Journal of Reliable Intelligent Environments (2023) 9:119–134
https://doi.org/10.1007/s40860-022-00176-3

ORIGINAL ARTICLE

Making distributed edge machine learning for resource-constrained communities and environments smarter: contexts and challenges

Hong-Linh Truong[1] · **Tram Truong-Huu**[2] · **Tien-Dung Cao**[3]

Received: 15 June 2021 / Accepted: 4 April 2022 / Published online: 6 May 2022

Abstract
The maturity of machine learning (ML) development and the decreasing deployment cost of capable edge devices have proliferated the development and deployment of edge ML solutions for critical IoT-based business applications. The combination of edge computing and ML not only addresses the development cost barrier, but also solves the obstacles due to the lack of powerful cloud data centers. However, not only the edge ML research and development is still at an early stage and requires substantial skills normally missed in resource-constrained communities, but also various infrastructure constraints w.r.t. network reliability and computing power, and business contexts from the resource-constrained environments require different considerations to make edge ML applications context aware through smart and intelligent runtime strategies. In this paper, we analyze representative real-world business scenarios for edge ML solutions and their contexts in resource-constrained communities and environments. We identify and map the key distinguished contexts of distributed edge ML and discuss the impacts of these contexts on data and software components and deployment models. Finally, we present key research areas, how we should approach them, and possible tooling for making edge machine learning solutions smarter in resource-constrained communities and environments.

Keywords Machine learning · Edge computing · Context awareness · Resource-constrained communities and environments

1 Introduction

Finding and developing novel ICT solutions for resource-constrained communities and environments are important activities for sustainable development goals [63]: such solutions would help to democratize the technological gaps in the world and foster economic growth and innovation. Given the abundant evidences demonstrating how machine learning (ML) has changed the landscape of smart solutions in various application domains, there is no doubt that ML will substantially impact societal and business solutions in resource-constrained communities and environments [21,51]. Resource-constrained communities and environments (RCCE), identified and characterized by various studies [10,31], often lack (1) mature technical infrastructure, such as powerful Internet access, network and computing facilities, (2) high skill workforces in cutting-edge ICT technologies, such as AI/ML and cloud engineering, and (3) clear policies and guidelines about data regulation and policy enforcement. Such communities and environments exist not only in least developed countries, but also in countries with highly developed economy. With the maturity of ML development ready for the production of real-world solutions, the lack of powerful data centers and network connectivities, and the substantial cost reduction of powerful edge devices, combining edge computing, distributed ML with edge ML [20,25,29,45,52,66] becomes an important research direction that can provide various ML-based solutions for resource-constrained communities and environments. Furthermore, this helps to make ML solutions more accessible and affordable to communities lacking strong ML workforces.

✉ Hong-Linh Truong
linh.truong@aalto.fi

Tram Truong-Huu
truonghuu.tram@singaporetech.edu.sg

Tien-Dung Cao
dung.cao@ttu.edu.vn

[1] Department of Computer Science, Aalto University, Espoo, Finland

[2] Singapore Institute of Technology (SIT), Singapore, Singapore

[3] School of Engineering, Tan Tao University, Duc Hoa, Vietnam

While powerful cloud-based ML has already attracted a huge amount of research effort, we have not seen the same dedication to studying distributed edge ML (DEML) in RCCE. Our motivation is to understand specific contexts of DEML in RCCE to suggest foundational development and research focuses for context-aware DEML, where we interpret "context aware" as how DEML solutions should be fitted into business, operational and infrastructure contexts, in this paper, RCCE. The need to focus on DEML is due to its potentials for RCCE: DEML naturally fits into RCCE due to the lack of centralized and powerful data centers, while DEML can be achieved via commodity resources for certain classes of applications. However, in our view, the work on distributed edge ML in resource-constrained communities and environments (DEML-RCCE) has several distinguishing characteristics. Various forms of distributed edge ML [28,39] are new, but it is unclear which ones can be adopted for RCCE. For example, solutions based on powerful edge computation and strong network connectivity with low latency (like 5G) will not be feasible in RCCE (or still a long time until 5G will be available for such solutions). From the perspective of human resources, the developer also needs to acquire depth knowledge and engineering skills to work with ML and managing end-to-end ML solutions is not trivial. Motivated by the practical applicability and potential research of DEML-RCCE, it is not enough if we just study only the technical aspects of distributed ML, such as ML models and data, without considering business contexts and infrastructure contexts, such as business requirements and key performance indicators (KPIs) for the implementation of ML solutions. The key reason is that, under various constraints, DEML-RCCE must show direct practical applications (the so-called applied, real-world ML).

In this paper, we will concentrate on the aspects related to distributed ML software components, distributed edge ML tasks, and resource provisioning and data for DEML. Figure 1 outlines the focus of our work. DEML-RCCE develops ML software products based on various requirements. Such requirements are the source for extracting *business context* (denoted by $\mathcal{B}$). On the other hand, the infrastructures in RCCE have many constraints; they are the source for extracting *Infrastructure Context* (denoted by $\mathcal{I}$). Able to understand DEML-RCCE *operational context* (denoted by $\mathcal{O}$), built from business context and infrastructure context, will help us to steer the development of data and software components and to decide suitable deployment models. To this end, we analyze a set of application scenarios, widely seen in different businesses, to investigate business context of DEML-RCCE. Then we identify and focus on some key DEML operational contexts derived from business contexts and infrastructure contexts. One aspect is that DEML-RCCE has to deal with the diversity of constraints and requirements for real-world business, reflecting the elasticity of service models in many aspects of DEML, including resource and data usage/sharing. Another aspect is to have suitable designs for resilience where certain limited performance can be accepted for businesses. Our contributions are:

- Use cases and business context analysis: present elastic business demands and their consequences for technical and scientific requirements.
- Context identification and impact: present key operational contexts for DEML-RCCE and analysis of their impacts on software components and deployments
- Research directions: identify important research focuses for achieving runtime context awareness for DEML-RCCE.

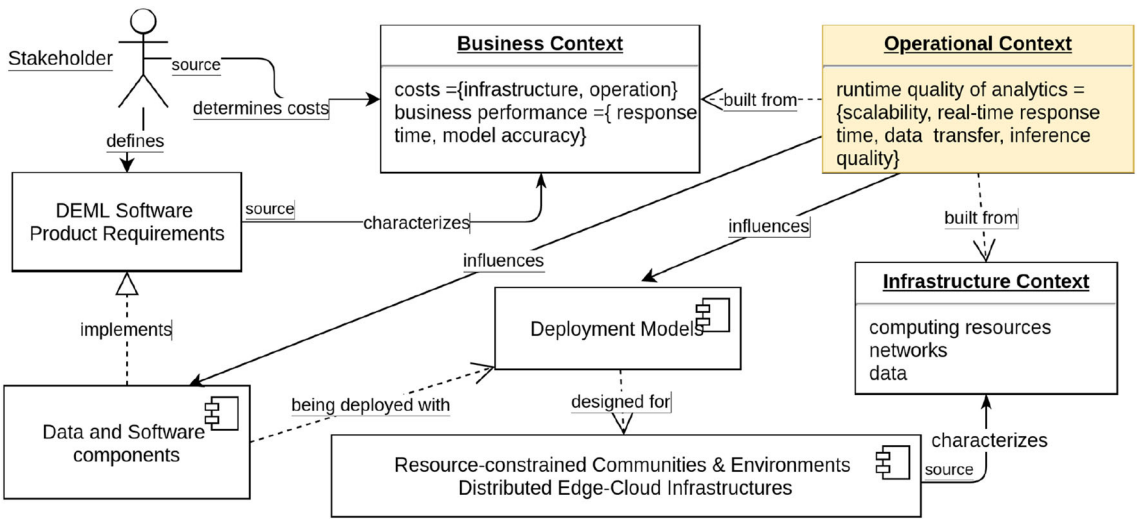

Fig. 1 Overall view of the context

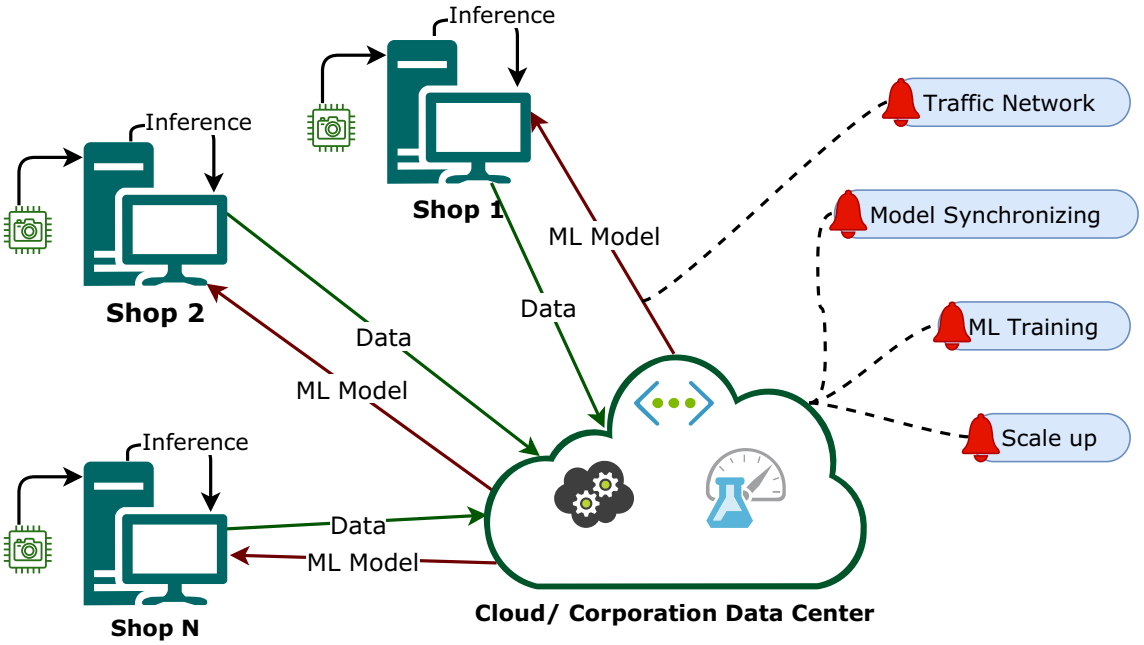

Fig. 2 An SME with multiple business venues sharing the same cloud-based system for customer experience management

The rest of this paper is organized as follows. Section 2 outlines representative application scenarios. Section 3 performs context identification and impact. Section 4 explains in detail research topics. We present further related work in Sect. 5. Section 6 summarizes the paper.

2 Application scenarios and business context

We will focus on the business domain, due to its active and important role that attracts ML development in RCCE. Furthermore, we will focus on applications leveraging IoT data and technologies to work with ML. We investigate three representative application scenarios and their use cases to highlight key aspects of DEML-RCCE. In particular, we concentrate on analyzing *the differences* between DEML in RCCE and other (developed) communities/environments, based on scenarios from Vietnam, to derive business contexts.

2.1 Geographically distributed customer experience management

Application description: Customer experience management is a known business where ML is increasingly used [58]. In particular, let us examine the business venues such as coffee shops, fashion halls, or food and beverage stalls in city/town settings. These businesses are from small and medium enterprises (SMEs), which play a crucial role in RCCE, and the required ML solutions naturally exhibit distributed computing and management. Currently, these businesses increasingly use IoT data from cameras, beacons, and smart loyal cards to verify whether a customer visited their venues in the past or not and to serve the customer better, e.g., offering or suggesting preferred services and dishes. A common scenario is that an SME has multiple venues in cities or towns, e.g., a coffee brand has multiple houses located in different shopping malls. Since all the shops or houses belong to the same SME, they will use the same ML solution for customer experience management. A naive solution is to deploy a cloud-based system for data storage and ML training, whereas edge systems located in these venues are used only for analytics and prediction. To serve customers visiting different venues, network conditions, ML model training and synchronization, and computing resource scaling are important issues. In Fig. 2, we present a business model of this scenario.

Similar use cases: customer shopping prediction [19], advanced image recognition shopping cart [1].

Characteristics in RCCE: Customer behaviors are different. For example, coffee shops in developing countries like Vietnam is a place where customers can spend a few hours for relaxing, talking with friends and collaborators or even studying and working.[1] This leads to two aspects that distinguish the coffee business in RCCE from those in resource-rich communities. First, customer behaviors in a coffee shop are diverse, as customers come to the coffee shops with different objectives. Second, to satisfy different customer demands,

[1] It is worth mentioning that coffee shop businesses are very popular in Vietnam. Many families do their own coffee shop business by converting their garden into a well-designed outlet.

coffee shops serve not only coffee or drinks, but also other types of foods for breakfast, lunch, and dinner, thus expanding their business. However, potential ML solutions face various challenges:

- The update of the customer recognition model is challenging. With a large number of customers, frequently retraining the recognition model incurs a very high *cost* that the business owner may not be able to afford. On the other hand, delaying the retraining could lead to business loss as the customer cannot be recognized by the system.
- ML model synchronization is challenging. Each venue (an edge point) is equipped with a recognition module that performs recognition in an online manner without sending customer data to the central entity that is used only for the retraining model. Using a sole ML model for the entire system will face the scheduling challenge as discussed above, while deploying a separate ML model under a separate ML service for each venue raises the synchronization issue as a customer may visit different venues at different time instants.
- Diversity in the quality of the data captured is a problem. We note that though these venues belong to the same chain/branch, they are operated by different owners who have different financial capabilities to invest in their customer experience management system. This means that the devices used for capturing customer data are heterogeneous and producing data with different quality levels.

Key business contexts: First, $\mathcal{B}_1$ *-low development and operation costs are a major KPI*. The gross revenue obtained from the business when using the system must be much higher than the cost incurred for the system operation and maintenance. Apart from a one-time cost for system deployment and hardware cost (infrastructure cost), a recurring operation cost including Internet subscription and computing resources for model retraining is another factor. Second, $\mathcal{B}_2$*—slow inference response time is tolerated*. For example, the recognition for customer experience management in coffee chains may not be required to provide a fast recognition in the order of seconds, since customers usually spend several minutes to enter the shop, if not hours.

2.2 Cross-business data for ML of an SME network

Application description: SMEs need a mechanism to reuse and share data. For example, the data collected from the increasingly popular cashless payment service in RCCE [46,64] brings a lot of information for businesses such as usual shopping time, location of shopping malls, online or off-line purchasing, and purchased item types. Businesses, such as mobile service providers, shopping malls, and shops in the malls, want to analyze data to improve their service quality. Building a complete centralized dataset of customer behaviors crossing different business domains may not be possible in RCCE due to multiple reasons including the scale of business, operational cost, data privacy, business privacy, and sharing incentive. Thus, a promising solution is that each business owner has its own database and a sharing protocol is required to enable the search and extraction of specific data to be shared. This will also strongly support the elastic business model—join and leave - of SMEs in RCCE. The data marketplace and sharing data have been used increasingly in RCCE (also due to economic and cultural factors) [42]. This leads to cross-business data for ML through a sharing mechanism for different datasets of customer behaviors owned by different SMEs. Figure 3 presents a business model of this scenario.

Similar use cases: patient data analysis [5] and intrusion detection systems [50].

Characteristics in RCCE: Data sharing has to come with a data contract specifying the agreed constraints with respect to the data usage, e.g., how the data consumer will pay the data owner. Similar to developed worlds, data privacy is the first challenge that has great importance as customer information is highly sensitive. The data owners, therefore, may not want their data to leave their premise when sharing the data with third parties. On the other hand, they do not have a suitable infrastructure to enable third-party training code running on their premise (e.g., follow the federated machine learning). Solving this dilemma is difficult in the context of RCCE with limited computing resources at the edge points.

Key business contexts: The key business context is $\mathcal{B}_3$ *- elastic, direct, secure data sharing with privacy assurance for model training* according to marketplace principles. A company has different models so it will train these models with its own capabilities. SMEs want to have elastic, direct solutions to empower each company to share certain data for improving training processes in other companies. Thus, trusted federated service/platforms are needed for facilitating direct data sharing.

2.3 Mobile concrete batching plant optimization

Application description: Due to traffic issues, the lack of strong construction infrastructure, and typical requirements of SMEs in construction, mobile concrete batching plants (CBPs) are common in developing countries. A company owns CBPs, but the software monitoring and controlling CBPs would be provided by another company, since the software and hardware of CBPs are offered by only a few vendors. For each concrete mix (a batch), many parameters must be optimized, such as the amount of water and additives, based on a recipe of concrete batches. Such parameters are

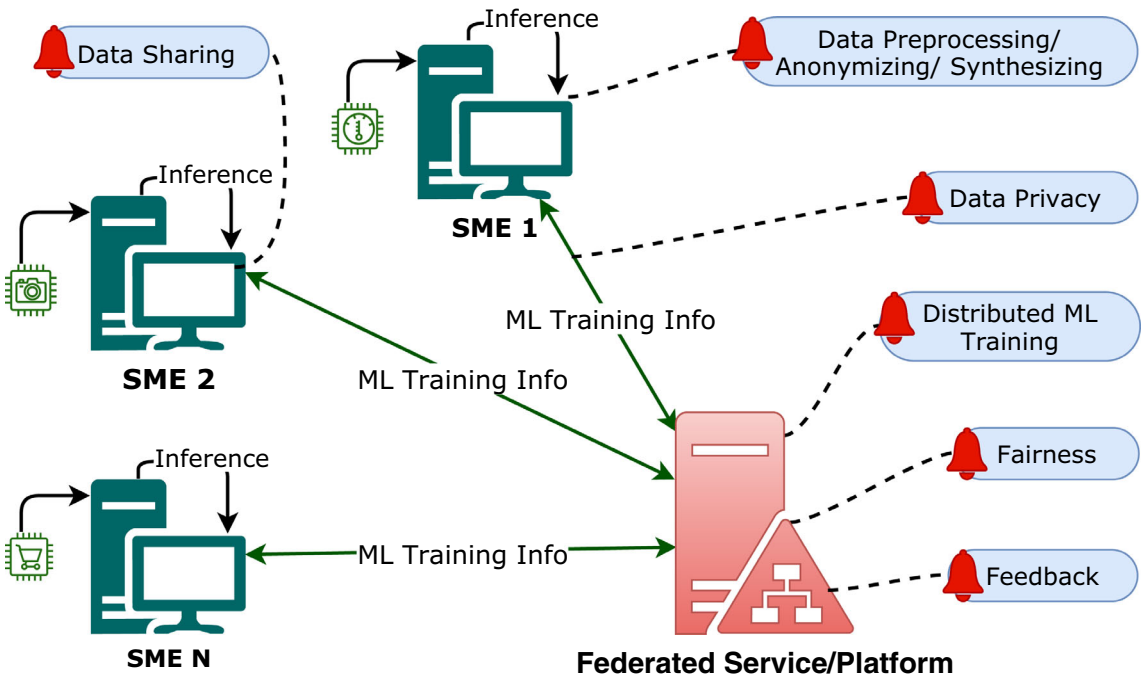

Fig. 3 SMEs with edge resources for inference

important to make sure that the quality of the batch is met, especially with the requirements of slump [4]. If the slump test is met but the water and additives are not optimal, the company incurs additional cost. On the other hand, the concrete batch becomes waste that needs to be dumped, incurring costs and creating environmental problems. Based on a large data set, different ML models can be built and deployed in the edge and cloud. Before running a batch, at the CBP, the operator of a CBP will provide input parameters about the batch and the ML models will infer important parameters. The ML service from the software company relies on a third-party cloud or the company's edge systems for ML serving and the ML models are invoked via unstable 4G networks from the CBP site.

Similar use cases: ML-based OCR in ports and mobile work [27,53].

Characteristics in RCCE: One major difference is that both infrastructures for running edge ML and networks are unreliable. Another aspect is that the cloud part of the ML is very limited. A preferable deployment model is that the entire infrastructure for ML is managed by the SME.

Key business contexts: Since humans play an important role in the control of CBPs, human-in-the-loop is needed. Therefore, it is acceptable that the inference can take a $(\mathcal{B}_4)$ *-longer response time to provide highly accurate* prediction, as any wrong recommendation would lead to a huge cost of waste processing. Overall, the key requirements are centered around the human-in-the-loop, ML accuracy, unreliable networks, and complete edge infrastructures.

3 Operational context identification and impacts

In this section, we first identify important operational contexts for DEML-RCCE and derive the key performance indicators (KPIs) for making DEML-RCCE smarter and intelligent under the influences of such operational contexts. We then analyze the impact of such operational contexts on software components and deployments.

3.1 Infrastructure contexts

Constrained infrastructure for ICT and ML, especially in RCCE, has been discussed intensively. In Table 1, we summarize the key factors related to infrastructures, extracted from business contexts, and their importance and impact on ML solutions. We define the importance and impact of each factor with three levels: `strong`, `medium` and `low`. As we can observe, edge resources and network conditions are the two common factors that have the highest importance and impact on the performance and business outcome for all the use cases. While edge resources are usually constrained, network conditions in RCCE are not always reliable. This requires a proper design of system infrastructure and scheduling algorithms for system operation. Data-related factors, such as data privacy and data sharing, are important for all the scenarios that require customer/client information such as customer experience management. Thus, it is important to design and develop algorithms and services to serve these

Table 1 Summary of application scenarios

Factors in business contexts	Customer experience management	Cross-business data	Concrete batching plant
Distributed data	Strong	Strong	Low
Data sharing	Medium	Strong	Low
Edge resources	Strong	Strong	Strong
Network demands	Strong	Strong	Strong
Use of clouds	Medium	Medium	Low
Response time	Medium	Medium	Medium
Model performance	Medium	Medium	Strong

purposes. From another point of view, we also observe that the scenario of cross-business data sharing is required to take into consideration almost all the issues of DEML. Therefore, solving these issues and enable DEML for these scenarios leads to a huge improvement of business outcomes of SMEs in RCCE.

From the above discussion and the literature on RCCE infrastructure, we identify three main infrastructure contexts that are crucial for DEML-RCCE: (1) constrained compute and storage resources, (2) limited networks, and (3) disparate data quality and robustness.

($\mathcal{I}_1$) -*Constrained computing resources:* It is understood that RCCE will lack the access to powerful computing resources. Furthermore, various SMEs exploring ML solutions will consider the resource cost factor as a key constraint [24]. Therefore, the expected expensive ML training and serving like in powerful companies will not be suitable for SMEs in our context [6,61]. Furthermore, most edge devices are usually resource constrained in terms of memory as well as CPU power. Various enterprises deploying ML consider the resource cost factor as a key constraint [24]. Due to problems of costs, the infrastructure situation will not be changed very soon in RCCE. Although distributed ML can still be deployed at the production level with such an infrastructure of resource-constrained devices, we should not expect to use it to support complex, high-performance ML such as training ImageNet in 1 hour [26]. This leads to the challenging task of selecting appropriate ML models that do not sacrifice performance for the training cost to deploy in DEML systems in RCCE.

($\mathcal{I}_2$) - *Unreliable/weak networks:* RCCE are facing the problem of network bandwidth shortage. On one hand, wired connections cannot reach all areas of the countries and are partially covered by optical networks. On the other hand, cellular networks, mostly with 3G/4G networks, are not strong due to lack of network spectrum or the distribution of base stations. The network usage, therefore, becomes an important factor that needs to be taken into account for DEML in RCCE. Latency incurred due to intermittence and low bandwidth of network connections [15] prevents advanced ML solutions that require many data exchanges.

($\mathcal{I}_3$) -*Disparate data quality:* Due to the infrastructure cost constraints, diverse types of devices/equipment and adequate maintenance processes of ICT lead to many issues of quality and robustness. These issues coupled with limited networks and resource constraints create a huge impact on system performance and quality. For instance, a previous study [49] reported that the quality of devices and networks causes a big problem for GPS data for real-time analytics. The problem of poor data in developing countries has been reported intensively [36]. ML solutions must be built atop processes and techniques dealing with such problems of data quality and reliability.

3.2 Operational contexts

Given the above popular application scenarios in developing worlds, for smarter solutions, technical design and development of DEML must be evaluated carefully considering contexts of RCCE. Figure 4 outlines key operational contexts and their dependencies.

($\mathcal{O}_1$) - *ability under constrained resources and unreliable networks:* Since the amount of data processed increases significantly over time due to the business expansion, a DEML-RCCE solution should be scalable with the increase in data volume and in the number of edge points. However, with a limited budget (($\mathcal{B}_1$)) and weak network infrastructure (($\mathcal{I}_2$)), deployment of a large-scale distributed ML infrastructure and maintenance of its operation is a challenging task. Businesses may not have sufficient capital to invest into the infrastructure while the demand could be intermittent over time.

The cost issue will not be solved very soon in RCCE, thus upgrading to a powerful infrastructure will not happen so early. This consequently needs a proper consideration of (1) ML technologies deployed in DEML-RCCE, and (2) technical approach for scaling computational infrastructures. For the former issue, expected expensive ML training and serving in developed countries or in big companies (e.g., training

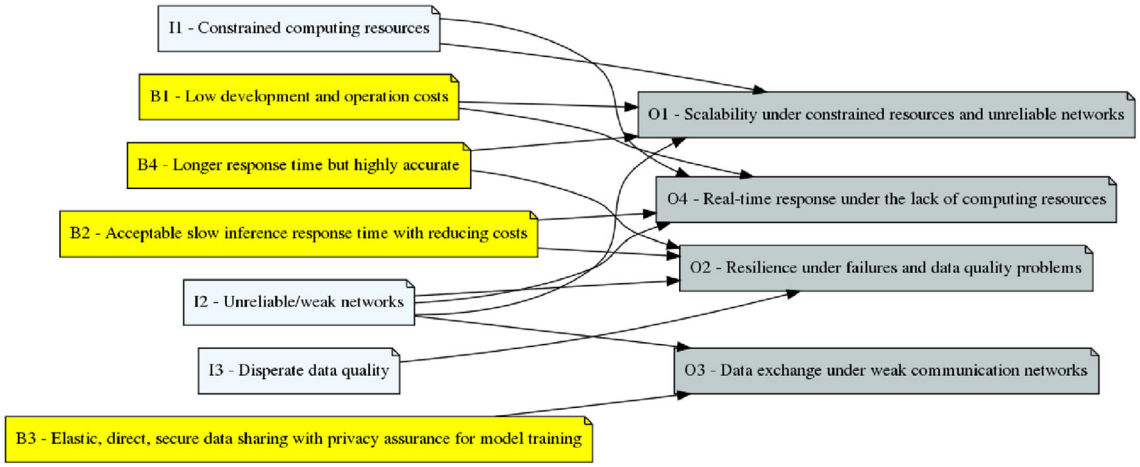

Fig. 4 Operational contexts and their dependencies on other contexts

ImageNet in 1 h [26]) will not be suitable for SMEs in RCCE context [6,61]. Yet, most edge devices are usually resource constrained in terms of memory as well as CPU capacity. This leads to the challenging task of selecting appropriate ML models that do not sacrifice performance for the training cost ($\mathcal{B}_4$) for DEML solutions in RCCE. For the latter, elasticity will be an important technique to achieve the expected performance of the systems while reducing/minimizing the operation cost ($\mathcal{I}_3$).

$\mathcal{O}_2$ *-Resiliency under failures and data quality problems:* Besides dealing with hardware and software failures, DEML-RCCE solutions have to additionally deal with the diversity of data quality and distribution. The heterogeneity of sensor quality could lead to the heterogeneity of IoT data. Due to limited budget ($\mathcal{B}_1$), businesses in RCCE may choose cheap data sensors (e.g., cameras). The heterogeneity of hardware equipment leads to the heterogeneity of data and its distribution ($\mathcal{I}_3$). This will be a challenge for training ML models on those heterogeneous datasets. Resilience against network interruption is also critical, since network condition in developing countries is very poor and may frequently be interrupted. This requires the system to be able to properly operate without any network connections and defer all communication activities till a network connection is available.

$\mathcal{O}_3$ *-Data exchange under weak communication networks:* RCCE are facing the problem of network bandwidth shortage ($\mathcal{I}_2$). Wired connections cannot reach all areas of the countries and partially covered by optical networks, while cellular networks mostly with 3G/4G networks are not strong due to lack of network spectrum or the distribution of base stations. The network usage, therefore, becomes an important factor that needs to be taken into account for DEML-RCCE. Advanced ML solutions that require many data exchanges will face the latency issue due to intermittence and low bandwidth of network connections [15]. To address this constraint, the data sent over the network for the execution of ML solutions should be minimized. Instead of sending raw data, an edge point could perform simple data preprocessing such as extract useful features or transform the data from raw format to a latent representation before sending it over the network. This approach could avoid network interruptions and failures when sending a large amount of data on a poor network condition in developing countries.

$\mathcal{O}_4$ *-Real-time response under the lack of computing resources:* for IoT-based scenarios, it requires the system operates in a real-time manner with minimum latency, with also many dynamic inference situations. However, achieving this design requirement is challenging in the context of RCCE due to, e.g., the lack of computing resources to process data on time ($\mathcal{I}_1$) or the network delay when transferring data or command communication between nodes in the system ($\mathcal{I}_2$). The design of system architecture is therefore important so that delay-sensitive tasks will be executed at the edges near to the users. Data synchronization among edge nodes also has to be scheduled properly to avoid network bottlenecks.

3.3 Impact on DEML-RCCE software components

Each SME can have a customized context-specific strategy, e.g., resilience-centric optimization due to network and resource issues versus efficiency-centric optimization. Therefore, it is possible to have different strategies mapped onto different software layers and components for DEML. However, the feasibility of each software layer and component for DEML in RCCE is strongly influenced by operational contexts. Thus, it is critical to analyze the influences of operational contexts on data, model and software.

Figure 5 presents a generic view on a DEML-RCCE conceptual framework in which we *only focus on key aspects of DEML-RCCE* that we will discuss in the rest of the paper;

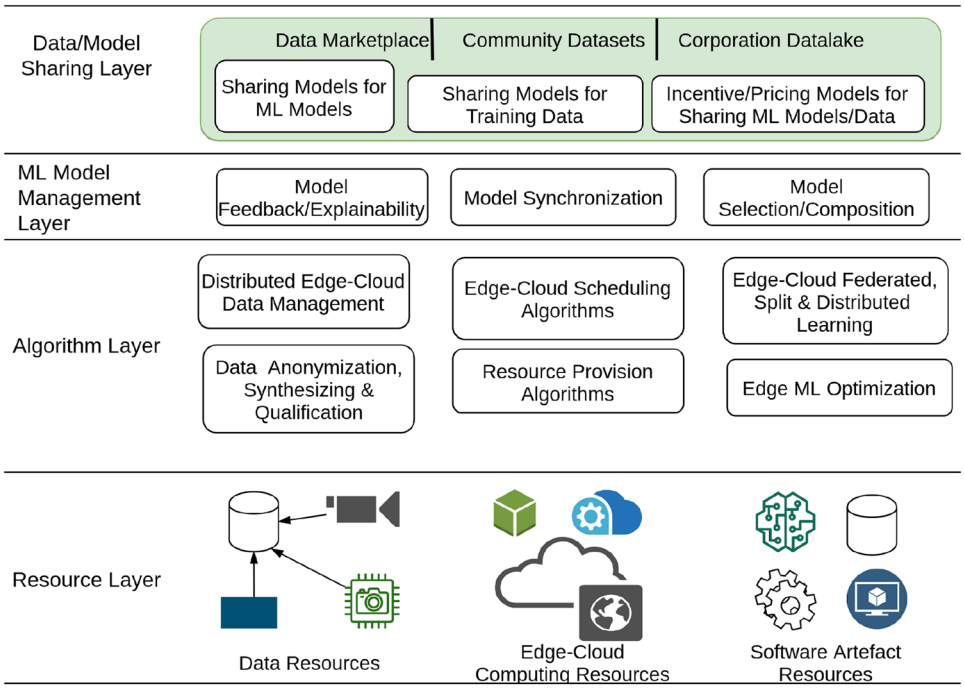

Fig. 5 Key distinguished conceptual components of distributed edge machine learning in resource-constrained communities and environments

other aspects are left for future work. Commonly, we can have four layers, namely the data and model sharing layer, ml model management layer, algorithm layer, and resource layer. This conceptual view reflects the impacts of business and infrastructure contexts on the development and deployment of ML solutions in RCCE.

Data and model sharing layer: The data and model sharing layer provides key aspects for sharing data and models needed by DEML-RCCE.

Impacts: Corporation datalake and community datasets are common ways for data collection, storage, and sharing in developed countries. In the conceptual view, we have data for training and developing ML models and pre-trained ML models that are ready to be deployed. These ways will also be employed in RCCE ($\mathcal{B}_3$). However, SMEs will face a great problem with respect to cost and the ability to master existing frameworks, as most of them are cloud based. Providers like Amazon, Google, and Microsoft offer scalable solutions for datalake that can be used for ML data, while SMEs struggle with human skills and networks to use these solutions. A marketplace is needed for different stakeholders to trade ML data and models. This marketplace should provide pricing models as well as incentive models to encourage different business models. This is a feasible aspect that is important to DEML-RCCE as sharing data/model is a great incentive for different application scenarios, e.g., ML enabling cross-business. It is to be noted that pricing models and incentive models should not focus only on monetary values even though business revenue is the most important factor of SMEs in RCCE. In several cases, the pricing and incentive models are used to guarantee engagements in community cohesion and prevent illegal trading/selling of data in RCCE. Previous marketplaces [13] are relevant but they are not designed for both data and models.

ML model management layer: The ML model management layer describes key aspects related to ML models.

Impacts: Given a distributed and evolving ML system, multiple versions of an ML model have to be stored (e.g., models trained on different datasets, original models downloaded from publicly available sources, and models customized by the companies with different training parameters). With a DEML solution, different edge points may be deployed with different models. Consequently, the selection and composition of existing models to satisfy the requirements of DEML-RCCE is also a challenging task [57]. Another aspect is that DEML-RCCE models can be borrowed or adapted from existing models in developed worlds or from cloud-centric environments. Thus, to make sure such models work well in RCCE infrastructures ($\mathcal{I}$) with RCCE data, we need to spend effort to translate these models.

Algorithm layer: This layer includes different distributed algorithms used for ML model training and serving that consists of data preparation algorithms, resource scheduling algorithms, and model training algorithms.

Impacts: Approaches for distributed ML (training and inferences) are increasingly being investigated [66], but they are not necessarily suitable for RCCE. Given the lack of strong concentrated resources (e.g., cloud-based data centers), loosely distributed computing models, such as embarrassingly distributed batch and workflow processing, could

play a strong role for ML algorithm implementation. Precisely, a federated learning approach [68] could be applied for training a shared model on different distributed datasets [14]. The model owner will interact with the data owners to schedule the training of the model on the respective dataset accordingly. Collaborative learning will be applied when the multiple data owners provide data for a subset of features. There also exist other techniques such as transfer learning or split learning [65] that can support training models while preserving data privacy. We believe that in DEML the resource scheduling algorithms are the key focus to enable the training of models on datasets and distributed computing resources. In RCCE, the scheduling algorithms must take into account the network condition, location of data as well as computing resources. Novel cost models may be needed to minimize the training cost.

Resource layer: The resource layer includes all types of resources needed for training and serving ML models, including data resources, edge-cloud computing resources where the models are trained, and software artefacts resources used to develop and compose ML models and ML pipelines as well as to manage data and models.

Impacts: Data resources consist of publicly available datasets that the companies downloaded, the datasets shared or purchased from their partners, and the data they have collected by themselves. Here, the focus is to utilize sharing models and incentives to enable the availability of data sources. Consider edge-cloud computing resources, which include local resources, edge clouds, or public clouds, it depends on the context of the scenarios discussed above to decide which type of resources can be used. We note that those resources can jointly be used rather than exclusive mutually depending on the operational cost. While local resources can satisfy the immediate computing demands but they may incur high operational costs due to their underutilization during off-peak hours. Public clouds could be cheaper but they may be unavailable due to poor network conditions and budget limits in RCCE. Different edge system topologies can be employed in which a high edge system can process the workload from low edge devices in a timely manner.

3.4 Impact on deployment architectures

The selection of a DEML deployment model as the most suitable for a business model is a challenging task, as it depends not only on the nature of the business model but also on the availability of resources (including both human resources and technical resources). In this section, we present two common deployment architectures, which could be adopted for business models presented in Sect. 2.

3.4.1 Model 1: multi-edge-centralized cloud

A common model is to have multiple edge sites connecting a centralized cloud. Shown in Fig. 6, such an architecture is widely used in distributed ML that supports multiple scenarios of IoT data collection and ML training and inference. In the context of DEML-RCCE, at the regional level, a logically centralized manager is deployed to manage a number of edge points located at the lower level. The manager resides in a regional data center (e.g., a private cloud managed by the system owner) where computational resources can be elastically provisioned depending on the load at the manager. The edge points are essentially access points at the business venues such as coffee shops, food, and beverage stalls, etc. connected to the sensors through different communication protocols such as Ethernet, WiFi, and Bluetooth. The deployment architecture is suitable for customer experience management and cross-business data sharing.

ML role distribution: With the multi-edge-centralized cloud deployment model, the data aggregation and model training can be done at the regional level. This can be applied

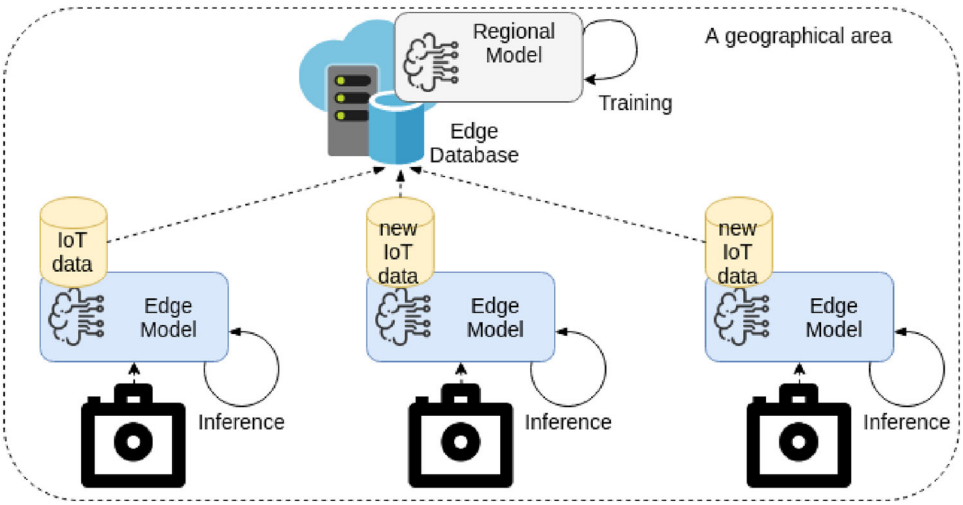

Fig. 6 Multi-edge-centralized cloud deployment model

for two scenarios: an ML solution for a single SME and for a set of cooperative SMEs. First, for a single SME, it can deploy ML components in both edge and cloud. This enables many different deployments known in the state of the art, such as training in the cloud and serving in the edge, distributed training in the edge and cloud, and distributed serving using parallelized ML models. Second, for a set of cooperative SMEs, such as customer experience management, depending on the nature of the business models and data ownership, respective distributed ML techniques will be applied for training models while preserving data privacy, e.g., using horizontal federated learning (or sample-based federated learning) but serving will be individual for each site. The datasets collected at different edge sites share the same feature space but differ in the samples [38]. Every edge site of an SME can maintain an edge model, which may be different from the regional model (e.g., the edge model is trained on the data collected within the coverage of the respective edge site). For cross-business data sharing, vertical federated learning (or feature-based federated learning) [23,41] can be applied. Since the data samples collected from various edge sites have distinct features, the features have to be aggregated to train the model while protecting the data privacy of the edges. It is also possible to do ML serving across edge and cloud by partitioning the model in both edge and cloud. ML at the edge serves only for a particular edge site while ML at the cloud serve the functionalities that need more data or global visibility of the entire system. For instance, in [59], Thangavelu et al. developed a distributed ML for IoT device detection and classification. In their work, each smart home or office is an edge site in which the ML models only monitor and perform detection of IoT devices in the coverage of the edge, and the ML models at the cloud analyze traffic features collected by the edge to perform attack detection, e.g., DDoS originated from the IoT devices of different edges.

Suitability: With the expansion of business, multiple edge points will be deployed. Multiple managers can be instantiated, each managing a subset of edge points. This enables the scalability of the deployment ($\mathcal{O}_1$). This distributed architecture allows ML solutions to be designed in collaborative manner that significantly reduces the data exchange ($\mathcal{O}_3$). For instance, a customer detection and recognition task can be done at the edge points while retraining of the models will be done at the regional manager with more computing resources. The modular design of the ML solution will also enable the resilience of the solution against poor conditions or interruption of the network that connects the edge points and the regional manager ($\mathcal{O}_2$). Depending on the ML solution, the regional manager runs a set of modules at the regional data center; such tasks are either computationally intensive or do not need immediate inputs from the edge points. The manager also has the capacity to push information and instructions to the edge points; for instance, model updating synchronization. It is worth mentioning that processing raw data captured from the sensors at the edge points allows the framework to provide analysis output in a real-time manner ($\mathcal{O}_4$).

In RCCE, 5G and beyond networks are deployed step-by-step, enabling real-time and high connections for connected IoT devices and edge sites. The communication delay between IoT devices/edges to base stations still depends on the location of the base stations decided by the telecom operators. SMEs have to decide the necessity of edge systems based on their individual business models by considering the trade-off between capital expenditure (CAPEX) to own a private edge system and operational expenditure (OPEX) to pay cloud computing resources and network bandwidth consumption for data transmission.

3.4.2 Model 2: edge to multiple edges

Fig. 7 shows the edge-to-multi-edge deployment model. In this model, at various edge sites, ML services are deployed. The ML service only supports inferences for the local input data. At the edge site, data collector is used to collect input data and inference/prediction results, which are pushed to the centralized edge system for training. Possible use cases are concrete batching plant.

ML role distribution: Similar to the multi-edge-centralized cloud model, various learning techniques can be applied to exploit the data collected at different edge sites while protecting data privacy if required. In this deployment, we focus on given edges belonging to the sample deployment of an SME. Therefore, it is possible to have: (1) distributed training but multiple, individual instances of serving in the edge, and (2) distributed training and distributed serving through the partition of ML models across edges. In the first case, it is similar to distributed training in the cloud-edge deployment model, but the serving is only at the edge. In the second case, the modular design of ML solutions and the corresponding ML algorithms are important to guarantee the efficiency of the deployed ML solution in terms of communication and computation overhead [60]. A naive solution to speed up the computation and reduce the workload on edge nodes is to enable parallelism of computation (i.e., model training and inference) on multiple edge nodes. However, significant communication overhead among edges may prevent us to carry out the parallelization of ML models. For example, layer-wise parallelism in [33] allows each layer of a deep learning model to be trained on a separate IoT device but the outputs from computing IoT devices need to be aggregated before dispatch again for the computation of the next layer.

Suitability: This architecture is suitable for SMEs where they do not rely on cloud services for training and model man-

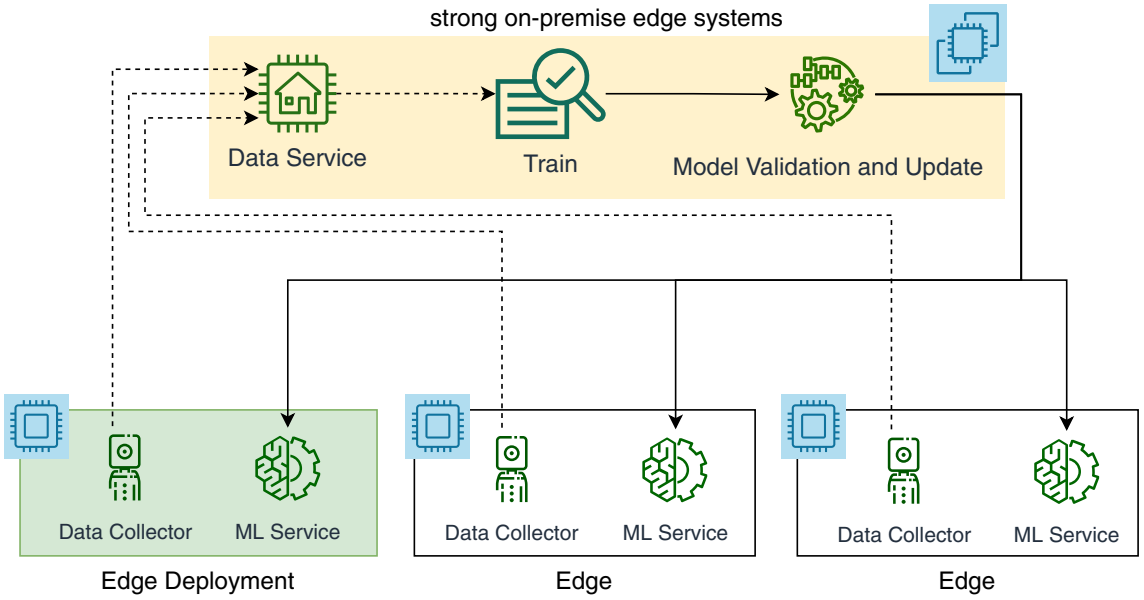

Fig. 7 Edge to multiple edges deployment

agement. It still satisfies the scalability requirement ($\mathcal{O}_1$) as more edge sites can be deployed depending on business expansion. As all the tasks are performed at the edge sites, this architecture will also support resiliency ($\mathcal{O}_2$), minimum data exchange under weak communication networks ($\mathcal{O}_3$), and real-time response ($\mathcal{O}_4$). However, the ML models deployed with this architecture might not be very complex to require powerful infrastructures for training (e.g., no need of a lot of CPU and GPU). Another reason is that cloud is still expensive and especially the connection to external clouds (Google, Amazon) is still hard. In terms of supported software, two approaches could be followed: (1) using different edge site management (such as Kubernetes) and building solutions to connect edge sites, and (2) using linked edge site solutions. For connecting an edge site to another edge site, Kubermatic techniques [2] could be leveraged. The recent development of AI-enabled edge devices such as NVIDIA Jetson Kit allows further flexibility of ML deployment at edges.

4 Key areas for smarter DEML-RCCE

To design smarter DEML-RCCE solutions for given operational contexts, specific characteristics of ML in RCCE need specific research to define suitable activities and interactions to meet the business and technical constraints. Cataloging important use cases and applications in RCCE is an important task. This avoids the situations that ML models (and corresponding ML algorithms/methods) with excellent quality requiring substantial data and computing resources and energy consumption are adopted for unsuitable RCCE infrastructures. In ML development activities, two important aspects are (1) how do developers in RCCE access, reuse, and customize existing ML models, and (2) how do they synchronize knowledge related to reusable ML models from RCCE and the original creator. In ML operation activities, we need a new measurement and interpretation of robustness, reliability, resilience, and elasticity quality in the context of constrained resources and costs in RCCE.

To address this issue, we need to employ techniques from requirement analysis, software engineering, and ML characterization. We suggest (1) characterizingthe stakeholder interactions in DEML-RCCE, e.g., following the recommendation in [11], (2) addressing non-functional requirements issues in DEML-RCCE, e.g., following the approach in [30], and (3) summarizing best practices of ML engineering for DEML-RCCE, such as following the work in [9]. In the scope of this paper, we will focus on non-functional requirements as they are strongly related to the operational contexts. Other important aspects which are not considered in this paper are fairness and ethics, in general FAIR in ML. The main reason is that these aspects are very complex and their discussion deserve a separate work.

4.1 Data synthesizing and data/model marketplace

What: Due to the lack of resources and data, data synthesizing can be applied to create training (labeled) data from captured (unlabeled) data. Additionally, with some incentives, data can be obtained through data marketplaces [13] as described in the data and model sharing layer in Fig. 5. In terms of data quality, solving data problems to prepare high-quality data

for ML is also challenging [54]. It is to be noted that data labeling is a laborious process that requires not only human effort, but also domain-specific knowledge. SMEs in RCCE typically do not have sufficient resources to carry out data collection in a long run.

How: It will be beneficial if DEML-RCCE has tools that generate labeled data with a distribution close enough to real-life scenarios; this would address several issues for RCCE, including the lack of a qualified workforce. The biggest challenge is how to know the distribution of realistic data with a few real samples, particularly caused by the lack of well-prepared sensing and data infrastructures. The process of data synthesizing will be implemented in the edge based on captured data (e.g., create new images). This process will be scheduled and customized based on resource constraints and other aspects such as resilience and network conditions. Different situations could be: fast synthesizing and update for training, lazy synthesizing and lazy update.

Tooling: To some extent, for data synthesizing, existing solutions, such as [47] for tabular datasets and [34] for data modeling, could be useful for DEML-RCCE. In a DEML framework, data collection at remote edge sites can be unlabeled due to issues of sensors, e.g., connection, communication failures, software bugs, or battery draining. Therefore, approaches for labeling unlabeled data using collaborative learning [39], i.e., collaboration among nearby sensors to label a data sample, could be a potential solution, although deployment of such a technique is still challenging in the distributed ML setting due to resource constraints. In terms of data marketplaces, the focus should be on the pricing models and incentives. Pricing models for data marketplaces have been discussed and supported by many industrial providers. Recently, pricing models for ML models [8,16] have been discussed. We need to adapt them for RCCE. To support the business requirements for direct IoT data sharing ($\textcircled{B_3}$) for ML training, the recent work Delta Sharing [3], which fosters secured, direct data sharing integrated into big data/ML code via an intermediate service, could stimulate the development of new protocols and techniques for DEML-RCCE.

4.2 Distributed resilient edge ML

What: A distributed end-to-end edge ML training/serving pipeline consists of multiple edge devices, which are resource constrained and connected with wired or wireless networks. This raises new challenges in guaranteeing the resilience of the pipeline against not only hardware and software failures, but also the uncertainty of data collected (noise, adversarial attacks, or unavailability). This challenge in RCCE is much hard to address due to resource problems.

How: Common techniques with redundant devices/resources are well understood, but are not well supported in RCCE. Especially, the cost of equipment purchase, device operation, and maintenance could reduce the economic profit of companies. Thus, from the research viewpoint, a less costly approach, partially solving the cost problem, is to develop better scheduling algorithms for ML pipelines that take into account all the resource constraints, environment uncertainties, and failures. From a software perspective, the resilience of a DEML-RCCE system must focus on dealing with data missing, network interruption and machine failures. For instance, to deal with data missing, data synthesizing could be a potential solution to generate data at the edge from historical data and send the synthetic data to the analysis module.

Tooling: In terms of software availability, there exist several systems for distributed machine learning platforms [32,71]. However, the question is how to use them in distributed edge-constrained resources, e.g., what would be the equivalence of Apache Spark [70] but for DEML? Recently, more advanced dataflow systems have been developed specifically for distributed ML problems such as Google TensorFlow [7] and MXNet [18]. In particular, TensorFlow has come with a version of constrained resources that could be used to implement ML in resource-constrained infrastructures. However, we will need to focus on the orchestration of such ML code across various edge resources. To support software frameworks in RCCE, it is also important to explore theoretical aspects for (1) network issues, network performance optimization [44], and (2) the Byzantine failures in a distributed machine learning setting [12,17,69] in the context of RCCE. In this view, edge-cloud testbeds for DEML-RCCE setting could be developed, e.g., using containers, to study these problems for RCCE.

4.3 Elasticity and dynamic scheduling

What: For some scenarios (e.g., shop and mobile work), edge points want to have a powerful edge server[2] for computation. This could lead to a high operational cost and resource wasting when they are underutilized, e.g., during the nighttime. For other scenarios (e.g., fields and farms), equipping computing resources at edge points may not be possible due to the environmental constraints. We may have to deal with scenarios where a bust demand incurs and requires a quick response from the system.

How: Scheduling in distributed ML is an important aspect, as models and data are geographically distributed. Nodes host-

[2] "powerful" means that the edge server can have many cores (e.g., see https://www.ibm.com/docs/en/cloud-private/3.2.0?topic=servers-preparing-install-edge-computing). Still, the resource is limited and cannot be scaled elastically as in the cloud.

ing ML relevant components are required to schedule their training with new data samples, update new models, or even schedule the synchronization among them. Furthermore, dynamic scheduling algorithms are important for dealing with constrained resources. First, the dynamic scheduling is carried out in both edge-cloud elastically, but it is also based on networks, data, etc. in RCCE, and the challenging task is dynamic scheduling of training or updating models (trained on new and unseen data samples), while the distributed ML system is still in operation (i.e., prediction or analysis of the data samples collected).

Tooling: There exist various approaches for retraining models to reflect the new data in the training dataset (with new data samples added) that might be adopted for DEML-RCCE. Naive algorithms to retrain the model with the entire dataset are obviously a time-consuming and computing-intensive task, as the number of data samples is increasing over time. New techniques such as lifelong learning [55] could be adopted to efficiently update the models with new data samples so as to minimize the amount of time and resources required. While there is a lack of tools that can be used directly for the dynamic scheduling scenario in RCCE, the research community has developed several research prototypes that can be adopted, such as an open-source ring architecture framework over TensorFlow [56] and a dynamic scheduling and scaling controller for managing distributed deep learning jobs in the Kubernetes cluster [40]. In our view, the combination between containers/Kubernetes scheduling and elasticity for local edges and coordination-aware elasticity across edges should be explored.

5 Further related work

ML surveys and roadmaps: There is no lack of survey and roadmap papers for distributed ML. However, such papers do not discuss DEML-RCCE. The papers [37,48] raise technical issues in edge computing, including architecture design and use cases. However, they do not discuss edge computing specifically in RCCE as well as how edge computing works with distributed ML. The paper [66] surveys distributed ML, but this work only focuses on algorithms, methods, distributed architecture as well as network topology without taking into account the context of RCCE. The work in [39] surveys federated learning with edge networks. It focuses only on one type of ML and does not concentrate on RCCE conditions. The survey in [22] is about resource provisioning challenges in ML in the edge. This is related to our feasibility analysis of the distributed ML infrastructures and the research areas of ML resource provisioning. However, the work just reflects only one aspect of DEML and does not focus on the RCCE context.

ML and developing worlds: The work in [21] discusses ML research in RCCE by defining a roadmap and identifying seven ML areas. However, this work does not analyze in detail to identify the KPIs for ML solutions using edge computing, which is the focus of our paper. Recently, various use cases and papers were presented in two workshops such as [43,62] and they discussed solutions for RCCE. However, the main topics are about the applications and/or algorithms. The discussion of resource constraints has been raised, as this issue is quite obvious. Khan [35] developed concrete but common applications focusing on data feature engineering; however, these applications do not focus on distributed scenario business use cases in the edge. The paper [51] outlines several areas where AI can solve problems in RCCE, but it does not address use cases, feasibility studies, and requirement analysis for DEML.

ML in resource-constrained devices: Many recent works have concentrated on porting ML frameworks from powerful platforms into resource-constrained platforms. The solutions from these works enable technologies for ML with constrained resources in RCCE. One of the common ways is to utilize a cluster of Raspberry Pi to coordinate ML tasks. However, this does not solve the problem of distributed ML where tasks are executed in different locations. The work in [67] outlines issues in federated learning ML in constrained devices. It is related to the infrastructures in DEML-RCCE. However, it is just one aspect related to a specific type of ML and does not consider business and other aspects in DEML. Sharma et al. [57] developed a model selection, named ExpertMatcher, for a remote client to search and use expert ML models in the resource-constrained type. This type of work is just a very specific solution for requirements in RCCE.

6 Conclusions

Advances in ML and edge computing introduce several benefits for applications in RCCE. However, both ML and edge computing are known to have strong demands of data and computing resources and complex software engineering methods that are very challenging issues in RCCE. In this paper, we have analyzed various contexts covering business, infrastructure, cost and operation and their impact on DEML-RCCE designs. We have discussed key impacts for key components on common layers of data, ML models, algorithms, and resources. To develop DEML-RCCE frameworks, researchers and developers can rely on our identified research areas which can provide specific problems, how to approach the problems and which possible tools could be leveraged. Our future work in this line is to explore the issue

of DEML-RCCE software development processes and incentive models for sharing data and ML models in RCCE.

Author Contributions HLT designed and led the work, carried out research and wrote the paper. TTH designed the work, carried out research and wrote the paper. TDC discussed the work and wrote the paper.

Funding Open Access funding provided by Aalto University.

Availability of data and materials Not applicable.

Code Availability Not applicable.

Declarations

Conflict of interests The authors declare that they have no competing interests.

Ethics approval and consent to participate Not applicable.

Consent for publication Not applicable.

References

1. https://www.imagr.co/en. Accessed 13 Feb 2021
2. https://www.kubermatic.com/. Accessed 06 Nov 2021
3. https://delta.io/sharing/. Accessed 06 Nov 2021
4. Concrete slump test. https://en.wikipedia.org/wiki/Concrete_slump_test. Accessed 06 Nov 2021
5. New machine learning method allows hospitals to share patient data privately. https://www.pennmedicine.org/news/news-releases/2020/july/new-machine-learning-method-allows-hospitals-to-share-patient-data-privately. Accessed 28 July 2020
6. The cost of training machines is becoming a problem. https://www.economist.com/technology-quarterly/2020/06/11/the-cost-of-training-machines-is-becoming-a-problem (2020). Accessed 06 Nov 2021
7. Abadi M et al (2016) TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. arXiv:1603.04467 [CoRR]
8. Agarwal A, Dahleh M, Sarkar T (2019) A marketplace for data: an algorithmic solution. In: Proceedings of the 2019 ACM conference on economics and computation, EC '19, pp 701–726. Association for Computing Machinery, New York, NY, USA . https://doi.org/10.1145/3328526.3329589
9. Amershi S, Begel A, Bird C, DeLine R, Gall H, Kamar E, Nagappan N, Nushi B, Zimmermann T (2019) Software engineering for machine learning: a case study. In: 2019 IEEE/ACM 41st international conference on software engineering: software engineering in practice (ICSE-SEIP), pp 291–300. https://doi.org/10.1109/ICSE-SEIP.2019.00042
10. Anderson R.E, Anderson R.J, Borriello G, Kolko B (2012) Designing technology for resource-constrained environments: three approaches to a multidisciplinary capstone sequence. In: 2012 frontiers in education conference proceedings, pp 1–6. https://doi.org/10.1109/FIE.2012.6462501
11. Bhatt U, Andrus M, Weller A, Xiang A (2020) Machine learning explainability for external stakeholders
12. Blanchard P, El Mhamdi EM, Guerraoui R, Stainer J (2017) Machine learning with adversaries: byzantine tolerant gradient descent. In: Guyon I, Luxburg UV, Bengio S, Wallach H, Fergus R, Vishwanathan S, Garnett R (eds) Advances in neural information processing systems, vol 30. Curran Associates Inc, New York, pp 119–129
13. Cao TD, Pham TV, Vu QH, Truong HL, Le DH, Dustdar S (2016) Marsa: a marketplace for realtime human sensing data. ACM Trans Internet Technol 16:3
14. Cao T.D, Truong-Huu T, Tran H, Tran K (2022) A federated deep learning framework for privacy preservation and communication efficiency. J Syst Arch **124**:20
15. Chavula J, Phokeer A, Calandro E (2019) Performance barriers to cloud services in Africa's public sector: a latency perspective. In: Mendy G, Ouya S, Dioum I, Thiaré O (eds) International conference on e-infrastructure and e-services for developing countries. Springer, Springer International Publishing, Porto-Novo, Benin, pp 152–163
16. Chen L, Koutris P, Kumar A (2019) Towards model-based pricing for machine learning in a data marketplace. In: Proceedings of the 2019 international conference on management of data, SIGMOD '19, pp 1535–1552. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3299869.3300078
17. Chen L, Wang H, Charles Z, Papailiopoulos D (2018) DRACO: byzantine-resilient distributed training via redundant gradients. In: Proceedings of the 35th international conference on machine learning, ICML 2018. ICML, Stockholm, Sweden. arXiv:1803.09877
18. Chen T, Li M, Li Y, Lin M, Wang N, Wang M, Xiao T, Xu B, Zhang C, Zhang Z (2015) MXNet: a flexible and efficient machine learning library for heterogeneous distributed systems. NIPS, Montreal
19. Cumby C, Fano A, Ghani R, Krema M (2004) Predicting customer shopping lists from point-of-sale purchase data. In: Proceedings of the tenth ACM SIGKDD international conference on knowledge discovery and data mining, KDD '04, pp 402–409. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/1014052.1014098
20. Daga H, Nicholson P.K, Gavrilovska A, Lugones D (2019) Cartel: a system for collaborative transfer learning at the edge. In: Proceedings of the ACM symposium on cloud computing, SoCC '19, pp 25–37. ACM, Association for Computing Machinery, Santa Cruz, CA, USA. https://doi.org/10.1145/3357223.3362708
21. De-Arteaga M, Herlands W, Neill DB, Dubrawski A (2018) Machine learning for the developing world. ACM Trans Manage Inf Syst 9:2
22. Duc TL, Leiva RG, Casari P, Östberg PO (2019) Machine learning methods for reliable resource provisioning in edge-cloud computing: a survey. ACM Comput Surv 52:5. https://doi.org/10.1145/3341145
23. Feng S, Yu H (2020) Multi-participant multi-class vertical federated learning. arXiv:2001.11154 [CoRR]
24. Frohlich K, Nieminen M, Pinomaa A (2019) Factors influencing the adoption of m-government: perspectives from a namibian marginalised community. In: Zitouni R, Agueh M, Houngue P, Soude H (eds) International conference on e-infrastructure and e-

services for developing countries. Springer, Springer International Publishing, Porto-Novo, Benin
25. Gopinath S, Ghanathe N, Seshadri V, Sharma R (2019) Compiling kb-sized machine learning models to tiny iot devices. In: PLDI. ACM . https://www.microsoft.com/en-us/research/publication/compiling-kb-sized-machine-learning-models-to-constrained-hardware/
26. Goyal P, Dollár P, Girshick R.B, Noordhuis P, Wesolowski L, Kyrola A, Tulloch A, Jia Y, He K (2017) Accurate, large minibatch SGD: training imagenet in 1 hour. arXiv:1706.02677 [CoRR]
27. Ha K, Chen Z, Hu W, Richter W, Pillai P, Satyanarayanan M (2014) Towards wearable cognitive assistance. In: Proceedings of the 12th annual international conference on mobile systems, applications, and services, MobiSys '14, pp 68–81. Association for Computing Machinery, New York, NY, USA.https://doi.org/10.1145/2594368.2594383
28. Han Y, Wang X, Leung V.C.M, Niyato D, Yan X, Chen X (2019) Convergence of edge computing and deep learning: a comprehensive survey. arXiv:1907.08349 [CoRR]
29. Hong R, Chandra A (2019) Dlion: decentralized distributed deep learning in micro-clouds. In: 11th USENIX workshop on hot topics in cloud computing, HotCloud 2019. USENIX, Renton, WA, USA
30. Horkoff J (2019) Non-functional requirements for machine learning: Challenges and new directions. In: 2019 IEEE 27th international requirements engineering conference (RE), pp 386–391. https://doi.org/10.1109/RE.2019.00050
31. Hui J, Toyama K, Pal J, Dillahunt T (2018) Making a living my way: necessity-driven entrepreneurship in resource-constrained communities. Proc ACM Hum Comput Interact 2:CSCW. https://doi.org/10.1145/3274340
32. Imteaj A, Thakker U, Wang S, Li J, Amini MH (2021) A survey on federated learning for resource-constrained iot devices. IEEE Internet Things J. https://doi.org/10.1109/JIOT.2021.3095077
33. Jia Z, Lin S, Qi C.R, Aiken A (2018) Exploring hidden dimensions in parallelizing convolutional neural networks. In: Dy JG, Krause A (eds) Proceedings of the 35th international conference on machine learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10–15, 2018. *Proceedings of Machine Learning Research*, vol 80, pp 2279–2288. PMLR
34. Jälkö J, Lagerspetz E, Haukka J, Tarkoma S, Kaski S, Honkela A (2019) Privacy-preserving data sharing via probabilistic modelling. arXiv:1912.04439
35. Khan MR (2019) Machine learning for the developing world using mobile communication metadata. Ph.D. thesis
36. Kinyondo A, Pelizzo R (2018) Poor quality of data in Africa: what are the issues? Polit Policy 46(6):851–877. https://doi.org/10.1111/polp.12277
37. Li C, Xue Y, Wang J, Zhang W, Li T (2018) Edge-oriented computing paradigms: a survey on architecture design and system management. ACM Comput Surv 51:2. https://doi.org/10.1145/3154815
38. Li Q, Wen Z, He B (2020) Practical federated gradient boosting decision trees. In: The thirty-fourth AAAI conference on artificial intelligence, pp 4642–4649. New York, NY, USA
39. Lim WYB, Luong NC, Hoang DT, Jiao Y, Liang Y, Yang Q, Niyato D, Miao C (2020) Federated learning in mobile edge networks: a comprehensive survey. IEEE Commun Surv Tutor 20:1
40. Lin C, Yeh T, Chou,J (2019)DRAGON: a dynamic scheduling and scaling controller for managing distributed deep learning jobs in kubernetes cluster. In: Proceedings of 9th international conference on cloud computing and services science, CLOSER 2019, pp 569–577. SciTePress, Heraklion, Crete, Greece
41. Liu Y, Zhang X, Wang L (2020) Asymmetrical vertical federated learning. arXiv:2004.07427 [CoRR]
42. Long J, Brindley W (2013) The role of big data and analytics in the developing world. Tech. Rep. 13-0997, Accenture Development Partnerships
43. Madhushani U, Leonard N.E (2020) Distributed learning: sequential decision making in resource-constrained environments
44. Mai L, Hong C, Costa P (2015) Optimizing network performance in distributed machine learning. In: Proceedings of the 7th USENIX conference on hot topics in cloud computing, HotCloud'15, p 2. USENIX Association, Santa Clara, CA
45. Murshed MGS, Murphy C, Hou D, Khan N, Ananthanarayanan G, Hussain F (2022) Machine learning at the network edge: a survey. ACM Comput Surv 54(8):1–37. https://doi.org/10.1145/3469029
46. Pal J, Chandra P, Kameswaran V, Parameshwar A, Joshi S, Johri A (2018) Digital payment and its discontents: Street shops and the Indian government's push for cashless transactions. In: Proceedings of the 2018 CHI conference on human factors in computing systems, CHI '18, pp 1–13. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3173574.3173803
47. Park N, Mohammadi M, Gorde K, Jajodia S, Park H, Kim Y (2018) Data synthesis based on generative adversarial networks. Proc VLDB Endow 11(10):1071–1083
48. Perera C, Qin Y, Estrella JC, Reiff-Marganiec S, Vasilakos AV (2017) Fog computing for sustainable smart cities: a survey. ACM Comput Surv 50:3. https://doi.org/10.1145/3057266
49. Pham TV, Tran QM, Truong L, Dam KH (2019) Smarter big data analytics for traffic applications in developing countries. IET. https://doi.org/10.1049/PBPC025E_ch2
50. Preuveneers D, Rimmer V, Tsingenopoulos I, Spooren J, Joosen W, Ilie-Zudor E (2018) Chained anomaly detection models for federated learning: an intrusion detection case study. Appl Sci 8:12. https://www.mdpi.com/2076-3417/8/12/2663
51. Quinn J, Frías-Martínez V, Subramanian L (2014) Computational sustainability and artificial intelligence in the developing world. AI Mag **35**(3):36–47. http://www.aaai.org/ojs/index.php/aimagazine/article/view/2529
52. Reiter A, Prünster B, Zefferer T (2017) Hybrid mobile edge computing: Unleashing the full potential of edge computing in mobile device use cases. In: Proceedings of the 17th IEEE/ACM international symposium on cluster, cloud and grid computing, CCGrid'17, pp 935–944. IEEE . https://doi.org/10.1109/CCGRID.2017.125
53. Sadlier D, Ferguson P, Zhang D, O'Connor N.E, Lee H (2011) Inspect: integrated surveillance for port container traffic. In: Proceedings of the 19th ACM international conference on multimedia, MM '11, pp 767–768. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/2072298.2072447
54. Sambasivan N, Kapania S, Highfill H, Akrong,D, Paritosh PK, Aroyo LM (2021) "Everyone wants to do the model work, not the data work": Data cascades in high-stakes ai
55. Savitha R, Ambikapathi A, Rajaraman K (2020) Online RBM: growing restricted Boltzmann machine on the fly for unsupervised representation. Appl Soft Comput 92:106278
56. Sergeev A, Balso MD (2018) Horovod: fast and easy distributed deep learning in TensorFlow. arXiv:1802.05799 [CoRR]
57. Sharma V, Vepakomma P, Swedish T, Chang K, Kalpathy-Cramer J, Raskar R (2019) ExpertMatcher: automating ml model selection for users in resource constrained countries
58. Spiess J, T'Joens Y, Dragnea R, Spencer P, Philippart L (2014) Using big data to improve customer experience and business performance. Bell Labs Tech J 18(4):3–17
59. Thangavelu V, Divakaran DM, Sairam R, Bhunia SS, Gurusamy M (2019) Deft: a distributed iot fingerprinting technique. IEEE Internet Things J 6(1):940–952
60. Thomas A, Guo Y, Kim Y, Aksanli B, Kumar A, Rosing TS (2019) Hierarchical and distributed machine learning inference beyond the

edge. In: 2019 IEEE 16th international conference on networking, sensing and control (ICNSC). Banff, AB, Canada, pp 18–23
61. Thompson NC, Greenewald K, Lee K, Manso GF (2021) Deep learning's diminishing returns: the cost of improvement is becoming unsustainable. IEEE Spectrum 58(10):50–55. https://doi.org/10.1109/MSPEC.2021.9563954
62. Trivedi A, Mukherjee S, Tse E, Ewing A, Ferres JL (2019) Risks of using non-verified open data: a case study on using machine learning techniques for predicting pregnancy outcomes in India
63. United Nations: Sustainable development. https://sdgs.un.org/goals. Accessed 06 Nov 2021
64. Vashistha A, Anderson R, Mare S (2019) Examining the use and non-use of mobile payment systems for merchant payments in India. In: Proceedings of the 2nd ACM SIGCAS conference on computing and sustainable societies, COMPASS '19, pp 1–12. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3314344.3332499
65. Vepakomma P, Gupta O, Swedish T, Raskar R (2018) Split learning for health: distributed deep learning without sharing raw patient data. In: Proceedings of ICLR 2018 workshop on AI for social good. ICLR 2018, Vancouver, Canada
66. Verbraeken J, Wolting M, Katzy J, Kloppenburg J, Verbelen T, Rellermeyer JS (2020) A survey on distributed machine learning. ACM Comput Surv 53:2. https://doi.org/10.1145/3377454
67. Wang S, Tuor T, Salonidis T, Leung KK, Makaya C, He T, Chan K (2019) Adaptive federated learning in resource constrained edge computing systems. IEEE J Sel Areas Commun 37(6):1205–1221
68. Yang Q, Liu Y, Chen T, Tong Y (2019) Federated machine learning: concept and applications. ACM Trans Intell Syst Technol 10:2. https://doi.org/10.1145/3298981
69. Yang Z, Gang A, Bajwa WU (2020) Adversary-resilient distributed and decentralized statistical inference and machine learning: an overview of recent advances under the byzantine threat model. IEEE Signal Process Mag 37(3):146–159
70. Zaharia M, Chowdhury M, Das T, Dave A, Ma J, McCauley M, Franklin M.J, Shenker S, Stoica I (2012) Resilient distributed datasets: a fault-tolerant abstraction for in-memory cluster computing. In: Proceedings of the 9th USENIX conference on networked systems design and implementation. ACM, USENIX Association, San Jose, CA, USA
71. Zhang K, Alqahtani S, Demirbas M (2017) A comparison of distributed machine learning platforms. In: 2017 26th international conference on computer communication and networks (ICCCN), pp 1–9. IEEE, Vancouver, BC, Canada