Edge Intelligence: The Convergence of Humans, Things, and AI

Thomas Rausch
Distributed Systems Group
TU Wien
Vienna, Austria
t.rausch@dsg.tuwien.ac.at

Schahram Dustdar
Distributed Systems Group
TU Wien
Vienna, Austria
dustdar@dsg.tuwien.ac.at

***Abstract*—Edge AI and Human Augmentation are two major technology trends, driven by recent advancements in edge computing, IoT, and AI accelerators. As humans, things, and AI continue to grow closer together, systems engineers and researchers are faced with new and unique challenges. In this paper, we analyze the role of edge computing and AI in the cyber-human evolution, and identify challenges that edge computing systems will consequently be faced with. We take a closer look at how a cyber-physical fabric will be complemented by AI operationalization to enable seamless end-to-end edge intelligence systems.**

***Index Terms*—edge AI, human augmentation, edge intelligence, edge computing, AI systems, AI operations**

I. Introduction

Edge AI and Human Augmentation are currently considered to be two of the major emerging technology trends [1], which we attribute to three key technological developments. First, the increase in data availability, consolidated computing power, as well as the advancement of machine learning (ML) techniques have fostered the development of AI supported applications that continue to transform virtually every aspect of our daily life. Second, the increasing number of Internet connected sensors and smart things embedded in our surroundings, i.e., the Internet of Things (IoT), and wearable smart devices, generate a wealth of opportunities to create many types of applications that can enhance our everyday experience and quality of life. Third, the growing friction between cloud-based AI solutions, and the need of many applications to analyze high volume and high velocity data streams in near real-time [2], [3], has pushed hardware developers to create miniaturized AI accelerators that promise to bring AI to the extreme edge.

It is clear that edge computing [4]–[6] will play a fundamental role in reconciling these developments. Not only are there physical limits to what cloud-based solutions can deliver [6], [7], there are legitimate concerns about privacy and trust, in particular as we become more dependent on AI. Furthermore, as we move towards artificial general intelligence (AGI) or "strong AI", isolated AI programs will become increasingly interconnected and begin to collaboratively solve increasingly complex tasks [8]. While we can bootstrap such a system from the cloud, a decentralized model where AI agents make use of edge resources is likely to follow. We can see that intelligence will gradually be pushed from the cloud closer to the edge.

In this new paradigm of Edge Intelligence, where a cyber-physical fabric not only provides raw data, but can intelligently act on it, edge computing is faced with some unique challenges from the AI systems problem space. The goal of this paper is to better understand these challenges. To that end, we first outline our vision of the convergence of humans, things and AI. Then, we take a systems perspective on the challenges along two dimensions. First, we discuss the underlying infrastructure and coordination mechanisms required for edge intelligence. Connecting the cyber-physical IoT, and combining general purpose compute infrastructure with AI accelerators for edge-based utility computing will require new coordination mechanisms. Second, we discuss operationalizing edge intelligence, that is, enabling end-to-end platforms and workflows to manage the edge AI application life cycle.

II. Convergence of Humans, Things and AI

The development of AI has taken spectacular leaps over the past decade. The increase in data availability, computing power, as well as the advancement of machine learning (ML) techniques and specialized AI hardware, has moved us into the fast-lane towards a society that is shaped by AI in all its aspects. Even in problem areas long thought to be unattainable without human reasoning and intuition, AI has achieved super-human capabilities, as impressively demonstrated by Jeopardy winning IBM Watson [9] or Google DeepMind's AlphaGo [10]. Although researchers and thinkers seem divided into AI optimists and pessimists, one things seems clear. The optimal strategy for us humans, for whatever future awaits us with the development of AI, is that we continue to foster a close partnership with AI. This partnership seems particularly important as the number of Internet connected sensors, devices, and autonomous agents that can sense and manipulate our physical surroundings continues to grow. A cyber-physical fabric of tomorrow that permeates our environment, together with the unprecedented amount of computing power and digitally persisted knowledge, is an opportunity for human cognition to evolve beyond its biological limits.

We take a step back and discuss the evolutionary steps of the cyber-human, as illustrated in Figure 1. The Internet allows us to store and access information at immense scales, and smartphones have become our main gateway to this world.

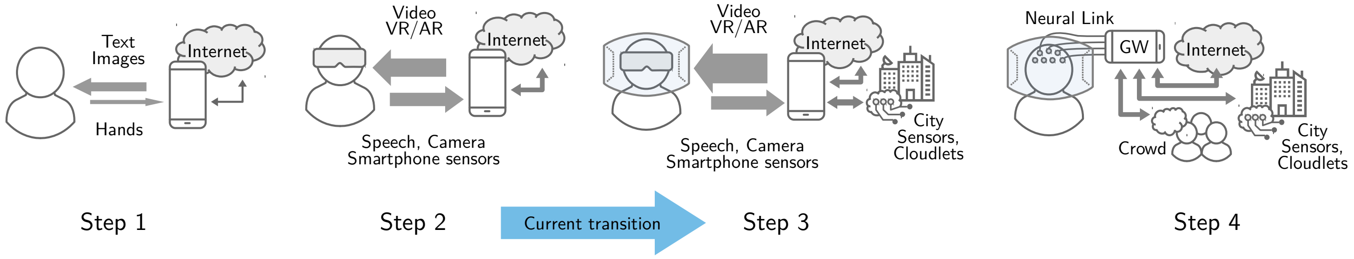

Fig. 1. The currently observable evolution of the cyber-human

For many, the smartphone is already an extension of their self, representing the first step in the cyber-human evolution. Typing in a search term into your smartphone and parsing the information it displays takes us a long time compared to the bandwidth available to a superhuman AI. Similarly, we control our smart things via manual gateways, such as the apps on our phone, at a very low bandwidth. With the development of increasingly sophisticated smart devices such as mixed reality smart glasses, combined with augmented reality (AR) and AI techniques, step two has nudged humans and things closer together [11]. AI applications help us or our cars to recognize objects and overlay our field of vision with contextual information. As edge devices become more intelligent, and more and more sensor data in our surroundings can be aggregated, processed and served via edge resource at high bandwidth, step three will fully realize a transparently immersive experience. However, for the seamless augmentation of human cognition, the bandwidth requirements will be even higher. Imagine a world in which we perceive and interact with the cyber-physical surroundings via neural-interface connected gateways and AI assisted control mechanisms. The requirements of step four in the cyber-human evolution are fundamentally challenging our way of building systems.

An important aspect to highlight in this evolution is the trend towards artificial general intelligence (AGI), i.e., the development of AI that can solve intellectual tasks that a human being can [12]. From a system's perspective, it is likely that AGI will not be achieved by a single method, but rather a context-sensitive ensemble of different task-specific AI agents that permeate our environment via an underlying computational substrate [13]. Some projects that work towards AGI are based on the idea of an interconnected and self-governing network of AI solution agents that can cooperate to solve increasingly more complex problems as the network grows [8]. In such a system, edge computing will play a fundamental enabling role. Compared to the cloud, the edge can provide trust, AI specialized hardware in close proximity to data, and thereby handle the immense bandwidth requirements.

A. How Edge Intelligence can increase the Fidelity of our Perception of Reality

Augmenting human cognition with computation is an idea that dates back decades and has been explored by science-fiction writers and researchers alike. In 2004, Satyanarayanan described a wearable cognitive assistance use case that he said at the time was "clearly science fiction" [14], but was made a reality almost ten years later [11]. This development is the transition between step two and three in our cyber-human evolution timeline. Today, products such as Google Lens[1] attempt to bring these systems to the masses by integrating cloud-based visual analytics AI systems with personal edge devices. Going a step further, consider a high-fidelity personalized visual discovery assistant. You use your smartphone to visually discover things that your assistant knows are important to you. By pointing your camera along a shopping street, the application recognizes streets, buildings and objects, and displays an AR overlay with retailers that have offers that you may be interested in, restaurants that have food you like, or points of interest you may want to visit, combined with real-time information such as the number of guests currently in a restaurant or shop. With current state-of-the art technology, we would realize such applications via a direct link between the camera app and the cloud.

There are however use cases where momentary information is important, such as high-speed manufacturing lines, or virtual assistants of fast-paced interactions, like navigating through a smart space, where cloud-based analytics become infeasible. There are numerous other wearable cognitive assistance applications that require such momentary information [6]. These scenarios focus mostly on analyzing the video stream of a camera, and annotating it with contextual information. Future applications will move beyond isolated sensing, but synthesize data from many sources that cannot be streamed into the cloud. Consider a cognitive assistance application that guides you through a public space shared with other people and autonomous agents such as self-driving cars and mobile robots. Suppose each agent broadcasts its sensor data (gyroscope, accelerometer, location, etc.). Edge resources aggregate sensor data sources in close proximity to create a hyper-reality overlay to give you a personalized experience and guiding you through the space. As fidelity of this overlay depends on available proximate resources and capabilities, an approach is needed for a seamless edge/cloud AI system that can transparently trade off accuracy and computation, where accuracy increases with computational power. Suppose these applications could be served by a continuum of edge resources and the cloud. If models are served via edge resources,

[1] https://lens.google.com/

fidelity of recognition can be much higher, because models can be trained for specific domains and then deployed in close proximity of where they are required. Data can then be streamed instead of chunked (e.g., like Google Lens sends individual images to the cloud), which can further increase fidelity and responsiveness of an AR system.

Step four in the cyber-human evolution will confront researchers with a multitude of challenges over the next decade. Engineering and managing a system to allow the pervasive and seamless integration of all these applications at scale into our everyday lives requires a new way of thinking about infrastructure and processes, beyond our current understanding in pervasive computing, in particular as AI agents become more involved and applications co-evolve with ML models. Deep neural networks can be pre-trained on massive datasets in the cloud, localized data can be used to refine models the edge using transfer learning, and applications can then be deployed on edge resources optimized for inferencing. We discuss this in depth in Section IV.

B. The AI Enhanced Smart City

Smart cities provide a number of exciting opportunities for edge computing [15]. In fact, we believe that edge intelligence will be the key to fully connect the cyber-physical city with its cyber-human inhabitants. Smart public spaces understand situations by interpreting activity. Many dispersed cameras can be used to create data analytics models (like crowd behavior [16], flooding or fire spreading models, or ecosystem/plant monitoring [17]). An AI could use weather data together with camera streams to determine road driving conditions, and put that information in context with historical data to assess the immediate risk of accidents. Autonomous vehicles can in turn use this information to adapt their driving behavior. In the case of accidents, a smart city system would react by immediately informing authorities, dispatching first responders, and rerouting traffic. Air quality sensors distributed throughout the city can be used to track development and spread of air pollution. A number of similar use cases with different sensor and analytics requirements could be conceived. What these use cases illustrate is the common need for real-time, location-based data about urban environments and activity at different levels of fidelity. Moreover, appropriate edge resources are necessary to both process and store data close to where they are generated [18]. Sending data to the cloud to do inferencing on ML models deployed as web services is not feasible in many cases, in particular as the required perception fidelity increases. Researchers and practitioners will therefore be challenged to reconcile smart-city scale sensing capabilities and compute infrastructure with scalable deployment of AI applications to the edge.

C. Democratized, Trusted, and Explainable AI

Not everyone has the same access to data and ML capabilities to train effective models from scratch. However, if someone wants to create an application that combines, say, speech recognition, emotion interpretation, and a conversational system, they could do so using of-the-shelf models and refining them with their domain-specific data [19]. Suppose a platform where users can provide the data they have, specify a set of goals for their application, and a system in the background applies a pipelines of steps to recommend necessary models, do hyperparameter tuning, and then deploy the created models automatically. This is one of the main goals of AutoML research [19], and there are tentative results in realizing such platforms [20]. It is clear that model marketplaces and AutoML techniques (automated data preparation, transfer learning, model selection [21], etc.) are key mechanisms for democratizing AI. Engineering these systems in a centralized way in the cloud seems the obvious course of action. However, often times, data required for training models may be sensitive and not allowed to leave a certain context, such as a companies premises or a network boundary. Similarly, infrastructure for processing the data may have to be certified by some governmental organization, as is the case, for example, in many e-health use cases where sensitive patient data are involved and data processing is audited. In such scenarios, hybrid edge/cloud computing infrastructure plays a significant enabling role.

Another key aspect of AI is trust and explainability [22]. Models are rarely used in an isolated and static environment. Instead, concept drift causes model performance to degrade [23], models may be vulnerable to adversarial attacks [24], or exhibit bias. It is important that we can trust AI, particularly when we employ it scenarios that can have an impact on our health or security. In these cases, model vulnerabilities need to be found and fixed quickly, for example when confidence drops below a given threshold. Moreover, when AI makes predictions in such scenarios, it is crucial that we are able to trace and explain these predictions, even once models have been replaced with updated versions [25]. In later sections, we discuss how trust and explainability are particularly challenging in edge intelligence.

D. Key Ingredients for Edge Intelligence

From our discussion and the presented use cases, we identify two orthogonal elements that play key roles in realizing edge intelligence systems.[2]

a) Computational Fabric: dispersed resources will allow the training, monitoring, and serving of models. The heterogeneity of applications and models will require flexible and modular infrastructure, and the scale of the infrastructure will require intelligent operations mechanisms (AI for operations).

b) Operationalization: automated AI application lifecycle management will have to move beyond the currently predominant cloud-based view. Current mechanisms will have to be extended and synthesized with edge computing techniques to provide scalable operationalization of edge intelligence (operations for AI).

[2] Vital elements of edge intelligence are, of course, specific ML algorithms and techniques. As we take a broader systems perspective, specific adaptations of ML algorithms for edge computing are out of scope of this paper.

In the remainder of the paper, we discuss these two dimensions and the associated challenges.

III. A Fabric for Edge Intelligence

To realize the vision we have outlined, we identify the following critical components for an edge intelligence fabric: 1) a sensing substrate, 2) a network of edge computers with modular AI capabilities, and 3) intelligent orchestration mechanisms to reconcile distributed and decentralized infrastructure. In particular, edge computers will be made up of different types of computing platforms. General purpose computing will be complemented by specialized HPC and AI optimized hardware, federated to create a powerful, high-density multipurpose compute units. Self-learning edge middleware will learn how to optimally schedule workloads depending on the type of workload and available capabilities, which is particularly important for operationalized AI workloads, as we will see in the next section. Hierarchies of connected clustered edge computers will form the backbone, and coordination mechanisms will weave these hierarchies together to a fabric that enables seamless edge intelligence. This highly dynamic and heterogeneous environment raises many challenges, in particular for middleware architectures and operations mechanisms. With the vision and use cases we have outlined in mind, we discuss in more detail each component and associated challenges.

A. *Sensing Substrate & Sensor Data as a Service (SDaaS)*

Implementing use cases such as smart public spaces on smart-city scale requires a large number of different sensing capabilities [15]. Having individual companies develop and deploy specific sensors for specific use cases seems inefficient in the long term. Instead, city planners will have a high stake in providing application developers with a sensing infrastructure to make use of smart city data. It seems much more likely that an initial set of smart city and IoT use cases will bootstrap a general set of requirements for sensor capabilities, which will then trigger a deployment at increasingly larger scale. Similar to a smartphone that has a wide array of sensors installed giving developers lots of opportunities to create novel and creative applications. Not all applications use all sensors, in fact, some usage behavior may not require specific sensors at all. They are there nonetheless, because they provide potential utility for future apps. It is reasonable to assume that a similar model will work well for smart-city scale edge intelligence. Once a family of edge intelligence application emerges, and we understand the broader range of requirements, companies can start building arrays of sensors to support applications with sensing capabilities, hardware and potential utility. In fact, there already projects, such as the Array of Things[3], that can work as substrate for providing Sensors Data as a Service (SDaaS) [26].

We foresee several challenges for edge computing in this model. Given the large number and dynamic and mobile nature of both publishers and subscribers of sensor data [27], and the stringent QoS requirements of edge intelligence use cases, we will be forced to rethink centralized messaging services such as Amazon AWS IoT or Microsoft Azure IoT Hub [28]. Novel messaging systems, such as osmotic message-oriented middleware [29], can help facilitate transparent access to this geographically dispersed network of sensors and actuators, even under highly dynamic node behavior. Furthermore, it is unlikely that sensing infrastructure will be static and managed by a single central authority. Consider the integration of cognitive assistance devices with self-driving cars and smart traffic systems. Applications require access to public sensor data, such as statically deployed arrays of sensors; dynamic sensor data from cars; as well as personal sensor data, such as location and movement of pedestrians. How can these different static and dynamic data sources be reconciled to let AI applications make full use of them? Moreover, while AI inferencing mostly works on real-time data in an event-based way, the majority of training techniques require batch access to labeled data. Storing and providing scalable access to these highly dispersed data is a fundamental challenge for edge intelligence.

B. *Modular AI Capabilities*

AI workloads benefit significantly from specialized hardware. GPU clusters dominate the cloud-based ML landscape, but the same type of GPUs used in these clusters, are impractical for many edge resources due to their size and power requirements (a popular example is the NVIDIA Tesla K80 that has a TDP of 300W). Instead, a new family of AI accelerators have emerged that promise to fully enable AI for resource constrained edge devices. Google has devised the Edge TPU[4], an application-specific integrated circuit (ASIC) that can enable, for example, high-fidelity, real-time vision applications running on a Raspberry Pi. A number of other prominent examples include Microsoft BrainWave, which uses field-programmable gate arrays (FPGAs) [30]; Intel Neural Compute Stick [31]; or Baidu Kunlun [32].

These are exciting developments for edge AI, and the benefits are clear: AI accelerators tend to be small and cost efficient, thereby making it feasible to fit them easily on edge devices, such as SBCs connected to cameras, smart phones, or similar compact or mobile devices. However, as the variety of AI accelerators increases, so does the heterogeneity of the edge fabric, thereby introducing additional complexity for platform abstractions. Moreover, because ASICs are as such designed for specific algorithms, hardware vendors also control what algorithms and platforms we can use, and potential vendor lock-ins pose a significant threat to the democratization of AI. While we see pluggable AI capabilities for edge computers as the way forward in enabling a rich edge intelligence fabric, it is clear that this requires the continued fostering of open standards and robust platform abstractions. It is particularly important that developers can trust the "write once, deploy

[3] https://arrayofthings.github.io/

[4] https://cloud.google.com/edge-tpu/

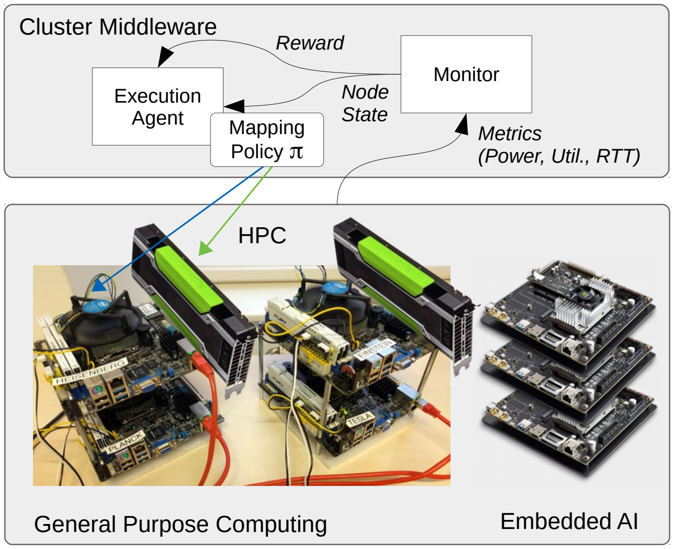

Fig. 2. Multi-purpose edge computer with self-adaptive middleware that learns how to efficiently schedule workloads

anywhere" philosophy, and that a programming models, such as a Function as a Service (FaaS) style model, hide the heterogeneity of the underlying edge resource [3]. Not only is this crucial for efficient development of AI applications, it is also a critical requirement for transparent resource provisioning, and to enable a seamless continuum of edge resources, where coordination middleware can transparently make trade-offs between fidelity, accuracy and computational costs.

C. Multi-Purpose Self-Adaptive Edge Computers

In contrast to cloud data centers, where the predominant infrastructure unit is the server computer that consolidates large amounts of CPU, RAM, and disk space, edge resources will be much more diverse and provide a variety of computational platforms to support the equally diverse range of use cases. Edge computing enables many different use cases, such as data analytics, visual analytics, cyber-physical systems gateways, or AR/VR support [6]. Highly specialized hardware platforms, such as high-density , embedded AI hardware, or FPGAs, will be consolidated to cohesive multi-purpose edge computers that enable these use cases. However, this extreme heterogeneity also poses several challenges. At smart city scale, it is impossible to manually map software services to the respective platforms. To make efficient use of these platforms, hardware needs to be grouped into cohesive units of multi-purpose edge computers. When provisioning resources for workloads, a provisioning system needs to understand the capabilities of the underlying resource, as well as the effects of scheduling specific workloads to specific platforms on the operational efficiency. For example, scheduling a ML model inferencing task will be both faster and more energy efficient on an embedded AI hardware than on a general purpose computing platform. In mobile scenarios that require portability and reliability, energy efficiency is an additional constraint to consider [33]. Ideally, a cluster middleware learns at runtime how to optimally provision resources for given workloads, learning the trade-offs between energy efficiency, latency, and accuracy. These self-learning and self-adaptive clusters will form the infrastructural unit of the edge computing fabric.

D. Edge Coordination Mechanisms

As we have shown, edge computing introduces new challenges for operations researchers. There is evidence that methods from cloud operations research, for example for energy efficiency, may only have limited applicability for edge computers [33]. Also, it is clear that maintaining a network of highly dispersed multi-purpose edge computer clusters introduces additional management and coordination complexity. We discuss some of the most pressing operations issues for edge intelligence.

1) Multi Tenancy & Isolation: Applications rarely live in isolation, and a utility-based edge computing fabric has to deal with challenges that come with multi tenancy, which is not as straightforward at the edge as it is in the cloud [34]. Powerful server computers in data centers allow us to easily host multiple VM as the unit of isolation. Although researchers have argued that Cloudlets will simply be VM hosts closer at the edge [7], we cannot necessarily assume that all edge resources can provide this level of virtualization. VMs require a lot of resources and not all edge computers are powerful enough to provide fully-fledged VM-based virtualization. Because this introduces a break in the edge/cloud continuum, there are legitimate doubts whether VMs will work as the predominant form of isolation for edge computing. Container-based deployment strategies have been recognized as a more viable solution, as they provide lightweight resource virtualization and isolation [35]. More recently, Unikernels have been gaining traction as an alternative way of developing applications as completely isolated machine images that can run directly on top of a bare-metal hypervisor [36]. However, such Unikernels rely on library operating systems, and are not yet as well understood as containers. Regardless of which technology will dominate, it is clear that multi-tenancy in resource-constrained environments (compared to a cloud data center) is challenging, as the efficient use of resources becomes more difficult. Security is another important factor. Consider a system where autonomous AI agents roam through the fabric to complete specific tasks. It is absolutely crucial that AI agents are safe from outside tampering and isolated from interference.

2) Intelligent Scheduling: The highly heterogeneous nature and massive geographic dispersion of edge resources poses significant challenges for workload scheduling. Multi tenancy in combination with a limited number of constrained resources further exacerbates this problem. In agent-based scenarios, intelligent eviction or suspend/resume strategies that respect the requirements of AI agents will be necessary, and call for sophisticated management mechanisms. Furthermore, latency and energy-aware scheduling techniques become more difficult in hierarchical architectures, where the underlying platform's operational optimizations must be treated as a black box [33].

Effectively scheduling workloads to the edge will require a large number of hard and soft constraints. Performance evaluations of state-of-the-art scheduling approaches (such as the Kubernetes container scheduler), exhibit clear scalability limitations when confronted with a large number of constraints [37]. We discuss these issues in more detail in Section IV.

3) Proximity & Mobility Awareness: In edge intelligence scenarios, both software agents and hardware resources can be dynamic and mobile. Allowing for mobility requires a good understanding of proximity, s.t., communication latencies between nodes can be optimized, and privacy policies can be enforced. Mobility can be particularly challenging for messaging systems that provide message delivery guarantees [27], [28]. Proximity awareness is therefore a fundamental enabling mechanism of mobility in edge computing [29]. In networks, proximity can be distinguished into logical and physical proximity, and both play an important role in edge computing. For example, physical proximity matters when low-range network techniques such as BLE are involved, and logical proximity matters when minimizing round-trip time. Classic techniques as used by Content Delivery Networks (CDN), such as using anycast DNS and latency monitoring, are challenged in highly dynamic environments. Also, simple monitoring techniques, e.g., via network latency, can produce undesirable strain on the network and become difficult to manage [28]. Besides network latency or logical distance, the application's response time should also be factored into proximity. For example, a message broker in close proximity with a that exhibits high response time because of congested queues can behave as if it were much farther a way. We believe that new methods synthesized from, for example, advanced techniques of interest management, as well as runtime application monitoring, will be necessary for enabling proximity and mobility awareness for edge intelligence at scale.

IV. Operationalized Edge Intelligence

Delivering AI applications involves complicated workflows where data scientists and software engineers collaborate to create and deploy ML models underlying the application [25], [38], [39]. Data is curated and explored, feature engineering and training is performed to create ML models, which are subsequently deployed, monitored, and updated during their lifetime. As edge intelligence transitions from individual and isolated prototypes into interconnected production-grade deployments, the ad-hoc way of building and deploying ML models and AI applications will become impractical. Instead, edge AI operations platforms will fully automate the end-to-end process and provide a closed feedback loop between runtime metrics of deployed models and the workflows that create them [25]. Such platforms are already a reality for cloud-based AI workflows as we will show. In the case of edge AI, however, the process becomes much more involved. Heterogeneous resource capabilities, data locality, scalability issues, etc. have to be considered when automating workflows. In this section, we first briefly summarize existing knowledge on AI operationalization. We will then analyze different operationalization requirements for edge intelligence, identify key mechanisms that can facilitate these scenarios, and then discuss various architectural possibilities and arising challenges.

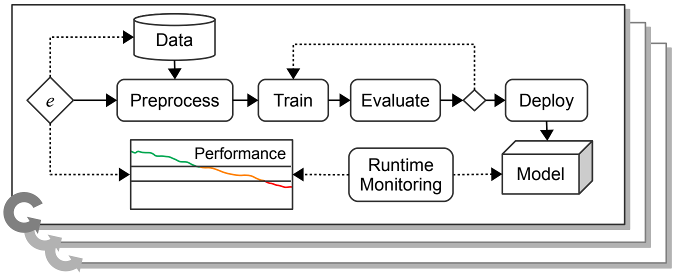

Fig. 3. AI lifecycle pipeline with a rule-based trigger e that monitors available data and runtime performance data to form an automated retraining loop

A. Operations for AI - Managing the AI Lifecycle

The idea of operationalizing AI has become pervasive throughout both industry and research [46]. Software engineering researchers and practitioners have over the past decades developed a variety of DevOps methods for managing the software lifecycle. Continuous integration (CI) and delivery (CD) have become de-facto standards in modern software development processes. Advanced methods such as A/B testing, or continuous experimentation are also employed more and more. Only recently have such life-cycle management techniques and tools for AI applications appeared.

Systems like Uber's Michelangelo [38], IBM Watson Machine Learning [47], ModelOps [25], TensorFlow Extended [39], or ModelHub [48], provide tools and platforms to define and automate AI pipelines, share data and models, and monitor runtime performance of models. Similar to a CI pipeline that, triggered when new code changes arrive, automatically compiles, tests, lints, and then deploys software artifacts, an AI pipeline defines a sequence of steps to manage the AI application lifecycle, from data preprocessing, to model training, to custom steps such as model compression or robustness checking. Runtime performance of models is monitored and used to trigger automated retraining, forming closed continuous retraining loops. Figure 3 shows such a pipeline. Retraining triggers can be as simple as a weekly schedule, or a combination of rules, such as detecting concept drift and monitoring the amount of data available for training. Such pipelines are not as well understood and not as well supported as, for example, CI/CD software pipelines. Moreover, most efforts for AI lifecycle management are focused on both training and deployment in the cloud, and largely neglect edge computing characteristics. In particular, it is unclear how platforms should handle training data that cannot leave the edge, how to reconcile hybrid edge/cloud infrastructure for executing pipelines, how to scale deployment of a large number of ML models to the edge, and how to achieve scalable runtime monitoring of live models. We further explore these issues by analyzing existing and novel cloud/edge workflows for AI lifecycle management.

TABLE I
OVERVIEW OF AI OPERATIONS WORKFLOWS

	Data characteristics	Model characteristics	Enabling technologies	Example use cases
C2C	- Training data is centralized - Massive data sets	- Models are large - Huge number of inferencing requests need to be load balanced	- Scalable learning infrastructure [40] - Data warehousing	- Image search - Recommender systems
C2E	- Training data is centralized - Inferencing data may be sensitive	- Inferencing may need to happen in near-real time - Large number of model deployments - Models run on specialized hardware	- Model compression [41] - Latency/accuracy tradeoff [42] - Distributed inferencing [43] - Transfer learning [44]	- Surveillance systems - Self driving cars - Fieldwork assistants
E2C	- Training data is distributed - Training data may be sensitive	- Models can be centralized - Huge number of inferencing requests need to be load balanced	- Decentralized/federated learning [45]	- Volunteer computing - Novel Smart City use cases
E2E	- Training data is distributed - Training and inferencing data may be sensitive	- Inferencing may need to be near-real time	- Decentralized/federated learning - Distributed inferencing	- Industrial IoT (e.g., predictive maintenance) - Privacy-aware personal assistants - Novel IoT use cases

B. Edge Intelligence Operationalization

Systems like IBM AI OpenScale[5] make use of cloud-based technologies to enable end-to-end AI operationalization in the cloud. Cluster computing systems such as Apache Spark, and clustered deep learning infrastructure operate on data persisted in cloud-based object storages [40], and models are deployed as web services on a cloud-based hosting platform from which they can be easily monitored. Companies are driving efforts to integrate edge resources into this process [49]. Google Cloud IoT Edge[6], Microsoft Azure IoT Edge[7], or Amazon AWS Greengrass[8] leverage edge resources to serve ML models, but data preprocessing, model training, and message brokering is still mainly performed by the cloud. To fully realize end-to-end edge intelligence workflows we need to make full use of edge resources not only for serving models, but for all steps within the AI lifecycle. With this in mind, we identify five different AI lifecycle workflows, from training to serving.

1) Workflows:

a) ***Cloud to Cloud:*** This is the status quo as supported by cloud-based AI platforms such as Microsoft Azure ML, Google Cloud prediction API, or IBM AI OpenScale. Models are trained in centralized training clusters using data aggregated in cloud-based storage silos. Models are then served on cloud resources (e.g., as web services in VMs).

b) ***Cloud to Edge:*** This workflow integrates edge resources as deployment targets. It is useful for use cases where training models requires a lot of computational power, and there are massive amounts of data involved (e.g., training classifiers on ImageNet), but inferencing must be fast (e.g, real-time object detection), or inferencing data may not leave the edge (e.g., patient data). Google, Microsoft, Amazon and others are driving effort with their enterprise-grade edge AI platforms, which are still largely in alpha or beta stages of development.

c) ***Edge to Cloud:*** This workflow makes use of edge resources to train data, and serves models in the cloud. The workflow may be useful for application where training data is massively distributed (such as in mobile computing scenarios [45]), training data may not travel to the cloud, or training needs to be performed close to data sources. Yet, serving models requires either massive scalability, or decentralized inferencing is impractical.

d) ***Edge to Edge:*** In this workflow, both training and serving happens in the edge. For reasons similar to the previous two workflows. Industry 4.0 use cases illustrate this, where sensitive data may not leave the companies premises, and inferencing needs to happen in near real-time at the edge.

e) ***Complex Hierarchical Models:*** It is likely that future scenarios will call for a creative mix of the above mentioned workflows. Some use cases may require base models to be trained on aggregate data in the cloud, then deployed and fine-tuned with data at the edge (e.g. demand forecasting for a retail chain where a model is built on data across all locations, and fine-tuned for specific stores; or personalized diabetes assistants, where anonymized data across patient groups are used to train a base model, that is then refined using the individual patient's data). Localization and context play a big role here, and a hierarchical edge/cloud architectures can help facilitate this as we will show.

Table I lists data and model characteristics for the basic workflows, specific ML mechanisms that are the key enablers, and some example use cases. Which workflow will apply to a given problem depends on many considerations: What are we training models for? How large are the models? How often are models re-trained? How much data is involved? How long does it take, and is it important that training is fast? Some simple curve fitting models for on-the-fly optimization can be executed in a matter of seconds, whereas training image

[5] https://www.ibm.com/cloud/ai-openscale
[6] https://cloud.google.com/iot-edge/
[7] https://azure.microsoft.com/en-us/services/iot-edge/
[8] https://aws.amazon.com/greengrass/

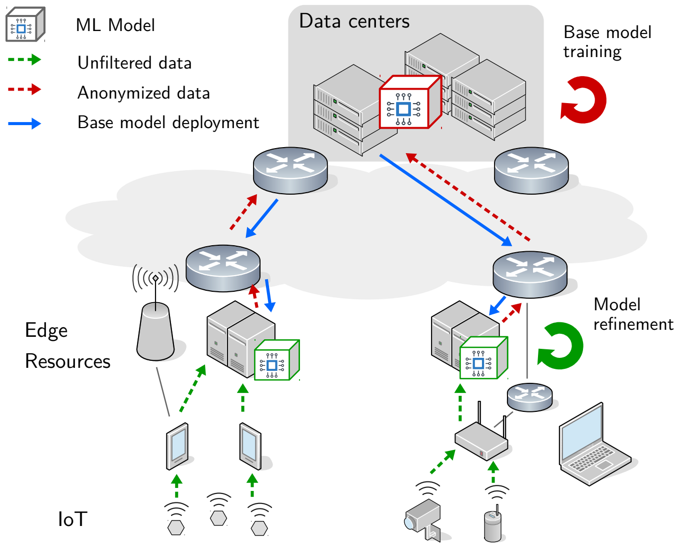

Fig. 4. Edge AI scenarios leveraging hierarchical edge architectures

classifiers may take several hours or even days. How fast does inferencing have to be? Can we tolerate latencies from sending data to the cloud? Is the data used for training or inferencing sensitive?

2) Facilitating Complex Edge AI Scenarios: It is clear why edge computing is an appealing model for advanced AI applications. A hierarchical hybrid cloud and edge computing architecture, as shown in Figure 4, can help facilitate complex edge AI operations scenarios. For example, consider a scenario where dispersed edge and IoT devices produce data that could provide locality-specific insights. Suppose many such premises exist, that share a common knowledge base. For example, a disaster recovery system that uses AI for decision making and forecasting. Any type of decision making will be highly locality specific, yet abstract patterns may be similar across locations. Public cloud-based ML facilities, that are cost efficient and can be provisioned on-demand, can be used to train a base model, that is then refined using transfer learning at on-premises cloudlets that have access to data from local sensors and devices. These cloudlets can use distributed learning to avoid aggregating all data in proximity, and then serve models for low-latency inferencing.

As promising as edge computing is, there are numerous challenges in realizing such an end-to-end architecture. Deploying models to the edge in a generalized way may be difficult (compared to, for example, packaging the ML model as a web service and deploying it to a PaaS like Heroku), as edge resources are very heterogeneous and can't yet be used like classic utility computing. Will the model run on a Raspberry PI or other SBC, or a high-density edge computer cluster, or a private cloud? How will data be stored and accessed from edge resources to make it available in a safe and privacy-aware way for training and inferencing? How and where will continuous training pipelines be managed and executed? How can we deploy models to a large number of edge resources? How does each model scale with different requirements (e.g., user-specific models for personal health assistants). We take a closer look at some of these questions, and discuss possible solutions and open challenges.

C. Scalable provisioning

The massive distribution and heterogeneous nature of edge resources pose several challenges for provisioning software systems in general, and AI applications in particular. As we discussed in Section III, the capabilities of edge computers are much richer and diverse than those of server computers in classical cloud computing. However, edge resources are also more constrained, both in terms of computing power and storage capacity. From an operations perspective, delivering AI at the edge has to include additional goals and constraints, in particular because proximity to data producers and consumers plays such a critical role in edge computing, both to maintain privacy and trust, and enable low-latency data processing.

Suppose a scenario in which edge resources serve ML models to support an application for detailed visual discovery of points of interest in a city via AR. An AR overlay (either via a smartphone or HMD) provides contextual information about objects or things that are in the camera stream. Ideally, a resource hosts ML models (e.g., for object recognition), for the points of interest in its proximity. Locality and context awareness play a key role in making this work. The resource understands that, in its proximity there exist, for example, an art museum, a botanical garden, or a shopping mall, which each have their own ML models for the given domain.

To ensure reliable operations and the efficient use of edge resources, it is necessary that: 1) applications and ML models are deployed in a way to meet certain operations goals (e.g., reduced bandwidth, low latency, etc.), 2) old applications and models are evicted (a sort of garbage collection), and 3) trust and privacy constraints are maintained.

These requirements are similar to those of large-scale IoT-cloud deployments, for which different provisioning systems have been developed [50], [51]. However, these provisioning frameworks do not consider the co-evolution and dependency of AI applications and the underlying ML models. In particular, they assume a classic compile-test-deploy workflow of software, whereas an AI operations pipeline is highly customizable and its steps have different computational needs. If we take into consideration that the edge AI fabric will be used for both building and serving models, operations challenges are two-fold: 1) provisioning resources for AI pipelines at the edge (mapping of capabilities to workloads), and 2) management of deployed AI applications at the edge.

D. Scalable Monitoring

Runtime monitoring of ML model metrics is a fundamental aspect of trusted and operationalized AI [52]. Deployed models are subject to degrading performance caused by concept drift, i.e., when the inferencing data starts to differ from the original training data [23]. AI may exhibit unintended behavior when trained using reinforcement learning techniques. Models may be subject to adversarial attacks, which can be quantified via robustness scores [24]. Automated AI pipelines require timely access to these metrics for evaluating retraining triggers. For cloud-based scenarios, from a systems perspective this monitoring is relatively straight forward. There are two aspects that make monitoring particularly challenging for edge computing: a) some metrics, such as concept drift, are calculated relative to the dataset the model was trained on [23], and b) components that manage pipeline triggers for continuous retraining need real-time access to monitoring data. Because edge resources are highly distributed and may have limited storage capability, it may not be possible to have continued access to the original training data, or the data may have been removed all together to free up storage. Furthermore, depending on how AI pipelines are provisioned to edge resources, it may be difficult to create a communication link between the resource that actually serves the model, and the pipeline platform that evaluates retraining triggers. Moreover, a large number of models may introduce serious scalability challenges. Consider an application where each user of a large user base requires a personalized model that is transfer learned from a base model. Monitoring and deploying each model individually may not be practical or possible.

E. Privacy and Trust

If personal data spaces turn out to be the dominant form of achieving privacy-aware data distribution, new methods for accessing data for training and inferencing purposes are needed. As people continue to gather personal information into their own space, they will be required to give third parties selective access to their data via complex access rules, or provide only anonymized views on their data via privacy-aware routing techniques. Only when people own their own data, the medium they are stored on, and can manage strict access controls, will they be in full control of their personal and sensitive data. As of today, we still put our trust in cloud providers to handle our data, even highly sensitive data such as health related records. Particularly problematic are the increasing number of surveillance cameras throughout public spaces. Although they provide exciting opportunities for visual analytics, situational awareness, or other AI applications, they are fundamentally a privacy concern. Researchers will be challenged to devise methods that reconcile data availability for AI agents, and guarantee data safety and the data owner's privacy.

F. Explainability

As AI continues to dominate further areas of human life, questions of bias, ethics, trust, and safety become more pressing and will continue to challenge our society and the development of AI. Being able to explain the behavior of AI, and understanding how and why specific predictions are made is crucial for achieving ethical and trusted Explainable AI (XAI) [22]. This is particularly the case for domains where human lives are at stake, such such as healthcare, military, judicial, autonomous driving, and autonomous mobile robots. Explainability techniques operate on different layers in the AI stack and stages in the AI workflow. From visualizing the neurons of a neural network that were activated by the given data input [53], up to tracing the data set the model was trained on. All facets of explainability will be more complex in edge intelligence, as data, learning, and inferencing are distributed and decentralized. Explainability in a broader sense also includes data provenance. For example, a feature of modern cloud-based AI platforms is the tracking of a lineage of models and the data they were trained on [25]. This is a reasonable approach for cloud-based workflows, where storage can be easily scaled, but may prove much more challenging in edge computing, where data is decentralized and storage capacity is limited. Generally, as edge AI systems grow more complex, and the development of AI becomes more and more fast paced, researchers will be challenged to devise methods for XAI that become an integral part of AI systems and methodology.

V. Conclusion

Through the increasing interconnectedness of humans and things, and the fast-paced development of edge AI methods and hardware, edge computing has emerged as a promising computational paradigm for delivering transparently immersive experiences and enabling the seamless augmentation of human cognition. In this paper we outlined the role of edge intelligence in the cyber-human evolution, and presented challenges that will confront edge AI systems for several years to come. Specifically, we have shown how edge computing exacerbates the complexity inherent to AI applications and ML workflows, and that new methods are necessary to leverage hierarchical edge/cloud architectures for the AI lifecycle.

For the full end-to-end realization of smart-city scale edge intelligence, there are still many questions that warrant further investigation, in particular concerning ownership and stake of edge computing infrastructure. With respect to stake, we observe that there are three categories of edge intelligence use cases: public (such as smart public spaces), private (personal health assistants (personal), predictive maintenance (corporate)), and intersecting (such as autonomous vehicles). It is unclear who will own the future fabric for edge intelligence, whether utility-based offerings for edge computing will take over as is the case in cloud computing, whether telecommunications will keep up with the development of mobile edge computing, what role governments and the public will play, and how the answers to these questions will impact engineering practices and system architectures.

Acknowledgment

We sincerely thank Waldemar Hummer who has provided us with deep and valuable insights into the current state of enterprise-grade AI operations platforms.

References

[1] K. Panetta, "5 trends emerge in the gartner hype cycle for emerging technologies, 2018," *Gartner*, 2018. [Online]. Available: https://www.gartner.com/smarterwithgartner/5-trends-emerge-in-gartner-hype-cycle-for-emerging-technologies-2018/

[2] M. Satyanarayanan, P. Simoens, Y. Xiao, P. Pillai, Z. Chen, K. Ha, W. Hu, and B. Amos, "Edge analytics in the internet of things," *IEEE Pervasive Computing*, vol. 14, no. 2, pp. 24–31, 2015.

[3] S. Nastic, T. Rausch, O. Scekic, S. Dustdar, M. Gusev, B. Koteska, M. Kostoska, B. Jakimovski, S. Ristov, and R. Prodan, "A serverless real-time data analytics platform for edge computing," *IEEE Internet Computing*, vol. 21, no. 4, pp. 64–71, 2017.

[4] P. Garcia Lopez, A. Montresor, D. Epema, A. Datta, T. Higashino, A. Iamnitchi, M. Barcellos, P. Felber, and E. Riviere, "Edge-centric computing: Vision and challenges," *SIGCOMM Comput. Commun. Rev.*, vol. 45, no. 5, pp. 37–42, 2015.

[5] W. Shi and S. Dustdar, "The promise of edge computing," *Computer*, vol. 49, no. 5, pp. 78–81, 2016.

[6] M. Satyanarayanan, "The emergence of edge computing," *Computer*, vol. 50, no. June, pp. 30–39, 2017.

[7] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The Case for VM-Based Cloudlets in Mobile Computing," *IEEE Pervasive Computing*, vol. 8, no. 4, pp. 14–23, 2009.

[8] G. A. Montes and B. Goertzel, "Distributed, decentralized, and democratized artificial intelligence," *Technological Forecasting and Social Change*, 2018.

[9] D. Ferrucci, E. Brown, J. Chu-Carroll, J. Fan, D. Gondek, A. A. Kalyanpur, A. Lally, J. W. Murdock, E. Nyberg, J. Prager *et al.*, "Building watson: An overview of the deepqa project," *AI magazine*, vol. 31, no. 3, pp. 59–79, 2010.

[10] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot *et al.*, "Mastering the game of go with deep neural networks and tree search," *nature*, vol. 529, no. 7587, p. 484, 2016.

[11] K. Ha, Z. Chen, W. Hu, W. Richter, P. Pillai, and M. Satyanarayanan, "Towards wearable cognitive assistance," in *Proceedings of the 12th Annual International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys '14. ACM, 2014, pp. 68–81.

[12] C. Pennachin and B. Goertzel, *Contemporary Approaches to Artificial General Intelligence*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 1–30.

[13] A. R. Hurson, E. Jean, M. Ongtang, X. Gao, Y. Jiao, and T. E. Potok, "Recent advances in mobile agentoriented applications," *Mobile Intelligence: Mobile Computing and Computational Intelligence*, pp. 106–139, 2010.

[14] M. Satyanarayanan, "From the editor in chief: Augmenting cognition," *IEEE Pervasive Computing*, vol. 3, no. 2, pp. 4–5, 2004.

[15] S. Dustdar, S. Nastić, and O. Šćekić, *Smart Cities: The Internet of Things, People and Systems*. Springer, 2017.

[16] B. Zhou, X. Wang, and X. Tang, "Understanding collective crowd behaviors: Learning a mixture model of dynamic pedestrian-agents," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 2871–2878.

[17] P. Beckman, "Artificial intelligence at the edge: How machine learning at the edge is creating a computing continuum," 2018, research Computing Centre. [Online]. Available: https://www.youtube.com/watch?v=N_k8Uh8Bl0E

[18] J. M. Schleicher, M. Vgler, S. Dustdar, and C. Inzinger, "Enabling a smart city application ecosystem: Requirements and architectural aspects," *IEEE Internet Computing*, vol. 20, no. 2, pp. 58–65, 2016.

[19] M. Feurer, A. Klein, K. Eggensperger, J. Springenberg, M. Blum, and F. Hutter, "Efficient and robust automated machine learning," in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Curran Associates, Inc., 2015, pp. 2962–2970.

[20] T. Li, J. Zhong, J. Liu, W. Wu, and C. Zhang, "Ease.ml: Towards multi-tenant resource sharing for machine learning workloads," *Proc. VLDB Endow.*, vol. 11, no. 5, pp. 607–620, 2018.

[21] C. Yu, B. Karlas, J. Zhong, C. Zhang, and J. Liu, "Multi-device, multi-tenant model selection with gp-ei," *arXiv preprint arXiv:1803.06561*, 2018.

[22] D. Gunning, "Explainable artificial intelligence (xai)," *Defense Advanced Research Projects Agency (DARPA), nd Web*, 2017.

[23] J. Gama, I. Žliobait, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A Survey on Concept Drift Adaptation," *ACM Computing Surveys*, vol. 46, no. 4, pp. 1–37, 2014.

[24] T.-W. Weng, H. Zhang, P.-Y. Chen, J. Yi, D. Su, Y. Gao, C.-J. Hsieh, and L. Daniel, "Evaluating the Robustness of Neural Networks: An Extreme Value Theory Approach," *arXiv e-prints*, p. arXiv:1801.10578, 2018.

[25] W. Hummer, V. Muthusamy, T. Rausch, P. Dube, and K. El Maghraoui, "Modelops: Cloud-based lifecycle management for reliable and trusted ai," in *2019 IEEE International Conference on Cloud Engineering (IC2E'19)*, Jun 2019.

[26] J. Zhang, B. Iannucci, M. Hennessy, K. Gopal, S. Xiao, S. Kumar, D. Pfeffer, B. Aljedia, Y. Ren, M. Griss, S. Rosenberg, J. Cao, and A. Rowe, "Sensor data as a service – a federated platform for mobile data-centric service development and sharing," in *2013 IEEE International Conference on Services Computing*, 2013, pp. 446–453.

[27] V. Muthusamy and H.-A. Jacobsen, "Mobility in publish/subscribe systems," *Mobile Intelligence*, pp. 62–86, 2010.

[28] T. Rausch, S. Nastic, and S. Dustdar, "Emma: Distributed qos-aware mqtt middleware for edge computing applications," in *2018 IEEE International Conference on Cloud Engineering (IC2E)*, 2018, pp. 191–197.

[29] T. Rausch, S. Dustdar, and R. Ranjan, "Osmotic message-oriented middleware for the internet of things," *IEEE Cloud Computing*, vol. 5, no. 2, pp. 17–25, 2018.

[30] D. Burger, "Microsoft unveils project brainwave for real-time ai," *Microsoft Research, Microsoft*, vol. 22, 2017.

[31] Intel, "Intel unveils the intel neural compute stick 2 at intel ai devcon beijing for building smarter ai edge devices," 2018. [Online]. Available: https://newsroom.intel.com/news/intel-unveils-intel-neural-compute-stick-2/

[32] C. Duckett, "Baidu creates kunlun silicon for ai," 2018. [Online]. Available: https://www.zdnet.com/article/baidu-creates-kunlun-silicon-for-ai/

[33] T. Rausch, C. Avasalcai, and S. Dustdar, "Portable energy-aware cluster-based edge computers," in *2018 IEEE/ACM Symposium on Edge Computing (SEC)*, 2018, pp. 260–272.

[34] P. Liu, D. Willis, and S. Banerjee, "Paradrop: Enabling lightweight multi-tenancy at the networks extreme edge," in *2016 IEEE/ACM Symposium on Edge Computing (SEC)*, 2016, pp. 1–13.

[35] R. Morabito, "Virtualization on internet of things edge devices with container technologies: A performance evaluation," *IEEE Access*, vol. 5, pp. 8835–8850, 2017.

[36] A. Madhavapeddy, R. Mortier, C. Rotsos, D. Scott, B. Singh, T. Gazagnaire, S. Smith, S. Hand, and J. Crowcroft, "Unikernels: Library operating systems for the cloud," *SIGPLAN Not.*, vol. 48, no. 4, pp. 461–472, 2013.

[37] H. Deng, "Improving kubernetes scheduler performance," 2016, coreOS Blog. Online. Posted 2016-02-22. Accessed 2019-03-14. [Online]. Available: https://coreos.com/blog/improving-kubernetes-scheduler-performance.html

[38] J. Hermann and M. Del Balso, "Meet michelangelo: Uber's machine learning platform," 2017. [Online]. Available: https://eng.uber.com/michelangelo

[39] D. Baylor, E. Breck, H.-T. Cheng, N. Fiedel, C. Y. Foo, Z. Haque, S. Haykal, M. Ispir, V. Jain, L. Koc, and Others, "Tfx: A tensorflow-based production-scale machine learning platform," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2017, pp. 1387–1395.

[40] B. Bhattacharjee, S. Boag, C. Doshi, P. Dube, B. Herta, V. Ishakian, K. R. Jayaram, R. Khalaf, A. Krishna, Y. B. Li, V. Muthusamy, R. Puri, Y. Ren, F. Rosenberg, S. R. Seelam, Y. Wang, J. M. Zhang, and L. Zhang, "IBM Deep Learning Service," *IBM Journal of Research and Development*, vol. 61, no. 4/5, pp. 10:1–10:11, 2017.

[41] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding," *CoRR*, vol. abs/1510.00149, 2015. [Online]. Available: http://arxiv.org/abs/1510.00149

[42] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *CoRR*, vol. abs/1704.04861, 2017. [Online]. Available: http://arxiv.org/abs/1704.04861

[43] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, M. Mao, M. aurelio Ranzato, A. Senior, P. Tucker, K. Yang, Q. V. Le, and A. Y. Ng, "Large scale distributed deep networks," in *Advances in Neural Information*

Processing Systems 25, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1223–1231.

[44] S. J. Pan, Q. Yang *et al.*, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[45] H. B. McMahan, E. Moore, D. Ramage, and B. A. y Arcas, "Federated learning of deep networks using model averaging," *CoRR*, vol. abs/1602.05629, 2016. [Online]. Available: http://arxiv.org/abs/1602.05629

[46] Gartner, "Integrate DevOps and Artificial Intelligence to Accelerate IT Solution Delivery and Business Value," https://www.gartner.com/doc/3787770/integrate-devops-artificial-intelligence-accelerate, 2017.

[47] IBM Corporation, "IBM Watson Machine Learning," https://developer.ibm.com/clouddataservices/docs/ibm-watson-machine-learning/, 2018.

[48] H. Miao, A. Li, L. S. Davis, and A. Deshpande, "ModelHub: Deep Learning Lifecycle Management," in *2017 IEEE 33rd International Conference on Data Engineering (ICDE)*. IEEE, 2017, pp. 1393–1394.

[49] R. Maguire and R. Schmit, "Hybrid machine learning: From the cloud to the edge," 2018, cloud Next '18. [Online]. Available: https://www.youtube.com/watch?v=M-29naAVmI4

[50] M. Vgler, J. Schleicher, C. Inzinger, S. Nastic, S. Sehic, and S. Dustdar, "Leonore – large-scale provisioning of resource-constrained iot deployments," in *2015 IEEE Symposium on Service-Oriented System Engineering*, 2015, pp. 78–87.

[51] S. Nastic, H. Truong, and S. Dustdar, "A middleware infrastructure for utility-based provisioning of iot cloud systems," in *2016 IEEE/ACM Symposium on Edge Computing (SEC)*, vol. 00, 2016, pp. 28–40.

[52] A. Bifet, G. de Francisci Morales, J. Read, G. Holmes, and B. Pfahringer, "Efficient Online Evaluation of Big Data Stream Classifiers," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '15. New York, NY, USA: ACM, 2015, pp. 59–68.

[53] W. Samek, T. Wiegand, and K. Müller, "Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models," *CoRR*, vol. abs/1708.08296, 2017.