenMV Cam H7 R2. [Online]. Available: https://openmv.io/collections/cams/products/openmv-cam-h7-r2

[116] SparkFun Electronics. (2022) SparkFun Edge Development Board - Apollo3 Blue. [Online]. Available: https://www.sparkfun.com/products/15170

[117] Syntiant Corp. (2022) Syntiant tiny machine learning development board. [Online]. Available: https://www.syntiant.com/tinyml

[118] STMicroelectronics. (2021) Cartesiam. [Online]. Available: https://cartesiam.ai/

[119] ——. (2022) Artificial intelligence ecosystem for STM32. [Online]. Available: https://www.st.com/content/st_com/en/ecosystems/artificial-intelligence-ecosystem-stm32.html

[120] ——. (2022) B-L475E-IOT01A – STM32L4 Discovery kit IoT node, low-power wireless, BLE, NFC, SubGHz, Wi-Fi. [Online]. Available: https://www.st.com/en/evaluation-tools/b-l475e-iot01a.html

[121] ADLINK Technology Inc. (2021) Edge AI platforms. [Online]. Available: https://www.adlinktech.com/en/Inference_Platform

[122] Blaize. (2022) New AI edge computing – edge AI hardware & software. [Online]. Available: https://www.blaize.com/

[123] Aetina Corporation. (2022) Industrial GPGPU & embedded edge AI computing solutions for critical imaging applications. [Online]. Available: https://www.aetina.com/

[124] Arm Limited. (2022) Ethos-U65 machine learning processor (NPU). [Online]. Available: https://www.arm.com/products/silicon-ip-cpu/ethos/ethos-u65

[125] R. P. Foundation. (2022) Raspberry Pi 4. [Online]. Available: https://www.raspberrypi.com/products/raspberry-pi-4-model-b/

[126] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/

[127] F. Chollet *et al.*, "Keras," https://keras.io, 2015.

[128] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[129] Kaggle. (2021) State of data science and machine learning 2021. [Online]. Available: https://www.kaggle.com/kaggle-survey-2021

[130] A. Alqahtani, X. Xie, and M. W. Jones, "Literature review of deep network compression," *Informatics*, vol. 8, no. 4, 2021. [Online]. Available: https://www.mdpi.com/2227-9709/8/4/77

[131] TensorFlow. (2022) TensorFlow Lite. [Online]. Available: https://www.tensorflow.org/lite/

[132] PyTorch. (2022) PyTorch Mobile. [Online]. Available: https://pytorch.org/mobile/home/

[133] M. H. Zhu and S. Gupta, "To prune, or not to prune: Exploring the efficacy of pruning for model compression," 2018. [Online]. Available: https://openreview.net/forum?id=S1lN69AT-

[134] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding," in *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2016. [Online]. Available: http://arxiv.org/abs/1510.00149

[135] Android. (2022) Android Neural Networks API. [Online]. Available: https://developer.android.com/ndk/guides/neuralnetworks/

[136] Apple. (2022) Core ML Framework. [Online]. Available: https://developer.apple.com/documentation/coreml

[137] Android. (2022) TensorFlow Lite NNAPI delegate. [Online]. Available: https://www.tensorflow.org/lite/performance/nnapi

[138] PyTorch. (2022) (Beta) Convert MobileNetV2 to NNAPI. [Online]. Available: https://pytorch.org/tutorials/prototype/nnapi_mobilenetv2.html

[139] Apple. (2022) Unified Conversion API. [Online]. Available: https://coremltools.readme.io/docs/unified-conversion-api

[140] ——. (2022) Create ML Framework. [Online]. Available: https://developer.apple.com/machine-learning/create-ml/

[141] Qualcomm Technologies, Inc. (2022) Qualcomm Neural Processing SDK for AI. [Online]. Available: https://developer.qualcomm.com/software/qualcomm-neural-processing-sdk

[142] HUAWEI. (2022) HUAWEI HiAI Foundation. [Online]. Available: https://developer.huawei.com/consumer/en/hiai#Foundation

[143] MediaTek Inc. (2022) Mediatek Neuropilot. [Online]. Available: https://neuropilot.mediatek.com/

[144] Samsung. (2022) Samsung Neural SDK. [Online]. Available: https://developer.samsung.com/neural/overview.html

[145] A. Ignatov, R. Timofte, W. Chou, K. Wang, M. Wu, T. Hartley, and L. Van Gool, "Ai benchmark: Running deep neural networks on android smartphones," in *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, September 2018.

[146] A. Ignatov, R. Timofte, A. Kulik, S. Yang, K. Wang, F. Baum, M. Wu, L. Xu, and L. Van Gool, "Ai benchmark: All about deep learning on smartphones in 2019," in *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, 2019, pp. 3617–3635.

[147] TensorFlow. (2021) TensorFlow Lite for Microcontrollers. [Online]. Available: https://www.tensorflow.org/lite/microcontrollers

[148] Shelby, Zach. (2019) uTensor and Tensor Flow Announcement. [Online]. Available: https://os.mbed.com/blog/entry/utensor-and-tensor-flow-announcement/

[149] AI Technology & Systems. (2021) deepC. [Online]. Available: https://github.com/ai-techsystems/deepC

[150] ONNX. (2019) Open Neural Network Exchange: ONNX. [Online]. Available: https://onnx.ai/

[151] N. Rotem, J. Fix, S. Abdulrasool, S. Deng, R. Dzhabarov, J. Hegeman, R. Levenstein, B. Maher, N. Satish, J. Olesen, J. Park, A. Rakhov, and M. Smelyanskiy, "Glow: Graph lowering compiler techniques for neural networks," *CoRR*, vol. abs/1805.00907, 2018. [Online]. Available: http://arxiv.org/abs/1805.00907

[152] W.-F. Lin, D.-Y. Tsai, L. Tang, C.-T. Hsieh, C.-Y. Chou, P.-H. Chang, and L. Hsu, "Onnc: A compilation framework connecting onnx to proprietary deep learning accelerators," in *2019 IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, 2019, pp. 214–218.

[153] T. Chen, T. Moreau, Z. Jiang, H. Shen, E. Q. Yan, L. Wang, Y. Hu, L. Ceze, C. Guestrin, and A. Krishnamurthy, "TVM: end-to-end optimization stack for deep learning," *CoRR*, vol. abs/1802.04799, 2018. [Online]. Available: http://arxiv.org/abs/1802.04799

[154] A. Demidovskij, Y. Gorbachev, M. Fedorov, I. Slavutin, A. Tugarev, M. Fatekhov, and Y. Tarkan, "Openvino deep learning workbench: Comprehensive analysis and tuning of neural networks inference," in *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, 2019, pp. 783–787.

[155] M. Sponner, B. Waschneck, and A. Kumar, "Compiler toolchains for deep learning workloads on embedded platforms," in *Research Symposium on Tiny Machine Learning*, 2021. [Online]. Available: https://openreview.net/forum?id=O0kjwqJyhNd

[156] J. Ma, "A higher-level Neural Network library on Microcontrollers (NNoM)," Oct. 2020. [Online]. Available: https://doi.org/10.5281/zenodo.4158710

[157] STMicroelectronics. (2022) X-CUBE-AI - AI expansion pack for STM32CubeMX. [Online]. Available: https://www.st.com/en/embedded-software/x-cube-ai.html

[158] Renesas Electronics Corporation. (2022) e-AI Solution. [Online]. Available: https://www.renesas.com/us/en/application/key-technology/artificial-intelligence/e-ai

[159] NXP Semiconductors. (2022) eIQ® ML Software Development Environment. [Online]. Available: https://www.nxp.com/design/software/development-software/eiq-ml-development-environment:EIQ

[160] Canaan Inc. (2022) nncase. [Online]. Available: https://github.com/kendryte/nncase

[161] O. Urbann, S. Camphausen, A. Moos, I. Schwarz, S. Kerner, and M. Otten, "A c code generator for fast inference and simple deployment of convolutional neural networks on resource constrained systems," in *2020 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS)*, 2020, pp. 1–7.

[162] M. Corporation. (2018) Embedded Learning Library (ELL). [Online]. Available: https://microsoft.github.io/ELL/

[163] Arm Ltd. (2021) CMSIS NN Software Library. [Online]. Available: https://www.keil.com/pack/doc/CMSIS/NN/html/index.html

[164] L. Lai, N. Suda, and V. Chandra, "CMSIS-NN: efficient neural network kernels for arm cortex-m cpus," *CoRR*, vol. abs/1801.06601, 2018. [Online]. Available: http://arxiv.org/abs/1801.06601

[165] STMicroelectronics. (2022) STM32CubeMX - STM32Cube initialization code generator. [Online]. Available: https://www.st.com/en/development-tools/stm32cubemx.html

[166] Au-Zone Technologies. (2022) DeepViewRT™ Inference Engine. [Online]. Available: https://www.embeddedml.com/deepviewrt

[167] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[168] George Robotics Limited. (2022) MicroPython. [Online]. Available: https://micropython.org/

[169] D. Morawiec, "sklearn-porter," transpile trained scikit-learn estimators to C, Java, JavaScript and others. [Online]. Available: https://github.com/nok/sklearn-porter

[170] J. Nordby, "emlearn: Machine Learning inference engine for Microcontrollers and Embedded Devices," Mar. 2019. [Online]. Available: https://doi.org/10.5281/zenodo.2589394

[171] Titov, Nikita and Zeigerman Iaroslav and Yershov Viktor and others. (2022) m2cgen. [Online]. Available: https://github.com/BayesWitnesses/m2cgen

[172] L. Tsutsui da Silva, V. M. A. Souza, and G. E. A. P. A. Batista, "Embml tool: Supporting the use of supervised learning algorithms in low-cost embedded systems," in *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*, 2019, pp. 1633–1637.

[173] Salerno, Simone. (2022) MicroML. [Online]. Available: https://github.com/eloquentarduino/micromlgen

[174] F. Sakr, F. Bellotti, R. Berta, and A. De Gloria, "Machine learning on mainstream microcontrollers," *Sensors*, vol. 20, no. 9, 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/9/2638

[175] A. P. Singh and S. Chaudhari, *Embedded Machine Learning-Based Data Reduction in Application-Specific Constrained IoT Networks*. New York, NY, USA: Association for Computing Machinery, 2020, p. 747–753. [Online]. Available: https://doi.org/10.1145/3341105.3373967

[176] D. Morawiec. (2022) weka-porter. [Online]. Available: https://github.com/nok/weka-porter

[177] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, *The WEKA Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques"*, 4th ed. Morgan Kaufmann, 2016. [Online]. Available: https://www.cs.waikato.ac.nz/ml/weka/Witten_et_al_2016_appendix.pdf