Distributed intelligence on the Edge-to-Cloud Continuum: A systematic literature review

Daniel Rosendo, Alexandru Costan, Patrick Valduriez, Gabriel Antoniu

Distributed Intelligence on the Edge-to-Cloud Continuum: A Systematic Literature Review

Daniel Rosendo[a], Alexandru Costan[a], Patrick Valduriez[b] and Gabriel Antoniu[a]

[a] University of Rennes, Inria, CNRS, IRISA, Rennes, France
[b] University of Montpellier, Inria, CNRS, LIRMM, Montpellier, France

ARTICLE INFO

Keywords:
Edge computing
Distributed Intelligence
Big Data Analytics
Computing Continuum
Reproducibility

ABSTRACT

The explosion of data volumes generated by an increasing number of applications is strongly impacting the evolution of distributed digital infrastructures for data analytics and machine learning (ML). While data analytics used to be mainly performed on cloud infrastructures, the rapid development of IoT infrastructures and the requirements for low-latency, secure processing has motivated the development of edge analytics. Today, to balance various trade-offs, ML-based analytics tends to increasingly leverage an interconnected ecosystem that allows complex applications to be executed on hybrid infrastructures where IoT Edge devices are interconnected to Cloud/HPC systems in what is called the *Computing Continuum*, the *Digital Continuum*, or the *Transcontinuum*.

Enabling learning-based analytics on such complex infrastructures is challenging. The large scale and optimized deployment of learning-based workflows across the Edge-to-Cloud Continuum requires extensive and reproducible experimental analysis of the application execution on representative testbeds. This is necessary to help understand the performance trade-offs that result from combining a variety of learning paradigms and supportive frameworks. A thorough experimental analysis requires the assessment of the impact of multiple factors, such as: model accuracy, training time, network overhead, energy consumption, processing latency, among others.

This review aims at providing a comprehensive vision of the main state-of-the-art libraries and frameworks for machine learning and data analytics available today. It describes the main learning paradigms enabling learning-based analytics on the Edge-to-Cloud Continuum. The main simulation, emulation, deployment systems, and testbeds for experimental research on the Edge-to-Cloud Continuum available today are also surveyed. Furthermore, we analyze how the selected systems provide support for experiment reproducibility. We conclude our review with a detailed discussion of relevant open research challenges and of future directions in this domain such as: holistic understanding of performance; performance optimization of applications;efficient deployment of Artificial Intelligence (AI) workflows on highly heterogeneous infrastructures; and reproducible analysis of experiments on the Computing Continuum.

1. Introduction

The current digital revolution is impacting human beings in the way they live, work, learn, and communicate. This has resulted in impressive progress in many areas such as Cloud Computing, High-Performance Computing (HPC), Artificial Intelligence (AI), Big Data Analytics, and the Internet of Things. Furthermore, new challenging application scenarios are emerging from a variety of domains such as autonomous vehicles, real-time manufacturing, precision agriculture, smart cities, to cite just a few [152, 93].

The explosion of data generated by many applications in the aforementioned areas and the need for real-time analytics and fast decision making has resulted in a shift of the data processing paradigms, as well as of Machine Learning (ML) paradigms, from centralized approaches towards decentralized and multi-tier computing infrastructures and services [89]. Data processing and AI workflows can no longer rely on traditional approaches that send all data to centralized and distant Cloud datacenters for processing or AI model training and inference. Instead, they need to leverage myriads of resources close to the data generation sites (*i.e.*, in the Edge or Fog) in order to promptly extract insights [10] and satisfy the ultra-low latency requirements of applications, while keeping reasonable resource usage and preserving privacy constraints. In practice, to balance contradictory requirements, in many situations it makes sense to weight the respective benefits of centralization and decentralization and make appropriate trade-offs to smartly use the advantages of each type of infrastructure.

This contributes to the emergence of what is called the *Computing Continuum* [49] (or the *Digital Continuum* or the *Transcontinuum*). It seamlessly combines resources and services at the center of the network (*e.g.*, in Cloud datacenters), at its Edge, and *in-transit*, along the data path. Typically, data is first generated and preprocessed (*e.g.*, filtering, basic inference) on Edge devices, while Fog nodes further process partially aggregated data. Then, if required, data is transferred to HPC-enabled Clouds for Big Data analytics, Artificial Intelligence model training, and global simulations.

Due to the complexity incurred by application deployments on such highly distributed and heterogeneous Edge-to-Cloud infrastructures, the Computing Continuum vision remains to be realized in practice. Deploying, analyzing, and optimizing large-scale, real-life applications on such infrastructures requires configuring a myriad of system-specific parameters (*e.g.,* from AI and Big Data systems, applications,

ORCID(s): 0000-0003-1175-8426 (D. Rosendo); 0000-0003-3111-6308 (A. Costan); 0000-0001-6506-7538 (P. Valduriez); 0000-0001-6525-3736 (G. Antoniu)

ingestion systems, among others) and reconciling many requirements or constraints in terms of interoperability, mobility, communication latency, network efficiency, data privacy, and hardware resource consumption (*e.g.,* GPU memory, CPU power, storage size, and others) [156].

Furthermore, enabling intelligence on the Edge-to-Cloud Continuum to allow fast and accurate decision making requires the efficient deployment of complex AI workflows on massively distributed infrastructures composed by heterogeneous resources. Therefore, enabling intelligence on the Computing Continuum requires the reproducible and extensive evaluations of AI workflow deployments exploring the combination of a variety of ML paradigms and frameworks and analyzing their performance trade-offs and impact on performance metrics such as model accuracy, training time, network overhead, energy consumption and application processing latency.

This systematic literature review provides a comprehensive vision of the main state-of-the-art libraries and frameworks for ML and Data Analytics. It also describes the main learning paradigms for enabling intelligence on the Computing Continuum. The main contributions of this paper are:

1. A taxonomy of Data Analytics and AI libraries and frameworks, and ML paradigms that may compose Edge-to-Cloud workflows to enable intelligence on the Computing Continuum.

2. A synthetic presentation of the main systems for simulation, emulation, and deployment, as well as the relevant large scale testbeds for experimental evaluation of complex Edge-to-Cloud workflows.

3. An analysis of how the studies included in our systematic review provide support for experiment reproducibility, an important requirement of the research community that allows scientific claims to be verified by others. We evaluated each article in terms of: (a) access to artifacts; (b) definition of the experimental setup; and (c) access to results.

4. A discussion of the relevant open research challenges and future directions to enable intelligence on the Edge-to-Cloud Continuum, such as: holistic understanding of performance of applications; performance optimization of Edge-to-Cloud workflows; efficient deployment of complex AI workflows on highly heterogeneous infrastructures; and support of the reproducible analysis of Edge-to-Cloud experiments.

The remainder of this paper is organized as follows. First, we compare our work with the existing surveys/reviews and motivate the need for our review in Section 2. In Section 3, we describe the methodology exploited to guide our systematic review. Then, we provide answers to the research questions raised by our methodology in Sections 4 to 9. In Section 4, we present the main frameworks and libraries for ML in the Edge and in the Cloud. In Section 5 we present the main frameworks and libraries for Data Analytics. Next, Section 6 discusses recent efforts on combining ML and Data Analytics across the Edge-to-Cloud Continuum, as well as the main learning paradigms used. Section 7 presents the main systems for simulation, emulation and deployment for experimental research and the relevant large-scale testbeds. Furthermore, it presents how the selected studies provide support for the experiment reproducibility. Finally, Section 8 highlights the major findings and Section 9 discusses the research challenges in this area. Section 10 concludes this review.

2. Related Work and Motivation

Previous surveys and systematic reviews in the context of the Computing Continuum focused on a variety of domains, such as: resource management [18, 94], security and privacy [104, 8], architectures [40, 29, 6, 64], robotics [149], blockchain [138, 58], just to cite a few. The scope of our work is larger: while focusing on distributed intelligence on the continuum, we review articles in the fields of Machine Learning (ML) and Data Analytics (DA) applied on Edge, Cloud, and Edge-to-Cloud environments. Below we discuss how our work compares to recent surveys in these fields.

Machine and Deep Learning across the Edge-to-Cloud Continuum. A recent survey [7] explores evolving computing paradigms such as Edge, Fog, and Cloud highlighting the latest innovations resulted from their fusion with ML. The authors discuss open research challenges such as: scalability, deployment, failure management, hardware heterogeneity, resource management, security, and interoperability. Furthermore, they present future prospects, including: Big Data Analytics for fast data-driven decision making; Artificial Intelligence to enhance resource management, energy management, security, and reliability; Serverless Computing for leveraging the infrastructure scalability and decreasing the application response time, latency, and energy consumption.

A review on ML for data processing and management tasks across the Edge-to-Cloud continuum is presented in [125]. The authors categorize the usage of ML according to the application domain, ML techniques, input data type, and where they belong in the continuum. Besides, they discuss the research trends toward efficient ML on the edge, in particular: the optimization of ML techniques to reduce their power consumption, memory requirement, and computation intensity; efficient hardware for embedded ML; offloading ML tasks among Edge-to-Cloud resources; and collaborative ML training.

In [98], the authors review communication-efficient distributed Machine Learning strategies for the Edge-to-Cloud continuum. They introduce the principles of distributed ML operations and approaches of implementing parallelism and distribution. Furthermore, authors discuss communication inefficiencies in distributed ML on the Edge and the existing communication-efficient processing techniques for training in resource-limited devices. Lastly, they present research directions where further advancements in communication-efficient distributed ML may be made.

A review of deep learning applications such as computer vision, virtual and augmented reality, and natural language

processing running on Edge devices is presented in [30]. Authors discuss edge-only, hybrid edge-cloud and distributed computing approaches to accelerate deep learning training and inference. They summarize the selected articles in terms of architecture, DNN model, application, key metrics, and Edge hardware used. Lastly, they discuss the challenges in deploying deep learning on Edge-to-Cloud environments, such as: management and scheduling of Edge resources, energy consumption, application migration, benchmarks, and privacy. In this research direction, authors [150] describe the methods and architectures to execute deep learning inference and training at the edge. They also discuss open issues regarding the deployment of deep learning at the edge.

An overview of existing Edge Computing systems is presented in [83]. It discusses techniques to support Deep Learning models at the Edge, for example: (1) systems and toolkits: OpenEI, a framework for Edge Intelligence; AWS IoT Greengrass, for ML Inference; Azure IoT Edge; and Cloud IoT Edge; and (2) open source Deep Learning packages: TensorFlow, Caffe2, PyTorch, MXNet, and some distributed Deep Learning models over Cloud and Edge such as DDNN and Neurosurgeon.

The confluence of IoT and AI detailing their potential applications and open issues is discussed in [96]. It presents the recent approaches for deploying DL on resource constrained Edge devices, Fog and Cloud. They also discuss the two main categories of IoT data generation, such as IoT streaming data and IoT Big Data, as well as their requirements for analytics. Lastly, authors highlight the challenges for the successful merging of DL and IoT applications, such as the lack of real-world datasets for IoT applications; the preprocessing of raw data for DL model training; ensuring data security and privacy in IoT applications; and online resource provisioning for IoT analytics; just to cite a few.

Data Analytics across the Edge-to-Cloud Continuum. In [14] the authors provide a review focusing on the efforts of using Big Data Analytics solutions in the Edge-to-Cloud Continuum. They present the relevant Data Analytics platforms (*e.g.*, Hadoop, Flink, Spark, Storm, Nifi, and others) and Machine Learning libraries (*e.g.*, Spark MLlib, Tensorflow, Keras, Scikit-learn, *etc.*) to enable a real-time Big Data pipeline from the Edge to the Cloud. Lastly, authors discuss the following open challenges: interoperability, characterizing smart city applications, and privacy issues.

A survey on IoT Big Data Analytics covering Big Data generation, acquisition, storage, learning, and analytics is presented in [131]. It discusses parallel processing models and engines for the analysis of Big Data such as Spark, Flink, and Storm. Regarding IoT Big Data learning, they present Machine Learning frameworks working on Big Data and processing in parallel such as Spark MLlib, SAMOA, and FlinkML. Lastly, authors highlight open issues related to Machine Learning and Big Data Analytics in IoT.

A review on Edge, Fog, and Cloud computing infrastructures used for IoT Big Data Analytics is presented in [13]. Authors review the combination of DL and Big Data Analytics in the development of smart cities and provide a comparison of deep learning frameworks and libraries; models; and datasets used in smart city applications. Furthermore, they review articles exploiting IoT and DL to develop intelligent applications and services for smart cities and outline the challenges in developing such applications.

In summary, all these related works *focus on specific domains* such as: Machine Learning on the Edge-to-Cloud continuum [7, 125, 98]; Deep Learning mainly focusing on the Edge, but also discussing hybrid Edge-Cloud deployments [30, 150, 83, 96]; and Data Analytics on Edge-to-Cloud environments [14, 131, 13].

Motivation: study the challenges of ML and DA convergence across the Continuum. As opposed to these previous studies, we are interested in the specific issues (and the frameworks that address them) arising at the frontier of DL and ML, as this combination is rapidly gaining traction as a standard for analytics on the continuum. To the best of our knowledge, our literature review is the first to systematically explore the recent efforts and to summarize the existing approaches on applying Machine Learning, Data Analytics, and their combination to enable distributed intelligence on the Edge, Cloud, and Edge-to-Cloud continuum. Furthermore, our review is unique especially from two main perspectives: (1) it discusses relevant open challenges and research opportunities identified after reviewing the articles; and (2) it provides an extensive analysis of the articles in terms of experimental evaluations and validation, allowing to identify: *(i)* the relevant large-scale testbeds; *(ii)* simulation, emulation, and deployment systems; *(iii)* the common ML/DA frameworks and libraries; metrics; the common models/algorithms, datasets, and Edge hardware; *(iv)* the scale of the testbeds used to validate the proposed solutions; and *(v)* their support for reproducibility.

3. Review Methodology

The systematic review methodology leveraged in this work is based on [71, 48]. The Figure 1 illustrates the three main processes of the review, which are: 1) Planning the Review; 2) Conducting the Review; and 3) Reporting the Review. Next, we describe their corresponding activities in detail.

3.1. Planning the Review

In more and more application areas, we are witnessing the emergence of complex workflows that combine computing, analytics and learning. Such application workflows are evolving towards an interconnected ecosystem that often require a hybrid execution infrastructure from IoT devices to Cloud/HPC systems (aka *Computing Continuum*). A holistic understanding of the complex continuum ecosystem is challenging.

3.1.1. Identify the Need for the Review

This systematic review aims to provide a taxonomy of libraries, frameworks, and learning paradigms that compose

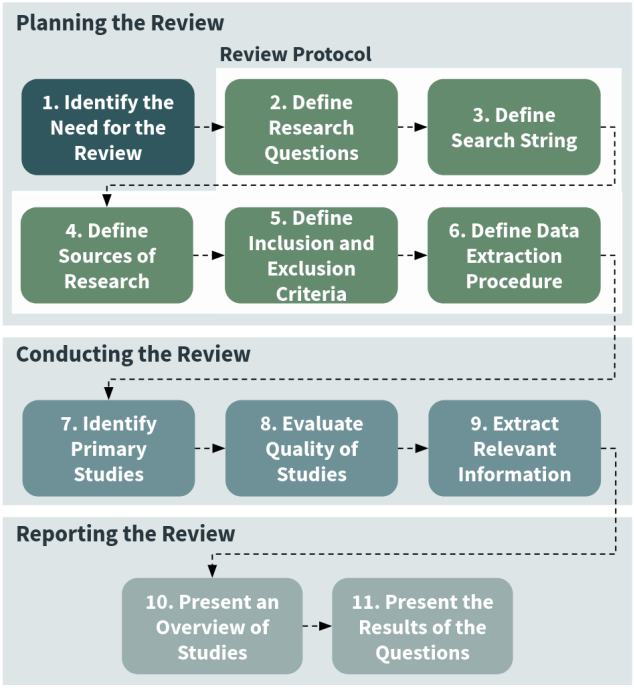


Figure 1: Systematic review methodology.

Edge-to-Cloud workflows to enable intelligent analytics. Furthermore, we highlight systems and testbeds that allow the analysis of such Edge-to-Cloud workflows, as well as the recent efforts to enable the computing continuum vision and the relevant research opportunities.

3.1.2. Define the Research Questions

The objective of the systematic review is to answer the following research questions with a **focus on the Edge-to-Cloud Continuum**:

RQ1. What are the **main state-of-the-art methods** for **Machine Learning** and **Data Analytics**?

RQ2. How are the **existing Machine Learning** and **Data Analytics** approaches **combined** to enable **intelligence**?

RQ3. What are the existing solutions for **experimental research** and how do the selected studies support the **reproducibility** of the experiments?

RQ4. What are the **open challenges** and **research opportunities** in this area?

3.1.3. Define the Search String

The keywords used in the search queries are: IoT, edge, fog, big data, stream processing, learning and intelligence. Therefore, the search string applied in the scientific databases is: "IoT" AND ("edge" OR "fog") AND "big data" AND "stream processing" AND ("learning" OR "intelligence").

3.1.4. Define the Sources of Research

The selected scientific databases are: ScienceDirect, ACM, IEEE Xplore, Springer Link, and Usenix (FAST, NSDI, ATC, HotEdge, and HotCloud).

3.1.5. Define Inclusion and Exclusion Criteria

The search scope of this study is limited to journal and conference articles, magazines and book chapters published between January 2016 and August 2021.

The main characteristics that the articles must present to be included in this systematic review are: (1) help to answer to at least one of the four research questions defined in Subsection 3.1.2; and (2) evaluate existing or propose novel systems, frameworks or architectures enabling intelligence on the Edge, Cloud, or Edge-to-Cloud environments. Articles not respecting these requirements are eliminated.

3.1.6. Define the Data Extraction Procedure

The process of extracting information from the articles consists in filling a form that is designed to answer the research questions of the systematic review. Therefore, the form is structured as follows: title, publication year, scientific database, venue, resume of the contributions, framework/libraries for learning or analytics cited, experimental approach, and open challenges or future works.

3.2. Conducting the Review

The search string applied on the scientific databases returned a total of 1159 articles: 242 from ScienceDirect; 206 from ACM; 325 from IEEE Xplore; 290 from Springer Link; and 96 from Usenix.

3.2.1. Identify the Primary Studies

As a refinement step, we started the screening process, which consists in reading the abstract and conclusions for each article. This refinement step eliminates out of scope articles. Finally, we selected a total of 69 papers for quality evaluation and extraction of relevant information.

3.2.2. Evaluate the Quality of the Studies

The quality evaluation of the selected articles is based on checking if they are related to techniques or approaches that enable intelligent analytics on the Edge-to-Cloud continuum.

3.2.3. Extract the Relevant Information

For each one of the 69 articles selected in 3.2.1 we read the whole paper to extract the relevant information and then fill the form defined in 3.1.6.

3.3. Reporting the Review

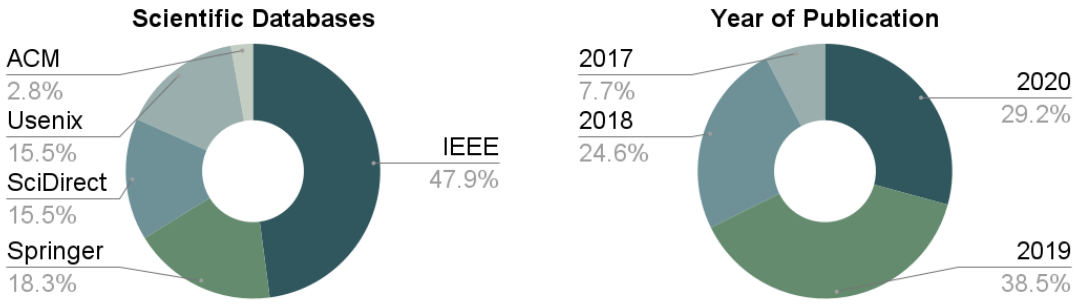
Lastly, two types of reports are issued: general statistics about the studies and answers to the questions raised by our methodology.

3.3.1. Present an Overview of the Studies

Since the form is filled, we generate relevant statistics derived from a global analysis of the articles. Such statistics are aligned to the research questions defined in 3.1.2 and they are presented in the next sections.

3.3.2. Present the Answers to the Research Questions

Finally, according to the relevant information extracted from all articles, we structured the remaining sections of our



(a) Selected articles by scientific database.

(b) Selected articles by year of publication.

Figure 2: Percentage of selected articles per year of publication and per scientific database.

Table 1
Selected articles by area and computing paradigm.

Area	Percentage	Qty.	Computing Paradigm	Percentage	Qty.	Papers
Machine Learning (ML)	35%	24	Edge	58%	15	[169, 77, 86, 167, 102, 75, 162, 166, 30, 62, 42, 80, 50, 76, 43]
			Edge-to-Cloud	42%	9	[144, 31, 128, 170, 97, 85, 57, 67, 119]
Data Analytics (DA)	23%	16	Edge	56%	9	[9, 36, 56, 3, 66, 147, 163, 68, 37]
			Edge-to-Cloud	44%	7	[148, 39, 161, 38, 132, 123, 73]
Combining ML and DA	42%	29	Cloud	36%	10	[4, 88, 118, 145, 109, 101, 84, 12, 99, 157]
			Edge-to-Cloud	64%	19	[151, 131, 96, 83, 165, 114, 72, 5, 160, 164, 60, 44, 111, 52, 127, 129, 106, 74, 120]

review based on the research questions. From the information registered in the form, we grouped, defined taxonomies, and summarized all articles in order to answer the research questions.

Figure 2 presents the percentage of selected papers per scientific database and per year of publication, respectively. We highlight that after the screening process, no article published in 2016 was selected. Table 1 summarizes the selected articles by area and computing paradigm exploited.

4. Machine Learning Methods on the Edge-to-Cloud Continuum

Figure 3 presents the taxonomy of learning methods with a focus on the Edge and across the Edge-to-Cloud Continuum. The distributed training can be achieved across the Continuum or among Edge devices, while the inference is typically done at the Edge, for latency purposes. The Machine Learning frameworks/libraries identified in the articles are presented in Tables 11 (designed for the Cloud) and 12 (designed for the Edge), respectively. Table 2 characterizes the selected articles with respect to the resources exploited in the experimental evaluations, such as: frameworks and libraries; application/task; metrics; hardware; models; and datasets. Table 3 presents a quantitative analysis summarizing Table 2.

4.1. Inference on the Edge

Next, we present the recent efforts to enable inference on resource-limited Edge devices. The following papers focus on hardware and software aspects, as well as, frameworks and algorithms for the efficient inference on Edge devices.

In [42] the authors discuss the potential research directions to enable Edge Intelligence. First, they discuss how AI technologies may help to solve complex problems in Edge Computing, such as: service placement; resource provisioning; network planning; mobility management; among others. Furthermore, they explore issues on performing AI on resource-scarce Edge devices and the recent research efforts to solve problems in this direction, such as: frameworks for model training and inference; accelerating DNN computation on hardware; model compression; asynchronous model aggregation, among others.

In [50] the authors investigate the benefits of using embedded Machine Learning in wearable sensors to increase battery lifetime. Their approach focus on optimizing data generation and transferring and uses Support Vector Machines for classification. Evaluations show that their approach significantly reduces the amount of data transferred and therefore extends the battery lifetime of resource-constrained sensors.

A tree-based algorithm for efficient prediction on IoT devices is proposed in [76]. Named Bonsai, the algorithm was

designed to be fast, accurate, compact and energy-efficient at prediction time. Evaluations show that Bonsai outperforms state-of-the-art algorithms such as kNN, SVM, and single hidden layer NN algorithms in terms of accuracy, model size, inference time, and energy consumption.

A novel framework for collaborative DNN inference on Edge devices is proposed in [80]. Named Edgent, the framework exploits DNN computation partitioning and DNN right-sizing to enable low-latency inference. Authors evaluate Edgent under static and dynamic bandwidth environments and considered performance metrics for model inference such as: accuracy, latency requirements, and throughput. Through experiments on Raspberry Pi (acting as a mobile device), they demonstrate the effectiveness of Edgent towards low-latency Edge intelligence.

In [43] the authors discuss strategies for accelerating DNN inference by partitioning the model between Edge devices. Next, they evaluate the implementation of an offloading system for Deep Learning inference in a Raspberry Pi 3 with the Intel Movidius hardware accelerator. Experimental results considering metrics such as processing latency, data transfer latency, and network bandwidth demonstrate that intelligent offloading may improve the performance when running in resource constrained Edge devices.

In [169] the authors propose three parallelism schemes (All-In-One, Pipeline, and Parallel) to deploy Deep Neural Networks (DNNs) on resource-constrained devices for inference. Each parallelism approach explores the finer granularity of containerizing a DNN model at the edge. Experimental evaluations show that parallelizing a VGG-16 model for inference starts to improve performance as network speed increases.

Since Edge devices are heterogeneous in terms of hardware characteristics and there is a variety of state-of-the-art Machine Learning packages that can be used for inference at the Edge, in [167] authors investigate how such learning packages perform on different Edge devices. They compare TensorFlow, Caffe2, MXNet, PyTorch, and TensorFlow-Lite running two trained CNN-based models (AlexNet as the large-scale model and SqueezeNet and MobileNet as the small-scale models) on Edge devices such as MacBook, FogNode, Jetson TX2, Raspberry Pi, and Nexus 6P. The performance comparison includes metrics such as latency, memory footprint, and energy.

In [62] the authors propose a framework to automatically port a Cloud-based model to a suite of models for Edge devices. Named Mistify, the framework decouples the model design (optimized for accuracy) and the deployment (optimized for resource efficiency) phases. Experimental results show that Mistify reduces the DNN porting time needed to cater to a wide spectrum of Edge deployment scenarios by more than 10 times.

In summary, the articles demonstrated that techniques to minimize the data transfer, reduce the model size, partitioning and offloading the model, and parallelizing the inference among Edge devices, are effective to improve the performance when running in resource constrained Edge devices.

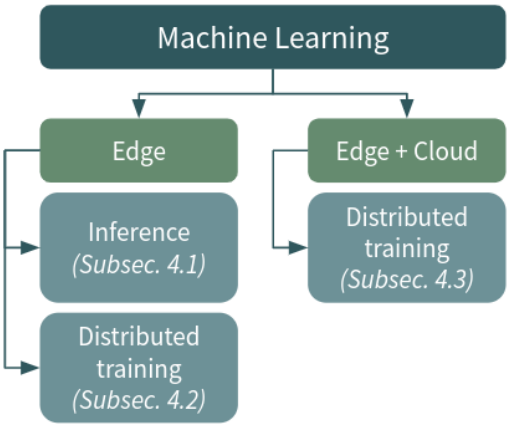


Figure 3: Taxonomy of learning methods on the Edge and Edge-to-Cloud Continuum.

4.2. Distributed Training on the Edge

Next, we present the recent efforts to enable distributed Machine Learning and Deep Learning training on Edge devices. The following papers focus on lightweight Deep Learning models, learning paradigms, collaborative learning systems, and the frameworks and libraries for optimizing Deep Learning on mobile devices.

In [77] the authors propose a system for enabling iterative collaborative processing (ICP) in resource constrained Edge environments with a focus on Machine Learning applications (*e.g.*, model training). The proposed system consists in a central controller that coordinates all the Edge devices (workers). The controller communicates the initial values of the model parameters to all the Edge devices and updates the model parameters at the end of every iteration using the individual model parameters from all the Edge devices. Lastly, it sends the updated parameters to the workers for next iteration. This process repeats until the model parameters have converged.

CLONE [86] is a collaborative learning setting on the Edge built on top of the Federated Learning algorithm and long short-term memory networks. In CLONE, the learning tasks are solved by a group of distributed Edge nodes. Each Edge node trains the neural network model locally based on its private data and uploads asynchronously the parameters to a *Parameter EdgeServer*. The *EdgeServer* aggregates those parameters and sends them back to Edge devices. Experimental results show that, compared to a stand-alone model training, CLONE reduces training time significantly without sacrificing prediction accuracy.

A Lightweight Convolutional Neural Network (L-CNN) is proposed in [102] to enable real-time human identification on a network Edge using fewer resources and preserving the high accuracy of CNNs. In order to enhance performance on Edge, authors propose a hybrid lightweight tracking algorithm named Kerman (Kernelized Kalman filter). Kerman works along with L-CNN to further improve the speed and reliability of feature extraction for human abnormal behavior detection. Experimental results demonstrate that the proposed algorithms can track the humans as objects in real-time

with decent accuracy at a resource consumption affordable by Edge devices.

Besides these novel approaches and systems to enable distributed training on the Edge, some recent efforts focus on understanding the performance of learning on the Edge.

A survey on Deep Learning applied on mobile networking is presented in [165]. Mobile Data Analytics on Edge devices is achieved either through distributed Machine Learning systems such as MLbase, Gaia, TUX [158], and Adam; or, through Deep Learning libraries such as TensorFlow, Theano, PyTorch, and MXNET. Since mobile networks are ever changing, applications should learn and adapt fast to the domain changes. Therefore, authors discuss learning paradigms such as Online Learning, Lifelong Learning, and Transfer Learning. Lastly, a discussion on open source platforms (*e.g.*, TensorFlow, Caffe, and NCNN) that seek to optimize Deep Learning on mobile devices is presented.

An empirical study of on-device Deep Learning for smartphones such as Android devices is presented in [162]. The study includes 21 frameworks based on their popularity (forks and stars on GitHub) in which authors investigate how those frameworks are used in DL applications. Another study [75] explores scenarios where it is advantageous to do training on the Edge. Experimental results show that peak memory footprint, which is crucial for training on Edge devices, can be reduced by checkpointing strategies such as full binomial checkpointing.

As a conclusion, the articles demonstrated that lightweight Deep Learning models may help to reduce the resource usage while preserving the model accuracy. Furthermore, collaborative model training strategies on resource-scarce Edge devices have been shown to be effective to reduce the training time without sacrificing accuracy.

4.3. Distributed training across the Edge-to-Cloud Continuum

The articles presented in this section focus on deploying and distributing the processing of Machine Learning and Deep Learning workloads among Edge and Cloud environments. They propose novel systems and architectures and analyze the performance trade-offs of Cloud only *vs.* Edge-to-Cloud collaborative training.

An overview of challenges and of existing approaches to distributed Machine Learning for IoT applications in the Fog is presented in [119]. The authors start by presenting the main challenges in processing IoT data, such as generation, transmission, and processing. Then, they highlight the challenges related to the execution of Machine Learning techniques in resource constrained Fog devices. Lastly, the authors present existing approaches to distribute intelligence on Fog devices with a focus on distributed processing and information sharing.

[85] provides a review of 5G on traditional and emerging technologies and share their ideas on future research challenges and opportunities. In particular, they exploit how 5G can help the development of Federated Learning. They present the domains impacted by 5G such as Edge computing, security and privacy, artificial intelligence, and database systems.

A decentralized distributed Deep Learning system named DLion is proposed in [67]. DLion builds on top of TensorFlow and implements techniques such as compute capacity-aware batching, adaptive model parameter tuning, and network-aware data exchange features in order to reduce training time, improving model accuracy, and providing system scalability for Deep Learning in micro-clouds. Experiments compare DLion with existing distributed Deep Learning systems such as Gaia and Ako, showing that DLion reaches the target accuracy faster than them. Besides, the compute capacity-aware batching technique implemented in DLion helps to reduce the training time.

The authors of [128] propose a container-based IoT gateway architecture for Ambient Assisted Living (AAL) scenarios to support the deployment of Deep Learning models. Such models are implemented and trained in the Cloud to detect the fall of people and deployed remotely on the Edge gateways to provide predictive analytics. Results show an improvement of the inference time compared to the Cloud-based approach.

Still in the same direction of fall detection, authors of [97] propose a system that detects falls on Edge devices (*e.g.*, mobile phones) by using Boosted Decisions Trees. The proposed approach reduces the amount of data and network traffic sent to the Cloud and presents almost the same detection capabilities as the classification process performed in the Cloud.

Authors of [170] propose SAFACE, a three-layer Edge computing system for face recognition. SAFACE employs Unsupervised Learning which can gradually fine-tune a portion of the face recognition model. The three-layer system consists of: Cloud, to train the CNN model; middle servers, for face recognition, fine-tune pre-trained CNN model, and context-aware scheduling; and Edge, for face detection. Experimental results demonstrate its advantages in improving recognition accuracy and reducing processing latency.

In [57], a novel Edge-Cloud Machine Learning system is proposed. The system combines Edge and Cloud Computing for IoT Data Analytics by taking advantage of Edge nodes to reduce the network traffic and latency for Machine Learning tasks. The results show that using sliding window techniques, the network traffic can be reduced by up to 80% without significant loss of accuracy.

A novel architecture named Edge Cloud Orchestrator (ECO) is proposed in [144]. This architecture aims to orchestrate and manage Machine Learning deployments and execution across distributed layers in both Edge and Cloud. It supports deployment scenarios such as Federated Learning, Transfer Learning, and Staged Model Deployment. Furthermore, it supports Machine Learning engines and algorithms such as Spark MLlib (supports a variety of learning algorithms for classification, regression, clustering, among others), FlinkML (supports SVM, multiple linear regression, k-Nearest neighbors, among others), and TensorFlow (supports SVM, Gradient Boosting Machine, Random Forests, Naive Bayes, k-nearest neighbors, *etc.*).

In [31] the authors explore the use of Synthetic Gradients (SG) for model-parallel training of a Deep Neural Network (DNN) model. This approach distributes the training of the various layers in the Cloud and resource-limited Edge devices. They compare the feasibility of the SG approach with the conventional back propagation method and evaluate its accuracy and convergence speed considering a four-layered, an eight-layered, and a VGG16 model. Results show that the four-layered model presents comparable performance for SG and back propagation, but an accuracy degradation is observed for the VGG16 model using SG. Regarding the convergence speed, using SG, the model learns slower than the back propagation method even while increasing the number of layers in the model.

The articles previously presented demonstrate the benefits of collaborative Edge-to-Cloud training. The main performance improvements of such Edge-to-Cloud approaches refer to: reducing the training time without significant loss of accuracy; reducing the amount of data sent to the Cloud and thus the network traffic; and reducing the end-to-end processing latency of Machine Learning and Deep Learning applications.

4.4. Main takeaways

This section aims to answer the following research question: *What are the main state-of-the-art methods for Machine Learning on the Edge-to-Cloud Continuum?* We organize the existing approaches and the selected studies in two main categories, they are: inference and distributed training on the Edge; and distributed training combining Edge and Cloud.

We highlight that there is not a single library or framework that fits all the needs given the heterogeneous and complex nature of the Edge-to-Cloud Computing Continuum. Therefore, the idea is to provide scientists and engineers a clear vision of the main solutions and existing artifacts and resources (*e.g.*, frameworks/libraries; application/task; metrics; hardware; models; and datasets) so that they can easily identify which ones may be exploited to better attend to their project and research needs.

Next, we summarize the **main limitations of performing Machine Learning over Edge devices**: *(i)* **computing power:** Edge devices are typically limited in terms of accelerator memory (CPU, GPU, TPU) and storage, thus they can not handle large ML/DL models. This can be alleviated either by minimizing the model size (while maintaining accuracy) or by distributing the model across devices [43, 169, 62, 77, 86, 102, 75, 67, 31]; *(ii)* **network communication:** Edge devices are typically interconnected through wireless low-bandwidth and unreliable network links. They may become offline at any time for any reason or the network may be congested. This requires fault-tolerant and communication-efficient approaches for distributed model training [80, 169, 57, 98]; and *(iii)* **energy consumption:** Edge devices are typically battery-powered, thus they can not handle energy-intensive ML/DL tasks. This should be addressed by energy-efficient inference and training techniques to extend the battery lifetime [50, 76].

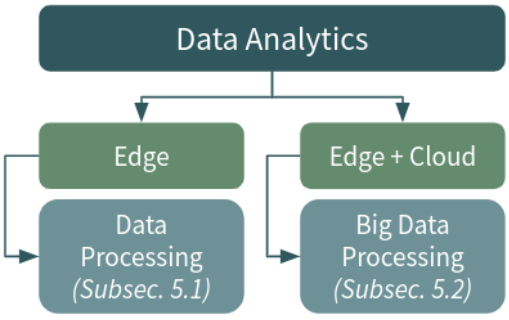


Figure 4: Taxonomy of analytics approaches on the Edge-to-Cloud Continuum.

The articles presented will serve as a basis to identify the recent efforts and also how such libraries and frameworks are being used for: collaborative learning on the Edge and Fog; deploying Neural Networks and performing distributed Machine Learning tasks on resource-constrained devices; how these libraries and frameworks perform on Edge devices; and performance trade-offs for training on the Edge vs. on the Cloud.

5. Data Analytics Methods on the Edge-to-Cloud Continuum

Figure 4 presents the taxonomy of analytics approaches with a focus on the Edge and Edge-to-Cloud Continuum. The Data Analytics frameworks identified in the articles are presented in Tables 13 (designed for the Cloud) and 14 (designed for the Edge), respectively. Table 4 characterizes the selected articles with respect to resources exploited in the experimental evaluations, such as: frameworks; application/task; metrics; and hardware. Table 5 presents quantitative analysis summarizing Table 4. In the next subsections, we present how these frameworks support analytics on the Edge-to-Cloud Continuum.

5.1. Data Processing on the Edge

Typically, IoT applications are latency-sensitive and they generate large amounts of data from sensors and Edge devices. Processing such data efficiently on the Edge of the network to obtain insights and react fast is critical. This section presents the recent efforts and novel systems proposed to distribute data processing among Edge devices to achieve high throughput and low latency.

In [147] the authors focus on using Edge computing for real-time analysis of healthcare systems. They discuss challenges of Edge computing such as: performance, deployment expertise in order to consider various parameters like infrastructure configurations, connectivity, and energy requirements; and data management.

A systematic study of data stream processing and analytics in the Fog considering four dimensions, such as system, data, human, and optimization is presented in [163]. For each dimension, the authors present technical issues and new design challenges. For example, high throughput and low

Table 2: Summary of artifacts, metrics, and hardware exploited in the Machine Learning experiments

Paper	Framework/Library	Application/Task	Metrics	Hardware	Model	Dataset
[167]	TensorFlow, Caffe2, MXNet, PyTorch, and TensorFlow Lite	Inference	inference time; memory footprint; and energy consumption	MacBook Pro, FogNode, Jetson TX2, Raspberry Pi 3 B+, Nexus 6P	AlexNet and SqueezeNet	Not informed.
[169]	TensorFlow	Inference	inference time considering different computation and network conditions	VMs with limited capabilities to emulate IoT devices (physical machine: 2x six-core Intel Xeon 2.40 GHz E5-2620 v3 CPUs, 64 GB RAM)	VGG-16	Not informed.
[77]	DeCaf, MOCHA	Collaborative processing of Support Vector Machines	model convergence speed; number of computations performed per iteration;	Master: Intel(R) Xeon(R) CPU E5-2620 v3 2.40GHz; and Workers: MacBook Pro 2.9 GHz Intel Core i5	Support Vector Machines	Not informed.
[31]	TensorFlow	Training with back propagation vs synthetic gradient	model accuracy and convergence speed	1x Nvidia K40 GPU and Xeon CPU and resource-limited containers to emulate Edge devices (one CPU and 1GB memory)	four-layered model; eight-layer model; and VGG16	MNIST
[86]	CLONE, Tensorflow, Keras, and Scikit-Learn	Model training	training time (from epoch); and evaluation scores including: precision, recall, accuracy, and F-measure	1x Intel FogNode (Parameter EdgeSever) and 2x Intel FodeNodes and 1x Jetson TX2 (Edge nodes: vehicles)	Random Forest (RF); Gradient Boosting Decision Tree (GBDT), and Long Short-Term Memory Networks (LSTMs)	Collected from a large EV company
[67]	DLion, TensorFlow	Model training	training time and model accuracy	4x machines: 2x 24 CPUs; and 2x 8 CPUs	2Conv + 2FC	CIFAR10
[102]	MXNet	Human-object tracking (model training)	performance of human-object detection (performance in FPS, CPU usage, memory usage, Average False Positive Rate, Average False Negative Rate)	Raspberry Pi 3 B; and Tinker Board	Lightweight Convolutional Neural Network (L-CNN)	Pascal Visual Object Classes (VOC) including VOC07 and VOC12
[128]	TensorFlow, Keras	Fall detection (inference)	model performance: inference time, accuracy, and precision; and Edge gateway: CPU, memory, and power consumption	Raspberry Pi 2 B	Long Short-Term Memory Units (LSTM), Gated Recurrent Unit (GRU), Support vector machines (SVM), and k-nearest neighbors (kNN)	SisFall

Paper	Framework/Library	Application/Task	Metrics	Hardware	Model	Dataset
[75]	Tensorflow, Caffe, PyTorch	Model training	memory footprint	ODROID XU4	ResNet	Waggle-based data
[170]	InsightFace, Pytorch, MXNet	face recognition	accuracy, speedup of partial fine-tuning, and throughput	Hisilicon Hi3516CV500 IP Cameras; 1x Intel i7-6700k CPU and Nvidia GTX1080 GPU	MobileNet, Sphere20, ResNet50	private dataset
[57]	Edge-Cloud ML system	human activity recognition	model accuracy, network consumption	simulated edge: VM 2GB of memory and single core CPU; cloud: 1x 6GB of memory and octa-core processor	Feed-Forward NN (FFNN)	MHEALTH Mobile Health
[97]	CoreML	Fall detection (model training)	model accuracy	Apple devices	Boosted Decisions Trees	SisFall
[62]	Mistify, TensorFlow	DNN model porting (Computer Vision and Natural Language Processing)	adaptation time; convergence speed; accuracy; and resource usage	1x Linux server (NVIDIA 2070 GPU); 1x server (NVIDIA P600 GPU); Google Edge TPU; and Samsung S9 smartphone	MobileNet, ResNet50, ResNeXt101, BiDAF, and BERT	ImageNet, Cifar100, and SQuADv1.1
[80]	Edgent, BranchyNet, Chainer	inference	accuracy, latency requirements, inference throughput, network bandwidth	Raspberry Pi and desktop PC (quad-core 3.40 GHz Intel processor; 8 GB RAM)	AlexNet, CIFAR-10	Belgium 4G/LTE bandwidth logs; Oboe
[50]	N.A.	classification	accuracy and battery lifetime	SPHERE	SVM	generated with SPHERE wearable
[76]	N.A.	inference	inference time, energy consumption, accuracy, model size	Arduino	Bonsai, kNN, SVM and single hidden layer neural network	Chars4K; CIFAR10, MNIST, WARD, USPS, Eye, RTWhale, CUReT
[43]	N.A.	inference	processing latency; data transfer latency; network bandwidth	Raspberry Pi 3; Intel Movidius Neural Compute Stick; smartphones; laptops; Nvidia Jetson Tx2	Squeezenet, AlexNet, Inception-v3	Not informed.

Table 3: Quantitative analysis of artifacts, metrics, and hardware exploited in the Machine Learning experiments. Percentages refer to the total number of papers in that domain.

Framework/Library		Metrics		Hardware (Edge)		Processors		Model		Dataset	
TensorFlow	28%	model accuracy	31%	Emulated	26%	CPU	77%	SVM	15%	CIFAR	22%
MXNet	10%	resource usage	16%	Raspberry Pi	22%	GPU	18%	SqueezeNet	9%	SisFall	11%
PyTorch	10%	inference time	16%	NVIDIA Jetson TX2	13%	TPU	5%	AlexNet	9%	MNIST	11%
Keras	7%	convergence speed	13%	Arduino	4%			VGG	9%	Vehicle	6%
Scikit-Learn	7%	energy consumption	13%	Intel FogNode	4%			ResNet	9%	VOC	6%
Caffe2	6%	network	13%	Nexus 6P	4%			MobileNet	9%	RTWhale	6%
TensorFlow Lite	3%			Tinker Board	4%			K-NN	6%	CUReT	6%
DeCaf	3%			ODROID XU4	4%			LSTM	6%	ImageNet	6%
MOCHA	3%			Apple devices	4%			RF	6%	SQuADv1.1	6%
CLONE	3%			Edge TPU	4%			GRU	3%	Chars4K	6%
DLion	3%			Samsung S9	4%			GBDT	3%	WARD	6%
Chainer	3%			SPHERE	4%			L-CNN	3%	USPS	6%
CoreML	3%							GoogleNet	3%	Eye	6%
Mistify	3%							DT	3%		
Edgent	3%							FFNN	3%		
BranchyNet	3%							BDT	3%		
								ANN	3%		

latency in stream processing systems can be achieved by optimizing their configurations such as: the number of bolts in the Storm DAG topology or the micro-batch size of Spark streaming, among others.

In the same direction of achieving high throughput and low latency, a novel stream processing engine focused on the Edge named EdgeWise is proposed in [56]. The idea behind EdgeWise is the use of a congestion-aware scheduler and a fixed-size worker pool to improve throughput and latency. The authors compare EdgeWise with Storm deployed on a cluster of up to 8 Raspberry Pi nodes and they observe that EdgeWise reports up to 3 times improvement in throughput while keeping latency low.

[37] proposes an approach to enable distributed data processing within a cluster of Edge devices. The proposed approach extends Apache NiFi core functionality to include three custom processors such as CaptureVideo, DetectFaces, and RecogniseFaces. Experiments show that the proposed approach has the potential to outperform the Cloud-enabled setup.

A novel distributed architecture that extends Apache NiFi to enable stream data processing at the Edge of the IoT network is presented in [36]. Edge Cluster Stream Processing (ECStream) allows time-constrained data-intensive applications to be entirely deployed and executed at the Edge and it is based on a task parallelism model where atomic tasks are offloaded to peer Edge devices, rather than the full workflow.

An Edge Intelligence framework for building service-oriented IoT is proposed in [68]. The framework allows developers to build stream processing capabilities on Edge server devices and use local streaming analytics to make IoT applications smart. Through annotation based programming primitives developers can design their local intelligent capabilities. The authors compare the latency of activity recognition engine implementations running on an Edge server and on the Cloud. Experiments show that the proposed framework can improve performance without degrading the recognition accuracy.

A new Fog platform for data stream analytics in IoT is proposed in [3]. It aims to exploit the computational capacity of Fog devices to process and analyze data without requiring a frequent use of Cloud resources. Experimental evaluations show that the proposed system can analyze data streams with low processing delay and low network utilization.

In [66] the authors propose Fed4Edge, a system that enables the coordination of resources available in Edge devices to process query pipelines in a collaborative way. Fed4Edge uses RDF Stream Processing (RSP) engines as autonomous processing agents. Large scale evaluations on a cluster of Raspberry Pi show that the scalability can be significantly improved by adding more Edge devices to a network of processing nodes.

A model synchronization mechanism for distributed and stateful data analytics named SCEDA is proposed in [9]. The authors use Reinforcement Learning to make dynamic scheduling decisions by learning individual network connectivity trends of Edge nodes as well as the significance of their updates. The proposed approach tackles the concept drift and connectivity issues in Edge data analytics to minimize its accuracy handicap without losing its timeliness benefits. Experimental results show that SCEDA can achieve a comparable level of accuracy as core data analytics.

In summary, the articles presented exploit the computational capacity of Edge devices to process and analyze data in a distributed way. The proposed approaches contribute to allow data-intensive and latency-sensitive applications to be entirely processed on Edge devices.

5.2. Big Data Processing across the Edge-to-Cloud

The articles presented in this section focus on novel architectures and frameworks exploiting collaborative Edge-to-Cloud data processing for enabling real-time data analytics. The articles also aim to analyze the performance trade-offs of Cloud only *vs.* Edge-to-Cloud collaborative data analytics.

In [38], the authors propose a novel IoT distributed Stream Processing architecture that distributes the workload among a cluster of Edge devices. The proposed solution extends Apache NiFi with new services to discover and select devices able to perform offloaded tasks according to hardware and software requirements. The evaluation scenario consists of an intelligent surveillance system and the authors compare the performance of a cluster of Edge devices with a Cloud setup. Results show that an Edge cluster of 6 nodes performs up to 5-6 times faster than the Cloud deployments.

Later, the same authors proposed [39] a distributed hierarchical data fusion architecture based on Complex Event Processing technology to handle streaming data. This approach produces timely and accurate results with minimum time delay, as soon as necessary information is generated and collected. The authors compare their solution (distributed hierarchical data fusion) with a traditional one (sending all low-level sensor readings to a central Cloud for analysis) and evaluations show that at lower levels (*e.g.*, Edge and Fog) decisions can be taken 20x and 90x times faster.

In [132], the authors propose a novel framework of collaborative Edge-Cloud processing for enabling live data analytics in wireless IoT networks. They also present potential key enablers for the proposed framework and highlight some of the research directions for Big Data aware collaborative Edge-Cloud processing, such as adaptive learning/prediction algorithms.

In [123], the authors propose a novel framework that allows the deployment and optimization [121] of Big Data Analytics applications on the Edge-to-Cloud continuum. They illustrate and validate the framework with a smart surveillance application composed by data processing frameworks such as Edgent (on the Edge) and Apache Flink and Kafka (on the Cloud). Experimental evaluations exploit the performance trade-offs of Cloud-centric vs. hybrid Edge-Cloud processing approaches to understand how they impact metrics such as latency and throughput of the application.

[161] proposes a novel framework that uses fine-grained stream processing to provide high resource utilization while meeting latency targets. Named Cameo, the framework dy-

namically calculates and propagates priorities of events based on user latency targets. Experiments show that Cameo reduces query latency in single and multi-tenant settings.

Several models, technologies and solutions for medical data processing and analysis are presented in [73]. The authors illustrate examples of case studies and practical solutions composed of health sensor data processed with Kafka and Spark (an application predicting skin temperature based on heart rate and step count values) using medical datasets publicly available such as PhysioNet, UbiqLog and CrowdSignals.

In [148], the authors review the state of the art of the analytics network methodologies for real-time IoT analytics. They also present some real-time IoT analytics use cases and software platforms such as Flink, Spark, Storm, and Druid along with their network requirements. Lastly, they present research problems and future research directions focusing on the network methodologies for the real-time IoT analytics.

In summary, the articles demonstrate the benefits of collaborative Edge-to-Cloud data analytics. The main performance improvements highlighted by these studies regarding the comparison of Edge-to-Cloud *vs.* Cloud only approaches refer to minimizing the processing latency of applications.

5.3. Main takeaways

This section aims to answer the following research question: *What are the main state-of-the-art methods for data analytics on the Edge-to-Cloud Continuum?* We organize the existing frameworks and the selected studies in two main categories, they are: data processing on the Edge; and Edge-to-Cloud Big Data processing.

From the analysis of the selected articles, we observe that, compared to the Cloud, a few stream processing frameworks tailored for the Edge exist, such as Apache NiFi, Edgent, and EdgeWise. We highlight that the recent works focus on proposing novel approaches for collaborative Edge-Cloud processing in order to enable live data analytics, instead of focusing on novel processing frameworks designed for running just on the Edge.

6. Approaches to Combine Machine Learning and Data Analytics

The convergence of Big Data and AI has become a research trend that grows over the years given the benefits of combining and applying both technologies in many areas, such as self-driving vehicles [61], precision agriculture [19], smart manufacturing [78], among others. Combining Big Data and AI leverages advanced analytics capabilities and allows the efficient extraction of valuable insights from vast amounts of data [46].

Next, we present the recent efforts on combining Big Data and AI to enable hybrid Cloud and Edge analytics. Figure 5 presents the taxonomy of approaches combining Machine Learning and Data Analytics with a focus on the Edge-to-Cloud Continuum. The ML and Data Analytics frameworks/libraries identified in the articles are presented

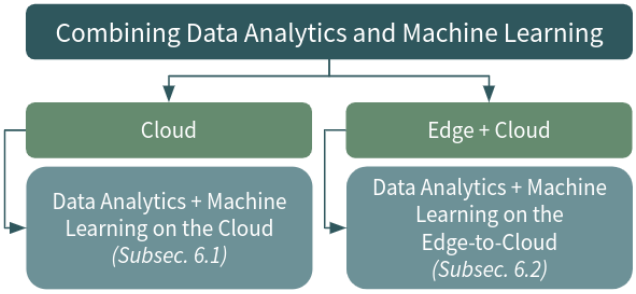


Figure 5: Taxonomy of approaches combining data analytics and learning on the Edge-to-Cloud Continuum.

in Tables 11 and 12 and Tables 13 and 14, respectively. Table 7 characterizes the selected articles regarding resources exploited in the experimental evaluations, such as: frameworks/libraries; application/task; metrics; and hardware; models; and datasets. Table 6 presents the main state-of-the-art learning paradigms for collaborative learning. Table 8 presents a quantitative analysis summarizing Table 7.

6.1. Combining Data Analytics and Learning on the Cloud

The articles presented in this section explore the joint usage of Data Analytics and Machine Learning frameworks and algorithms for analytics on Cloud resources. They also discuss the relevance of learning paradigms (*e.g.*, Deep Learning, Online Learning, and Transfer Learning, among others) for Big Data Analytics.

Distributed Cloud-based Machine Learning tools such as Mahout, Spark MLlib, and FlinkML are presented in [118]. Authors also present research directions and opportunities in the domain of developing parallel and distributed Machine Learning algorithms. For instance, they highlight that in streaming systems, there is a lack of online Machine Learning algorithms that are used to process real-time data to provide faster insights.

The survey [101] presents ML and DL frameworks and libraries oriented towards fast processing and streaming of large-scale data, such as: TensorFlow, PyTorch, MXNet, Theano, FlinkML, and Spark MLlib. Authors highlight that there is no single tool suitable for every problem and often a combination of them is needed to succeed.

Authors of [109] present key characteristics and challenges of handling Big Data. Regarding current trends in Big Data Analytics they refer to IoT and Edge analytics (to provide responses quickly as the events occur) and domain adaptation (where training data and test data are sampled from different distributions).

In [4] the authors survey existing solutions for Big Data stream processing in terms of learning type, supported languages, and supported Machine Learning tools. Authors discuss frameworks and platforms such as Apache Spark, MOA, Samza, Storm, and Kafka.

Another survey [145] presents existing open source tools for Big Data (*e.g.*, Hadoop, Spark, Storm, and Flink) and Machine Learning (*e.g.*, Mahout, Spark MLLib, and SAMOA).

Table 4: Summary of artifacts, metrics, and hardware exploited in the Data Analytics experiments.

Paper	Framework/Library	Application/Task	Metrics	Hardware
[38]	NiFi	Surveillance System (collaborative data processing)	processing time delay (network latency + data serialization + queueing)	Edge setup: 8-Megapixel camera acted as the CCTV source; 3x Raspberry Pi 3; 3x Google Nexus 4. Cloud setup: Heroku (Intel Xeon CPU 2.5GHz, 512MB RAM) and Amazon EC2 (Intel Xeon CPU 2.4GHz, 4GB RAM)
[39]	Flink	Smart Healthcare (data stream processing)	time delay: difference between the moment when sensor data are first generated and the moment when valuable insights are drawn based on these data	Raspberry Pi 3; 1x machine 2.00 GHz Intel Core i7-4510U CPU, 16 GB RAM; and Google Compute Engine n1-standard-1;
[37]	Nifi	Face detection and recognition	processing time delay (image capture + face recognition + return results to the coordinator)	5x VirtualBox instances of Raspberry Pi; Heroku; and Amazon Elastic Beanstalk;
[36]	Nifi	Stream processing	response time (time difference between an image is first sampled and the recognition task)	Raspberry Pi 3; Samsung Galaxy J5; Amazon EC2
[56]	EdgeWise, Storm	Stream processing	throughput-latency performance	8x Raspberry Pi 3 B; and 1x desktop machine
[3]	Edgent, FoT-Stream, Kafka	Stream processing	processing delay and network utilization	Sonoff SC; Raspberry Pi 3 B+; and Amazon EC2 servers
[161]	Cameo, Flink	Stream processing	query latency	DS12-v2 and DS11-v2 Azure virtual machines
[66]	Fed4Edge	Stream processing	query processing throughput	85x Raspberry Pi B
[68]	Flink, Kafka	Activity recognition (random forest classifier model)	classification time; processing time; end to end latency; and activity recognition accuracy	Edge: Raspberry Pi 3; Cloud: Intel Xeon CPU E5-2640 v3 (2.6 GHz)
[123]	Flink, Kafka, Edgent	video stream processing	end-to-end latency, processing throughput	35x machines of the Grid5000 testbed

Table 5
Quantitative analysis of artifacts, metrics, and hardware exploited in the Data Analytics experiments. Percentages refer to the total number of papers in that domain.

Framework/Library		Metrics		Hardware (Edge)		Processors	
Kafka	19%	end to end latency	47%	Raspberry Pi	53%	CPU	100%
Flink	15%	network	16%	Emulated	18%	GPU	0%
NiFi	11%	model accuracy	16%	Arduino	12%	TPU	0%
Storm	11%	throughput	16%	Google Nexus 4	6%		
Hadoop	7%	classification time	5%	Samsung Galaxy J5	6%		
Spark	7%			Sonoff SC	6%		
Edgent	4%						
Flume	4%						
EdgeWise	4%						
FoT-Stream	4%						
Cameo	4%						
Fed4Edge	4%						
MOA	4%						

Besides, authors review Machine Learning algorithms in Big Data such as Supervised, Unsupervised, and Semi-supervised learning.

In [88] the authors present the challenges associated with Machine Learning in the context of Big Data and categorize them according to the Velocity, Volume, Variety, and Veracity dimensions of Big Data. They also present existing Machine Learning approaches and techniques for data manipulation (*e.g.*, dimensionality reduction and data cleaning); processing manipulation (such as vertical/horizontal scaling and batch/stream oriented); and algorithm manipulation (*e.g.*, algorithm modification with new paradigms). Lastly, they present learning paradigms relevant to Big Data such as Deep Learning, Lifelong Learning, Online Learning, and Transfer Learning.

A parallel Machine Learning algorithm for fault classification of mobile robotic roller bearings is proposed in [157]. The proposed algorithm combines Support Vector Machine with Spark to realize parallel operations. Experimental evaluations under different training set sizes demonstrate that the proposed algorithm (Spark SVM on Mesos) outperforms MapReduce SVM, Storm SVM, Radial Basis Function Neural Network (RBFNN), Deep Belief Network (DBN), presenting higher: classification accuracy, processing speed, and convergence rate.

In [12] the authors explore Spark MLlib with a variety of Big Data Machine Learning experiments on massive datasets to understand the qualitative and quantitative attributes of the platform. Experimental evaluations compare the performance of classification and clustering models (such as SVM, Decision Tree, Naive Bayes, Random Forest, and K-Means) on a variety of hardware and software configurations. Results show that with large datasets Spark MLlib outperforms Weka in terms of running time.

In [99] the authors propose a system for real-time health status prediction based on the Spark Big Data processing framework. The proposed system predicts user's health status by applying a variety of decision tree models on streams of data. Experiments evaluate the generalization error of Decision Tree models based on maxDepth (tree depth) and maxBins (ordered splits) parameter values.

In [5] the authors propose an Edge-Cloudlet-MultiResource three-tier architecture to enable real-time processing of video streams. The proposed system performs Deep Learning inference on Cloudlets and distributes processing stages on the available resources using an algorithm to satisfy user Quality of Service requirements. Results show that for a 10K element data streams, with a frame rate of 15-100 per second, the job completion in the proposed system takes 49% less time and saves 99% bandwidth compared to a centralized Cloud-only based approach.

An overview on Spiking Neural Networks (SNN) for Online Learning scenarios is presented in [84]. According to the authors, the use of SNN in Online Learning allows fast real-time simulations of large networks and a low computational cost. SNN make possible the accumulation of knowledge as data become available without the requirement of storing and retraining the model with past samples. The authors also highlight research trends in the field of SNN and Online Learning such as Lifelong Machine Learning and Deep SNN Learning.

6.2. Combining Data Analytics and Learning on the Edge-to-Cloud Continuum

Next, we present the recent efforts on combining state-of-the-art Big Data Analytics and Machine Learning approaches to enable intelligence on the Edge-to-Cloud Continuum. The following works focus on novel systems, frameworks and architectures.

A novel architectural design for enabling machine and deep learning over heterogeneous data streams on hybrid Cloud and Edge Computing infrastructures is proposed in [74]. Named Stream to Cloud and Edge (S2CE), the platform aims to enable mining of Big Data streams over Cloud and Edge. It provides functionalities of scalable processing, such as distributed processing, data fusion and preprocessing, and

Table 6
Quantitative analysis of Learning Paradigms for learning on the Edge-to-Cloud Continuum. Percentages refer to all papers reviewed which are discussing or exploiting in their experiments these learning paradigms.

Paper	Pct.	Learning Paradigm	Main ideas	Characteristics
[169, 77, 31, 67, 127, 67, 151, 88, 131, 96, 132, 83, 165, 114, 72, 160, 84, 164, 52, 9, 106, 119]	30%	Distributed Machine/ Deep Learning [130]	Exploit distributed resources of a cluster to speed up the convergence of model training.	• aims at parallelizing computing power. • simultaneously places and processes data for model training and testing into a number of distributed nodes.
[60, 67, 88, 96, 132, 165, 118, 145, 101, 72, 160, 84, 60, 44, 52, 170]	21%	Online Learning or Sequential Learning [81]	Allow updating models upon arrival of new data without the need to retrain the complete model.	• models learn one instance at a time: does not rebuild the model every time new data arrives, instead, it updates the existing knowledge based on the new incoming data. • models are frequently updated: dynamic updates of the trained models are determining factors for a reliable and efficient analytic module. • uses data streams for model training: can accommodate bigger datasets than batch learning and matches the need for stream analysis of IoT data. Besides, it can be efficiently deployed in an Edge device as they do not accumulate large amount of data.
[151, 131, 96, 165, 72, 84, 52, 9, 119, 30, 120]	16%	Reinforcement Learning [142]	Learning is based on feedback obtained from interactive actions in the environment.	• does not require a training data set: it interacts with the external environment to continuously adapt and learn on given points as a kind of feedback.
[77, 144, 88, 96, 165, 109, 72, 44, 85, 30]	14%	Transfer Learning [107] and Multi-task Learning [168]	Exploit the knowledge obtained from a task to improve generalization about another.	• useful when you have lack of training data sets. • models may be reused as a starting point for predicting in another, but related, domain. • compared to separated model training, it improves learning efficiency and prediction accuracy for the task-specific models.
[86, 77, 144, 96, 165, 44, 9, 85, 30, 120]	14%	Federated Learning [21]	Train a centralized model in a distributed way without the need to share private data.	• aims at training on heterogeneous datasets. • data privacy: transmits and aggregates trained model parameters instead of training data. • enables scalability: leverages the concurrent use of a high number of Edge devices to be independently trained and periodically synchronized through a central parameter server.
[88, 165, 84, 52]	5%	Lifelong Learning or Continual Learning [133]	Learning is continuous and knowledge is retained and used to solve different problems.	• does not rebuild the knowledge model every time a new piece of data arrives, but only updates the existing knowledge with the new incoming data. • can accommodate bigger datasets than batch learning. • may be a promising solution for lack of data, real-time processing, and concept drift.

Table 7: Summary of articles combining Data Analytics and Machine Learning in their experimental evaluations.

Paper	Framework/Library	Application/Task	Metrics	Hardware	Model	Dataset
[151]	5G I-IoT framework	5G channel utilization	latency, spectrum efficiency, energy efficiency, data rates, reliability, QoS, and load	simulation	NA	NA
[60]	Flink, Flume, Kafka, MOA	Processing of vehicle location data	prediction accuracy	Not informed.	Combined Stream and Neural Network (CSaNN), Combined Stream and Random Forest (CSaRF)	Warsaw tram data (WAW)
[111]	R	Traffic forecasting (model training)	amount of data collected by the Fog Nodes; impact of network between Fog Nodes and Cloud; model accuracy Edge vs Cloud;	Cluster of 8 servers: 2x Xeon E5-2630v4 (broadwell) and 128 GB of DDR4-2400 R ECCRAM. Edge devices: RaspberryPi 3 B	Conditional Restricted Boltzmann Machines (CRBM)	Floating Car Data (FCD)
[127]	Scikit-learn, Storm, Nifi, Hadoop, Kafka	Distributed model training	network latency and service latency	IoT device: 10x Arduino; Edge device: 4x Raspberry Pi 3; Edge gateway: 2x Intel E5645@2.4 GHz and memory of 24 GB	Data Flow and Distributed Deep Neural Network (DF-DDNN)	IoT truck simulator by Horton works
[129]	Tensorflow, Keras, Scikit-Learn, Spark, Kafka	Fall detection system (model training)	model accuracy, sensitivity, and precision	Edge: Arduino Uno; Fog: Raspberry Pi 2 B; Cloud: 5x Amazon EC2 and 2x T2.micro; and 1x T2.medium	RNN (LSTM/GRU)	SisFall
[12]	Spark, Weka	inference	running time	2x virtual machines with 8 GB 4 vCPUs; and 16 GB 8 vCPUs	SVM, Decision Tree, Naive Bayes, Random Forest, and K-Means	HEPMASS, SUSY, HIGGS, FLIGHT, HETROACT
[99]	Spark	health status prediction	generalization error	1x Intel i5 and 8GB RAM; and 3x Amazon EC2	Decision tree	heart disease
[157]	Spark, Storm, and Hadoop	fault classification	classification accuracy, classification time, convergence rate	18 virtual machines on 6 physical machines	SVM, Radial Basis Function Neural Network (RBFNN), Deep Belief Network (DBN)	fault pattern
[5]	TensorFlow, Spark, Kafka, Storm, Hadoop	Video stream analytics (deep learning inference)	job execution time; and bandwidth consumed	Xeon E5 system with 8 cores and Nvidia GTX 970 GPU	MobileNet	Tiny ImageNet; and Stanford vehicle dataset and CASIA-Webface (face recognition)

Table 8: Quantitative analysis of artifacts, metrics, and hardware exploited in the experiments combining Machine Learning and Data Analytics. Percentages refer to the total number of papers in that domain.

Framework/Library		Metrics		Hardware (Edge)		Processors		Model		Dataset	
Spark	19%	network	24%	Emulated	28%	CPU	100%	SVM	14%	WAW	10%
Kafka	15%	model accuracy	19%	RaspberryPi	21%	GPU	0%	CSaNN	7%	FCD	10%
Storm	11%	processing time	14%	Arduino	10%	TPU	0%	CSaRF	7%	SisFall	10%
Hadoop	11%	spectrum efficiency	5%					DF-DDNN	7%	HEPMASS	10%
Scikit-learn	7%	energy efficiency	5%					LSTM	7%	SUSY	10%
Tensorflow	7%	data rates	5%					GRU	7%	HIGGS	10%
5G I-IoT	4%	reliability	5%					kNN	7%	FLIGHT	10%
Flink	4%	QoS	5%					MobileNet	7%	HETROACT	10%
Flume	4%	load	5%					Decision Tree	7%	Stanford vehicle	10%
MOA	4%	amount of data collected	5%					Naive Bayes	7%	CASIA-Webface	10%
R	4%	generalization error	5%					Random Forest	7%		
Nifi	4%	convergence rate	5%					K-Means	7%		
Keras	4%							Tiny ImageNet	7%		
Weka	4%										

Cloud and Edge resource management.

In [151] the authors propose 5G Intelligent Internet of Things (5G I-IoT). This approach is based on Big Data mining, Deep Learning, and Reinforcement Learning to process data intelligently and to optimize communication channels. The framework consists of three building blocks: (1) a processing center in the Cloud to handle real-time data for decision making using Deep Learning and Reinforcement Learning; (2) an object processor in the Fog to processes the raw data from sensing regions; and (3) the sensing regions in the Edge. Evaluations show that 5G I-IoT outperforms 4G-IoT and 5G-IoT in terms of effectiveness of channel utilization.

Authors of [127] propose a Data Flow and Distributed Deep Neural Network (DF-DDNN) that integrates data flow and distributed Deep Learning in the IoT-Edge environment to bring down the latency and increase accuracy. Experimental results show that the proposed solution enables a latency reduction of up to 33% when compared to the existing traditional IoT-Cloud model. In [60], a hybrid technique combining batch learning, Online Learning, and stream mining to predict delays of public transport vehicles is proposed. The hybrid approach is validated by experiments with real public transport delay data streams.

Recent studies have also proposed novel architectures for collaborative Edge-Cloud learning and data analytics. A review of existing reference architecture designs of Big Data systems such as FAR-Edge [51] and Global Edge Computing Architecture [135] is presented in [106]. Authors propose a novel reference architecture design of a Big Data system with a focus on the utilization of ML in Edge Computing environments. In [72] the authors present an overview of AI approaches for Autonomous Vehicle (AV) and propose a concept architecture for integrating Artificial Intelligence with Edge Computing. They also discuss key issues and challenges on: data fusion, such as the reconstruction and understanding of the environment of AV; and Big Data Analytics for training systems and real-time decision-making of AV volumes of data.

A novel architecture that combines a data distribution layer connecting Fog nodes with a Cloud focusing on resilience, near real-time communication, and a traffic modeling approach is proposed in [111]. The modeling approach is an Online Machine Learning technique named Conditional Restricted Boltzmann Machines (CRBM) to learn and predict traffic telemetry. Experimental results show that the Cloud-based processing approach can produce severe impact in the accuracy of Cloud-learned models due to network connectivity outages between the Fog and the Cloud.

An Edge-Cloud collaborative computing platform for Artificial Intelligence of Things (AIoT) is proposed in [120]. Named Sophon Edge, the platform helps to build and deploy AIoT applications efficiently. It addresses challenges related to building AIoT applications in practice, such as heterogeneity (*e.g.,* communication protocols, data format, operating systems, among others) and accuracy of AI algorithms (*e.g.*, model refinement and tuning).

Authors of [129] propose a system to detect falls leveraging an Edge-Fog-Cloud architecture to deploy DL models into resource-constrained devices for DL inference. The architecture exploits Big Data Analytics resources for training DL models on the Cloud and performing inference on devices. They also present a practical and experimental deployment of DL models on Fog devices and the lightweight virtualization technologies, such as Docker containers, to optimize the resource usage. Their solution leverages the RNN (LSTM/GRU) algorithms since they are appropriate for sequential data such as IoT monitoring and they fulfill the resource constraint requirements and provide very high accuracy.

In summary, the systems, frameworks and architectures proposed by the articles aim to mainly: enable Cloud and Edge resource management; optimize network communication; provide efficient application deployments; and allow distributed data processing. Some approaches also exploit hybrid techniques combining batch learning, Online Learning, Reinforcement Learning, Deep Learning, among others to improve application performance (*e.g.*, minimize latency, increase accuracy, *etc.*).

6.3. Main takeaways

This section aims to answer the following research question: *How are the existing Machine Learning and Data Analytics approaches combined to enable intelligence on the Edge-to-Cloud Continuum?* We organize the selected studies in two main categories, they are: Data Analytics and Machine Learning on the Cloud; and Data Analytics combined with Machine Learning on the Edge-to-Cloud Continuum.

We highlight that the recent efforts (*e.g.* systems, frameworks and architectures) focus on applying, combining, and deploying Machine Learning paradigms such as Federated Learning, Transfer Learning, Multi-task Learning, Reinforcement Learning and Online Learning on distributed Edge devices for collaborative Edge-to-Cloud analytics. Such efforts focus mainly on addressing open challenges, such as: enabling fast and accurate predictive analytics; optimize communication channels and minimize connectivity issues in Edge data analytics; minimize the processing latency of applications to satisfy Quality of Service requirements; just to cite a few.

7. Experimental Research and Reproducibility

In this Section we introduce the main state-of-the-art simulation, emulation, and deployment systems supporting experimental research on the Edge, Fog, Cloud, and Edge-to-Cloud Continuum [143, 164]. Besides, we discuss the relevant experimental testbeds enabling Edge-to-Cloud experiments. We then analyze the previously selected articles in terms of experimental evaluation aspects, such as the size of the experimental testbed and the support to the reproducibility of experiments.

Table 9
Simulation, Emulation, and Deployment Systems for Experimental Research on the Edge-to-Cloud Continuum.

Simulation Systems	Edge	Fog	Cloud	Main Goal	Key Features
CloudSim [28]			✓	Modeling, simulation, and experimentation of Cloud infrastructures and application services.	• modeling and simulation of large scale Cloud computing data centers. • modeling and simulation of virtualized server hosts and application containers. • modeling and simulation of energy-aware computational resources. • modeling and simulation of data center network topologies.
SCORE [53]			✓	Simulate energy- efficiency, security, and scheduling strategies in Cloud Computing environments.	• allows to prototype and compare different cluster scheduling strategies and policies. • generates synthetic cluster workloads from empirical parameter distributions. • allows the analysis of scheduling performance metrics.
ElasticSim [26]			✓	Simulate autoscaling algorithms.	• supports resource runtime auto-scaling. • supports stochastic task execution time modeling.
iFogSim [63]		✓		Modeling and simulation of resource management techniques in IoT, Edge and Fog Computing environments	• inherits a number of features from CloudSim. • provides resource management techniques in IoT, Edge and Fog. • allows the execution of multiple applications on the infrastructure at the same time. • supports migration of application modules from one fog device to another.
FogNetSim [115		✓		Simulate distributed fog computing environments.	• covers the network aspects such as delay, packet error rate, transmission range, handover, scheduling, and heterogeneous mobile devices. • allows to simulate a large fog network. • allows to simulate heterogeneous devices with varying features. • supports handover: allows static and dynamic nodes in the network.
FogTorch [25]		✓		QoS-aware deployment of IoT applications through the Fog.	• allows the specification of a Fog infrastructure along processing (*e.g.,* CPU cores, RAM memory, storage) and QoS (*e.g.,* latency, bandwidth) capabilities. • allows the specification of applications to be deployed along with needed IoT devices, processing and QoS requirements.
FogExplorer [65]		✓		Simulate QoS and cost evaluation of fog-based IoT applications.	• simulates processing cost and processing time for individual application modules. • simulates transmission cost and transmission time for individual data streams.

Simulation Systems	Edge	Fog	Cloud	Main Goal	Key Features
IoTSim-Edge [100]	✓			Simulate the distribution and processing of streaming data generated by IoT devices in Edge computing environments.	• allows to define data analytic operations and their mapping to different parts of the infrastructure. • supports modeling of heterogeneous IoT protocols along with their energy consumption profile. • supports modeling of mobile devices and captures the effect of handoff caused by the movement of mobile devices.
EdgeCloud-Sim [137]	✓			Simulate environments specific to Edge Computing scenarios.	• considers computing and networking resources. • supports network modeling specific to WLAN and WAN. • supports device mobility model and provides realistic and tunable load generator.
YAFS [79]	✓			Analyze the design and deployment of applications through customized and dynamical strategies.	• allows dynamic scenarios: placement, path routing, orchestration, and workload movement. • supports placement allocation algorithms and orchestration algorithms. • provides functions to obtain metrics such as network utilization, network delay, response time, and waiting time.
XFogSim [90]	✓			Simulate federated fog computing environments.	• provides resource allocation algorithms for resource sharing. • supports static and mobile nodes (handover mechanisms). • supports application evaluation in terms of: energy consumption, processing latency, scalability, and resource usage.
Emulation Systems	**Edge**	**Fog**	**Cloud**	**Main Goal**	**Key Features**
EmuFog [91]		✓		Enable the design of Fog Computing infrastructures and the emulation of real applications and workloads.	• generates networks that can be emulated easily with MaxiNet [154]. • supports topologies from BRITE [92] and Caida [27]. • places fog nodes based on user-defined constrains (*e.g.*, network latency or resource constraints).
Fogbed [34]		✓		Enable the rapid prototyping of Fog components in virtualized environments.	• allows dynamic topology changes. • provides traffic control links such as delay, rate, loss, and jitter. • enables the deployment of Fog nodes as software containers under different network configurations.
RADICAL-DREAMER [116]	✓	✓	✓	Emulate resource and task/workload definition in Edge-to-Cloud applications.	• allows to evaluate workload and resource management aspects of applications. • supports modeling task placement in Edge-to-Cloud applications. • allows to evaluate deployment modalities and performance trade-offs.

Deployment Systems	Edge	Fog	Cloud	Main Goal	Key Features
E2C*lab* [123]	✓	✓	✓	Understand and optimize the performance of Edge-to-Cloud workflows through reproducible experiments on large-scale infrastructures.	• supports reproducible experiments. • provides a *Services* abstraction to support other applications. • provides resource monitoring (*e.g.,* CPU, GPU, memory, network) and network emulation to define Edge-to-Cloud constraints such as delay, loss and rate. • maps application parts with the underlaying testbed. • allows to optimize workflows through optimization libraries for hyperparameter search in Ray Tune [82].
KubeEdge [159]	✓			Deploy complex high level applications to the Edge.	• provides containerized application orchestration and device management to hosts at the Edge. • provides core infrastructure support for networking, application deployment and metadata synchronization between Cloud and Edge. • supports MQTT which enables Edge devices to access through Edge nodes.
Kubernetes [24]			✓	Manage and automate the deployment, scaling, and management of containerized applications across multiple hosts.	• provides mechanisms for deployment, maintenance, and scaling of applications. • provides service discovery and load balancing. • allows to automatically mount storage systems, such as local storage and public Cloud providers.

7.1. Simulation, Emulation, and Deployment Systems for Experimental Research

Table 9 summarizes the main open-source state-of-the-art simulation, emulation, and deployment systems for experimental research on the Edge-to-Cloud Continuum.

7.1.1. Simulation Systems

Building experimental testbed environments is expensive and brings challenges to conduct reproducible experiments. In this sense, simulation systems play an important role as they allow to analyze systems behavior at very large scale while easily tuning a myriad of configuration parameters. Next, we present simulation systems used in the modeling of Cloud, Fog, and Edge computing environments.

Cloud-based simulation systems. CloudSim [28] framework allows modeling and simulation of Cloud computing infrastructures and services in a repeatable manner. CloudSim allows users to model the behavior data centers, Virtual Machines and resource provisioning policies. ElasticSim [26] is a workflow simulator that extends CloudSim. It focuses on supporting resource runtime auto-scaling and stochastic task execution time modeling. SCORE [53] allows the execution of heterogeneous workloads for simulating energy-efficient monolithic and parallel-scheduling models.

Fog-based simulation systems. FogExplorer [65] provides modeling and simulation to estimate QoS and cost evaluation of Fog-based IoT applications. FogExplorer allows users to choose good application designs during its design phase. FogTorch [25] aims to support the deployment of IoT applications in Fog infrastructures considering software, hardware and QoS requirements. FogNetSim++ [115] focuses on simulating large Fog networks and differs from others mainly by providing features that allow users to incorporate customized mobility models, scheduling algorithms, and manage handover mechanisms. XFogSim [90] extends FogNetSim++ to simulate federated fog computing environments. xFogSim is lightweight, configurable, scalable and introduces the concept of fog federation for resource sharing among fog locations. Furthermore, it allows users to evaluate applications in terms of energy consumption, processing latency, scalability, and resource usage. YAFS [79] aims to allow users to analyze application designs and incorporate strategies for placement, scheduling and routing. YAFS also supports dynamic allocation of new application modules, dynamic failures of network nodes, and user mobility. Furthermore, it facilitates the shareability of experiment results by generating logs of workload generation and computation, and link transmissions. Lastly, iFogSim [63] focuses on resource management techniques in IoT, Edge and Fog computing environments. iFogSim allows users to measure, in a repeatable manner, the impact of resource management techniques in terms of latency, network congestion, energy consumption, and cost.

Edge-based simulation systems. EdgeCloudSim [137] focuses on Edge Computing scenarios and allows one to conduct experiments considering computational and networking resources. IoTSim-Edge [100] allows users to easily configure their Edge infrastructures and to capture the behavior of heterogeneous IoT and Edge devices in terms of sensing, processing, mobility, and data rate. Both Edge systems extend CloudSim.

7.1.2. Emulation Systems

Compared to simulation, the emulation approach provides more realistic results. While simulators mimic the behavior and configurations of a real device, emulation systems duplicate the hardware and software features of a real device [139]. Emulation systems are also a less expensive solution when compared to real deployments.

Fogbed [34] allows resource provisioning emulation in Fog environments. It combines Containernet [112] and Maxinet [154] (both are extensions of the Mininet [69] network emulator) to allow the use of virtual instances for resource provisioning emulation.

EmuFog [91] focuses on the design of Fog Computing infrastructures and the emulation of real applications and workloads. In EmuFog, users can: design the network topology; embed Fog nodes in the topology; and run Docker-based applications on those nodes connected by an emulated network.

RADICAL-DREAMER [116] provides the concepts of *Task* and *Workload* to model the characteristics of an application according to heterogeneous tasks. Besides, it provides the concept of *Resource* to model distributed infrastructures. RADICAL-DREAMER allows users to evaluate deployment configurations, performance trade-offs, and workload placement strategies for Edge-to-Cloud applications [87].

7.1.3. Deployment Systems

Deploying real-life applications on large-scale testbeds provides the most realistic results compared to simulation or emulation approaches. In this direction, a few systems have been proposed in the past few years.

E2Clab [123] is a framework that implements a rigorous methodology for designing experiments with real-world workloads on the Edge-to-Cloud Continuum. E2Clab provides guidelines to move from real-world use cases to the design of relevant testbed setups for reproducible experiments enabling researchers to understand and optimize [121] the performance of applications. The key features provided by E2Clab are [122]: (1) reproducible experiments; (2) the mapping of applications parts executed across the computing continuum with the physical testbed; (3) the support for experiment variation and transparent scaling of the scenario; (4) network emulation to define Edge-to-Cloud communication constraints; (5) experiment deployment, monitoring and backup of results; and (6) the application optimization.

Kubernetes [24] aims to simplify the deployment and management of services that compose an application by providing mechanisms for deployment, maintenance, and scaling. Using Kubernetes, users can manage containerized applications across multiple hosts. KubeEdge [159] builds on top of Kubernetes to extend Cloud capabilities to the Edge and allows containerized application orchestration and device management to hosts at the Edge. KubeEdge key features are: core infrastructure support for networking; application deployment; and metadata synchronization between Cloud and Edge.

7.2. Large-Scale Experimental Testbeds for Edge-to-Cloud Experiments

Several experimental testbeds allow researchers to evaluate their proposals in real-life settings by providing access to a large amount of resources (grouped in homogeneous or heterogeneous clusters, upon convenience) and, more importantly, supported by some vibrant communities of users and solid technical teams. We cite here just a few.

Grid5000 [20] is a large-scale French testbed for experimental research with a focus on parallel and distributed computing including Cloud, HPC, Big Data, and AI. Grid5000 is merging with FIT IoT-Lab to enable Edge-to-Cloud experiments. FIT IoT-Lab [2] is a large-scale multi-radio (*e.g.*, IEEE 802.15.4, Bluetooth Low Energy, LoRa, etc.) and multi-platform (*e.g.*, Arduino Zero, nRF52840-MDK, LoRa gateway, and many others) infrastructure for the Internet of Things. FIT IoT-Lab consists of more than 1.5K nodes and provides tools for monitoring energy consumption and network-related metrics, such as end-to-end delay, throughput and overhead. A recent effort on supporting experiments combining Grid5000 and FIT IoT-Lab testbeds is EnOSlib [32]. EnOSlib is a library which brings reusable building blocks for configuring the infrastructure, provisioning software on remote hosts as well as organizing the experimental workflow.

Chameleon [70] is a large-scale US experimental platform that aims to support Computer Science research in many areas, such as: systems, storage, networking, GPU, security, Artificial Intelligence, and High Performance Computing. CHI@Edge is an extension of Chameleon testbed that aims to support Edge Computing experiments. Combining Chameleon and CHI@Edge testbeds allows more realistic Edge-to-Cloud experiments since it provides access to real-life IoT/Edge devices such as Raspberry Pis, Jetson Nanos, among others.

ORBIT [105] (Open-Access Research Testbed for Next-Generation Wireless Networks) is based on a 20x20 two-dimensional grid of programmable radio nodes which can be interconnected into different topologies. ORBIT provides access to: radio resources, including WiFi 802.11a/b/g 802.11n 802.11ac, Bluetooth (BLE), ZigBee, and Software Defined Radio platforms; Software defined networking (SDN) resources; LTE and WiMAX base stations and clients; and Cloud resources such as nodes with Tesla-based GPUs.

SmartSantander [126] is a large scale testbed composed of around 2000 IEEE 802.15.4 devices deployed in a 3-tiered architecture (IoT node, repeaters, and gateway node) deployment in the Spanish city of Santander. The testbed allows IoT native experimentation (*e.g.* wireless sensor network experi-

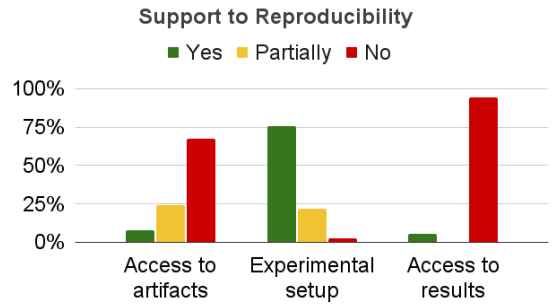


Figure 6: Support to the reproducibility of experiments provided by the selected studies.

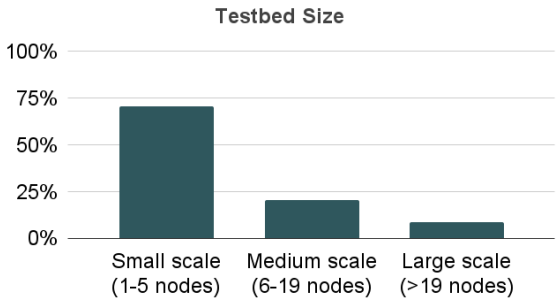


Figure 7: Testbed size used in the experimental evaluations: small scale [169, 77, 31, 86, 167, 67, 151, 39, 128, 37, 60, 170, 97, 5, 57, 3, 161, 62, 80, 50, 76, 43, 12, 99], medium scale [38, 111, 127, 129, 36, 56, 157], and large scale [66, 123, 121]

ments) and service provision experiments (*e.g.* applications using real-time real-world sensor data).

Fed4FIRE+ [41] is a project offering the largest federation worldwide of Next Generation Internet (NGI) testbeds. Fed4FIRE aims to provide open, accessible and reliable experimental infrastructures supporting a wide variety of research, such as 5G, IoT, Cloud Computing, Wired and Wireless Computer Networking. The list of testbeds [103] federated with Fed4FIRE are: CityLab [141], PlanetLab [55], ExoGENI [15], Tengu [146], NITOS [110], w-iLab [22], among others.

7.3. Support to Experimental Reproducibility

A desired feature of any experimental research is that its scientific claims are verifiable by others in order to build upon them. This can be achieved through Repeatability, Replicability, and Reproducibility [16, 140]. Find in Table 10 the terminology proposed by the ACM Digital Library.

We evaluate the support to the reproducibility of experiments for each selected article. This evaluation is based on the following three main relevant aspects:

Access to artifacts: if authors provide access to a public repository with the artifacts used to run the experiments, such as: datasets, codes, applications, systems, configuration files, among others.

Experimental setup: if authors provide a description of the experimental setup, such as: hardware configuration

Table 10
ACM Digital Library Terminology [54]

Repeatability	*Same team, same experimental setup: the measurement can be obtained with stated precision by the same team using the same measurement procedure, the same measuring system, under the same operating conditions, in the same location on multiple trials. For computational experiments, this means that a researcher can reliably repeat their own computation.*
Replicability	*Different team, same experimental setup: the measurement can be obtained with stated precision by a different team using the same measurement procedure, the same measuring system, under the same operating conditions, in the same or a different location on multiple trials. For computational experiments, this means that an independent group can obtain the same result using the author's own artifacts.*
Reproducibility	*Different team, different experimental setup: the measurement can be obtained with stated precision by a different team, a different measuring system, in a different location on multiple trials. For computational experiments, this means that an independent group can obtain the same result using artifacts which they develop completely independently.*

of physical machines, software or systems used, network configurations, among others.

Access to results: if the computed experimental results are available in a public repository, such as: log files, files metric collected during runtime, monitoring data, code to plot charts, among others.

Figure 6 summarizes the support to the reproducibility of experiments provided by the selected studies. Regarding the **access to artifacts**, 68% of papers do not provide access to them, and just 24% partially provide (a few artifacts, but not all). Analyzing the description of the **experimental setup**, 76% of papers describe it in detail in a dedicated section of the paper, while 21% only partially describe it and just 3% do not provide enough information. Lastly, regarding the **access to results**, 95% of the articles do not provide access and just 5% provide a public repository with the results. In general, we notice a lack of support to the experimental reproducibility in the domain of Edge-to-Cloud experimental research.

Lastly, Figure 7 presents the size of the testbeds used in the experimental evaluations. As one may note, 70% of papers use small scale setups, composed by at most 5 machines or devices, while 20% of them use testbed setups composed by 6 to 19 nodes, and just 10% experiment in large scale setups with 20 nodes or more.

7.4. Main takeaways

This section aims to answer the following research question: *What are the existing solutions for experimental research and how do the selected studies support the reproducibility of the experiments?* We identify and summarize the key characteristics of the main state-of-the-art simulation, emulation, and deployment systems. Furthermore, we discuss the recent efforts and initiatives merging large scale testbeds for enabling more realistic setups for Edge-to-Cloud experiments, such as: Grid'5000 and FIT IoT-Lab; Chameleon and CHI@Edge; and the Fed4FIRE project.

We also analyze the selected studies regarding their sup-

Table 11
Machine Learning frameworks/libraries designed for the Cloud. Marked with a ⋆ are the mostly exploited in the experiments.

Framework/Library	Qty.	Paper
⋆ **Tensorflow**	24	[169, 31, 86, 167, 144, 67, 88, 96, 83, 165, 114, 109, 101, 128, 75, 162, 164, 129, 106, 119, 30, 62, 120, 43]
⋆ **PyTorch**	10	[167, 96, 83, 165, 101, 75, 162, 164, 170, 30]
Caffe	7	[96, 165, 101, 75, 162, 30, 43]
SAMOA	6	[4, 88, 118, 145, 84, 74]
Mahout	6	[4, 88, 163, 114, 118, 145]
⋆ **Scikit-Learn**	5	[86, 101, 84, 127, 129]
CNTK	5	[167, 67, 96, 101, 162]
MOA	5	[4, 88, 84, 60, 74]
Spark MLlib	5	[4, 118, 73, 145, 101]
⋆ **Keras**	4	[86, 101, 128, 129]
Chainer	4	[96, 101, 162, 80]
R	3	[4, 88, 111]
Vowpal Wabbit	3	[88, 101, 74]
Theano	3	[96, 165, 101]
Gaia	2	[67, 165]
Flink ML	2	[118, 101]
Mistify	1	[62]
CLONE	1	[86]
DLion	1	[67]
Ako	1	[67]
TUX2	1	[165]
Weka	1	[12]

Table 12
Machine Learning frameworks/libraries designed for the Edge. Marked with a ⋆ are the mostly exploited in the experiments.

Framework/Library	Qty.	Paper
⋆ **MXNet**	11	[167, 67, 96, 83, 165, 101, 102, 162, 164, 170, 30]
⋆ **Caffe2**	7	[167, 83, 165, 101, 162, 164, 30]
TF Lite	5	[167, 83, 162, 164, 85]
CoreML	4	[83, 165, 162, 97]
TF Federated	2	[77, 85]
Neurosurgeon	2	[77, 83]
MOCHA	1	[77]
FedProx	1	[77]
TensorRT	1	[77]

Table 13
Data Analytics frameworks designed for the Cloud. Marked with a ⋆ are the mostly exploited in the experiments.

Framework/Library	Qty.	Paper
⋆ **Spark**	22	[144, 4, 88, 148, 131, 96, 163, 83, 114, 118, 73, 145, 109, 101, 84, 129, 74, 120, 12, 99, 157]
⋆ **Flink**	15	[144, 148, 131, 96, 83, 118, 145, 101, 39, 68, 84, 60, 74, 161, 123]
⋆ **Kafka**	15	[4, 163, 118, 73, 145, 109, 101, 68, 60, 127, 129, 74, 3, 123]
Storm	11	[4, 88, 148, 131, 96, 145, 84, 127, 56, 74, 157]
Hadoop	8	[88, 96, 114, 118, 145, 109, 101, 157]
Samza	4	[4, 118, 84, 74]
Flume	3	[118, 145, 60]
Cameo	1	[161]
Druid	1	[148]

Table 14
Data Analytics frameworks designed for the Edge. Marked with a ⋆ is the mostly exploited in the experiments.

Framework/Library	Qty.	Paper
⋆ **Nifi**	5	[38, 37, 127, 36, 74]
Edgent	3	[83, 3, 123]
EdgeWise	1	[56]

port to the reproducibility of experiments. As a conclusion, the results presented in Figure 6 reinforce the need for rigorous experimental methodologies that provide guidelines to the reproducibility of experiments in the Edge-to-Cloud research domain. At the same time, Figure 7 highlights the need for methodologies and deployment systems guiding researchers to evaluate and validate their proposed approaches in large-scale environments.

The development of novel systems, frameworks, or libraries abstracting the complexities of deploying Edge-to-Cloud workflows on large scale testbeds in addition with the management of the whole experimental cycle such as monitoring, gathering of results, and provenance of the experimental setup are extremely relevant. Recent advances in this direction exist, like the EnOSlib [32] library or the **E2C***lab* [123] framework, but further advances are still needed.

8. Major Findings

We gained the following insights and learned some lessons from this systematic review:

1. The **most common AI frameworks and libraries** exploited in the articles are: Tensorflow and PyTorch for the Cloud; and MXNet and Caffe2 (now part of PyTorch) for Deep Learning on the Edge. In turn, the **most common Data Analytics frameworks** used in the articles are: Apache Spark, Apache Flink, and Apache Kafka for the Cloud; and Apache Nifi for the Edge. Very **few open source Data Analytics** frameworks designed **for the Edge** were identified.

2. The most cited AI learning paradigms are respectively: **Distributed ML/DL, Online Learning, Reinforcement Learning, Transfer/Multi-task Learning**, and **Federated Learning**. Although widely used, very few articles are exploring their **performance trade-offs at scale** and their potential jointly utilization across the Edge-to-Cloud Continuum.

3. The **hardware heterogeneity**, regarding Edge devices, is **not sufficiently analyzed** in the validation phase of the proposed systems, frameworks and architectures designed to enable intelligence on the Edge-to-Cloud Continuum. Evaluations mainly rely on Raspberry Pi's or emulate resource-limited devices. Given the highly heterogeneity characteristic of the Edge-to-Cloud Continuum, it is strongly recommended that future works exploit **GPUs** and **TPUs** enabled devices, in addition to CPUs.

4. The majority of the articles proposing novel approaches or exploring existing solutions to enable distributed intelligence on the Edge-to-Cloud Continuum are performing evaluations on **small-scale testbeds** (*e.g.*, with less than 6 machines or devices, in average). It is strongly recommended that the proposed distributed approaches for ML and DL training/inference or data stream processing be validated in larger-scale environments, to assess the issues of real-life Edge-to-Cloud applications.

5. We observed that the articles **do not follow systematic experimental methodologies**. Hence, most of them do not provide enough support to enable the **reproducibility** of the experiments by other researches. Despite most of the articles describing the experimental setup in a dedicated section of the paper, most of them do not provide access to public repositories sharing the artifacts used neither the results obtained. It is strongly recommended that future works consider adopting rigorous methodologies in their experimental evaluations. We highlight that relevant conferences and journals on Computer Science are adopting the *Reproducibility Initiative* [108], which consists in assigning reproducibility badges to articles submitting their artifacts for post-publication peer review.

9. Open Challenges and Research Opportunities

As presented in the previous sections, distributed digital infrastructures for Data Analytics and learning are now evolving towards an interconnected ecosystem allowing complex applications to be executed from IoT Edge devices to the HPC Cloud. Therefore, new challenging application scenarios are emerging from a variety of domains such as healthcare, asset monitoring in industry, precision agriculture and smart cities, where processing can no longer rely only on traditional approaches that send all data to centralized datacenters for Data Analytics and Machine Learning. Next, we present some of the relevant challenges and research opportunities to be addressed to enable the Computing Continuum vision.

9.1. Understanding Performance of Application Workflows on the Edge-to-Cloud Continuum

Understanding end-to-end performance on the complex Edge-to-Cloud heterogeneous ecosystem is challenging. Deploying large-scale real-life applications on such infrastructures requires configuring a myriad of system-specific parameters and reconciling many requirements or constraints in terms of hardware capacity, mobility, network efficiency, energy, and data privacy, with low-level infrastructure design choices. One important challenge is to accurately reproduce relevant behaviors of a given application workflow and representative settings of the physical infrastructure underlying this complex continuum.

A first step towards reducing this complexity and enabling the Computing Continuum vision is to enable a **holistic understanding of performance** in such environments. That is, finding a rigorous approach to answering questions like: *(1) How to identify infrastructure bottlenecks across the whole Edge-to-Cloud Continuum? (2) Which system parameters and network configurations impact on the application performance and how? (3) How Edge-to-Cloud hardware configurations impact on the energy consumption and on the processing latency of the application?*

Approaches based on workflow modeling [124] and simulation or emulation, as presented in Table 9, raise some important challenges in terms of specification, modeling, and validation in the context of the Computing Continuum [1, 143]. For example, it is increasingly difficult to model the heterogeneity and volatility of Edge devices or to assess the impact of the inherent complexity of hybrid Edge-Cloud deployments on performance. At this stage, **experimental evaluation** remains the main approach to gain accurate insights on performance metrics and to **build precise approximations** of the expected behavior of large-scale applications on the Computing Continuum, as a **first step prior to modeling**.

A key challenge in this context is to be able to **reproduce in a representative way the application behavior in a controlled environment**, for extensive experiments in a large-enough spectrum of potential configurations of the underlying hybrid Edge-Cloud infrastructure. However, this process is non-trivial due to the multiple combination possibilities of heterogeneous hardware and software resources, as well as, system components for Data Analytics and Machine Learning. Therefore, the Computing Continuum vision calls for novel approaches to **map the real-world application components and dependencies to infrastructure resources**.

Further research efforts shall necessarily focus on the design and implementation of novel methodologies and systems for large-scale experimental evaluation covering the characteristics of hybrid Edge-Cloud infrastructure deployments. Novel systems allowing **the combination of simulation and emulation** systems in addition to supporting **the deployment of state-of-the-art systems** for Data Analytics and Machine Learning **on real-world large-scale testbeds**, considering the same experimental evaluation package, would be relevant to accurately reproduce complex application behaviors.

9.2. Optimizing the Performance of Edge-to-Cloud Application Workflows

The optimization of application workflows on highly distributed and heterogeneous resources is challenging. Real-

world applications deployed on hybrid Edge-to-Cloud infrastructures (*e.g.,* smart factory [152], autonomous vehicles [93], among others) typically need to comply with many **conflicting constraints** related to hardware resource consumption (*e.g.,* GPU memory, CPU power, main memory size, storage size and bandwidth), software components composing the application and requirements such as QoS, security, and privacy [156].

Furthermore, Edge-to-Cloud deployment optimization problems aim at **optimizing metrics** [17, 11] related to performance (*e.g.*, execution time, latency, and throughput), resource usage, energy consumption, financial costs, and quality attributes (*e.g.*, reliability, security, and privacy). Therefore, the parameter settings of the applications and the underlying infrastructure result in a complex multi-infrastructure configuration search space [117].

Therefore, one important challenge is to accurately and efficiently answer questions like: *(1) How to configure the hardware and system components to minimize processing latency and energy consumption? (2) Where should the workflow components be executed across the Edge-to-Cloud Continuum to minimize communication costs and end-to-end latency? (3) How to efficiently autoscale the application resources concerning workload fluctuations and infrastructure changes?*

Such optimization problems are of NP-hard complexity and multi-objective. Furthermore, the environment settings and configuration parameters are extremely vast and their combination of possibilities virtually unlimited [132, 160]. Hence, the process of searching the ideal deployment and configuration of those real-life applications is challenging given the search space complexity: bad choices may result in increased financial expenses during deployment and production phases, decreased processing efficiency and poor user experience [147].

Given these complexities, future research should focus on proposing **novel optimization methodologies** supporting the parallel deployment and evaluation of such complex application workflows on real-life large scale testbeds. The objective is two fold: speeding up the optimization computations, as well as obtaining more accurate results.

Novel approaches should also rely on the development of **fully automated surrogate model building** to mimic and approximate the complex behavior of Edge-to-Cloud workflows and then perform **optimization and sensitivity analysis**. These new solutions may combine computationally tractable optimization techniques [113] such as Bayesian Optimization [136] methods (*e.g.,* Gaussian process (Kriging) [134], Decision Trees [153], Random Forest [23], among others) to build surrogate models; and then combine with techniques such as evolutionary algorithms and swarm intelligence based algorithms (*e.g.,* Genetic Algorithm [95], Differential Evolution [35], Particle Swarm Optimization [45], *etc.*) to perform and speed up the optimization (*e.g.,* to find the optimal deployment configuration using the built surrogate model).

Novel contributions are required for workload characterization and prediction, for autoscaling strategies to enable the efficient scaling of distributed application resources across the Edge-to-Cloud continuum, in response to workload fluctuations and infrastructure changes. Contributions in this context, should be aligned to the complex heterogeneous characteristics of the Computing Continuum paradigm, in terms of: computing resources; network constraints; and application requirements.

9.3. Enabling Intelligence on the Highly Heterogeneous Edge-to-Cloud Continuum

The right selection of Machine Learning techniques for fast and accurate decision making on the highly heterogeneous (in terms of hardware and software) Edge-to-Cloud Continuum requires extensive experiments and evaluations on real-life hybrid infrastructures combining HPC, Cloud, and Edge systems.

The goal is to understand how: (a) infrastructure design choices, (b) optimized learning algorithms with tunable parameters, and (c) the combination of learning paradigms impact on performance metrics such as memory usage, energy consumption, model accuracy, training time, network overhead, application processing latency, among others [119].

This comes down to answering questions like: *(1) How to efficiently deploy complex AI workflows on heterogeneous and distributed infrastructures to reduce training time and improve model accuracy? (2) How to combine Machine Learning paradigms to leverage the massively distributed resources for training across the Edge-to-Cloud Continuum?*

A relevant challenge, worthy of further consideration, is **to understand the performance trade-offs at scale of combining a variety of learning paradigms** such as Reinforcement Learning [155], Deep Learning [59], Online Learning [44], Stream Learning [84], Lifelong Learning [52], Transfer Learning [44], Federated Learning [9], Distributed Learning [164, 166], Multi-task Learning [168], and others.

Approaches leveraging the **incremental evolution of models over time** (*e.g.,* instead of reconstructing new models from scratch) should be considered for streaming data (*e.g.,* instead of batch learning, where the whole training data set should be available for training). They are useful for applications that require high speed processing and analysis of data, and also to avoid the concept drift problem, where predictions become less accurate as time passes, or in cases where the accumulation of large volumes of data is impractical (*e.g.,* due to memory, storage, and processing limitations of Edge devices) [160].

Novel approaches should also leverage **the transfer of knowledge to/from different domains** (*e.g.,* useful when data for training is scarce) and also take advantage of the parallelism and scalability provided by state-of-the-art distributed stream processing systems (*e.g.,* Flink, Spark, *etc.*) combined with Machine Learning paradigms [52] in order to speed up the training and inference time.

Other open challenges [102, 85, 114] include: exploring the massively distributed Edge devices for AI training to achieve scalable and distributed deployment of models on

Edge-to-Cloud infrastructures; applying Neural Architecture Search [47] and Hyperparameter Search [33] to obtain Deep Learning networks that require less resource without losing accuracy; and exploring Knowledge Distillation [30] (*i.e.,* transferring knowledge from a large model to a smaller model without loss of validity) to leverage model deployment on resource-limited devices.

Lastly, further research is needed on novel approaches proposing rigorous methodologies and systems for reproducible experimental evaluations to enable **the performance comparison of AI models and learning paradigms** deployed on large scale and heterogeneous Edge-to-Cloud infrastructures. Such approaches should publish the experimental artifacts on public repositories to allow their reproducibility [30].

These directions are still ongoing and active research areas in the Big Data and AI communities, and as presented in this systematic review, we have not seen reported studies exploring such challenges at large scale on hybrid Edge-to-Cloud infrastructures.

9.4. Supporting Reproducible Analysis of Complex Edge-to-Cloud Workflows

Given the relevance of experimental reproducibility in scientific research to allow the verification of the scientific claims and also to evolve the studies, in addition to the lack of support to the reproducibility of experiments identified in recent articles, as presented in Subsection 7.3, future research efforts should focus on the design and implementation of rigorous methodologies for experimental reproducibility.

Supporting reproducibility of experiments carried out on large scale distributed and heterogeneous infrastructures is non-trivial. The experimental methodology, the artifacts used, and the data captured should provide additional context that more accurately explains the experiment execution and results.

One relevant challenge is to provide mechanisms to allow researchers to **repeat**, **replicate**, and **reproduce** the scientific claims and to help them answer questions like: *(1) What machines/devices were used to execute the entire workflow? (2) What steps were invoked during the workflow execution? (3) Which infrastructure configurations and application parameters produced these results?*

Therefore, novel approaches should focus on enabling the repeatability, replicability and reproducibility of experiments. This requires the **definition of rigorous experimentation methodologies** (*e.g.* well-defined description of: hardware and software resources required to run the experiments and their configurations, network setups, resource interconnections, and workflow execution logic); the **access to the experimental artifacts** (*e.g.* datasets, scripts, libraries, applications, systems, configuration files, among others); and the development of **mechanisms to automatically manage the data derived from experiments**, including: the data provenance capture (*e.g.* runtime configuration of physical machines, software and systems setups, network configurations, *etc.*) and the access to results (*e.g.* log files, metrics collected during execution, monitoring data, code to plot results, among others).

In particular, an important challenge is **the data provenance** capture on such highly heterogeneous and distributed infrastructures. It requires the design and development of novel provenance systems to efficiently capture data from heterogeneous hardware resources ranging from HPC/Cloud servers to resource constrained Edge devices (*e.g.* requires smart data capture strategies to reduce capture overhead) interconnected by different network capabilities (*e.g.* requires provenance data transmission balancing to mitigate the network overhead).

10. Conclusions

In this paper, we did a systematic review of the current state-of-the-art methods to enable intelligence on the Edge-to-Cloud Continuum. First, we discussed the main libraries and frameworks for Machine Learning and Deep Learning inference, centralized training, and distributed training with a focus on the Edge and Cloud. We also presented the main methods for data processing on the Edge, as well as the methods for Big Data stream analytics across the Edge-to-Cloud Continuum.

We reviewed the recent systems, frameworks and architectures that combine Machine Learning and Data Analytics through the main state-of-the-art learning paradigms such as Online Learning, Transfer Learning, Federated Learning, among others, for collaborative Edge-to-Cloud training and decision making. Finally, we discussed experimental research that covers the whole Edge-to-Cloud Continuum with a focus on simulation, emulation, and deployment systems, as well as large-scale experimental testbeds and how the studies included in our systematic review provide support for experiment reproducibility.

There are several open challenges left to realize the Computing Continuum vision. We highlighted the complexity of a holistic understanding of performance of application workflows deployed on the Continuum, as well as the performance optimization of such applications on highly distributed and heterogeneous environments. Many advances are yet required to enable intelligence across the Edge-to-Cloud Continuum in an efficient way. In particular, there is a need for approaches that allow the optimized deployment of complex AI workflows to reduce training time and improve model accuracy, and novel ideas that leverage the massively distributed Edge-to-Cloud resources for fast decision making. Given the lack of support for reproducibility, there is also a need for approaches that help scientists to repeat, replicate, and reproduce the analysis of complex Edge-to-Cloud workflows in a large scale. These challenges can be addressed through novel methodologies, algorithms, systems and frameworks.

Acknowledgments

This work was funded by Inria through the HPC-BigData Inria Challenge (IPL) and by French ANR OverFlow project (ANR-15- CE25-0003).

References

[1] Abreu, D.P., Velasquez, K., Curado, M., Monteiro, E., 2019. A Comparative Analysis of Simulators for the Cloud to Fog Continuum. Simulation Modelling Practice and Theory , 102029.

[2] Adjih, C., Baccelli, E., Fleury, E., Harter, G., Mitton, N., Noel, T., Pissard-Gibollet, R., Saint-Marcel, F., Schreiner, G., Vandaele, J., et al., 2015. Fit iot-lab: A large scale open experimental iot testbed, in: 2015 IEEE 2nd World Forum on Internet of Things (WF-IoT), IEEE. pp. 459–464.

[3] Alencar, B.M., Rios, R.A., Santana, C., Prazeres, C., 2020. Fotstream: A fog platform for data stream analytics in iot. Computer Communications .

[4] Ali, A.H., Abdullah, M.Z., 2018. Recent trends in distributed online stream processing platform for big data: Survey, in: 2018 1st Annual International Conference on Information and Sciences (AiCIS), IEEE. pp. 140–145.

[5] Ali, M., Anjum, A., Rana, O., Zamani, A.R., Balouek-Thomert, D., Parashar, M., 2020. Res: Real-time video stream analytics using edge enhanced clouds. IEEE Transactions on Cloud Computing .

[6] Alli, A.A., Alam, M.M., 2020. The fog cloud of things: A survey on concepts, architecture, standards, tools, and applications. Internet of Things 9, 100177.

[7] Angel, N.A., Ravindran, D., Vincent, P., Srinivasan, K., Hu, Y.C., 2022. Recent advances in evolving computing paradigms: Cloud, edge, and fog technologies. Sensors 22, 196.

[8] Ansari, M.S., Alsamhi, S.H., Qiao, Y., Ye, Y., Lee, B., 2020. Security of distributed intelligence in edge computing: Threats and countermeasures, in: The cloud-to-thing continuum. Palgrave Macmillan, Cham, pp. 95–122.

[9] Aral, A., Erol-Kantarci, M., Brandić, I., 2020. Staleness control for edge data analytics. Proceedings of the ACM on Measurement and Analysis of Computing Systems 4, 1–24.

[10] Asch, M., Moore, T., Badia, R., Beck, M., Beckman, P., Bidot, T., Bodin, F., Cappello, F., Choudhary, A., de Supinski, B., et al., 2018. Big data and extreme-scale computing: Pathways to convergence - toward a shaping strategy for a future software and data ecosystem for scientific inquiry. The International Journal of High Performance Computing Applications 32, 435–479.

[11] Aslanpour, M.S., Gill, S.S., Toosi, A.N., 2020. Performance evaluation metrics for cloud, fog and edge computing: A review, taxonomy, benchmarks and standards for future research. Internet of Things , 100273.

[12] Assefi, M., Behravesh, E., Liu, G., Tafti, A.P., 2017. Big data machine learning using apache spark mllib, in: 2017 ieee international conference on big data (big data), IEEE. pp. 3492–3498.

[13] Atitallah, S.B., Driss, M., Boulila, W., Ghézala, H.B., 2020. Leveraging deep learning and iot big data analytics to support the smart cities development: Review and future directions. Computer Science Review 38, 100303.

[14] Badidi, E., Mahrez, Z., Sabir, E., 2020. Fog computing for smart cities' big data management and analytics: A review. Future Internet 12, 190.

[15] Baldin, I., Chase, J., Xin, Y., Mandal, A., Ruth, P., Castillo, C., Orlikowski, V., Heermann, C., Mills, J., 2016. Exogeni: A multi-domain infrastructure-as-a-service testbed, in: The GENI Book. Springer, pp. 279–315.

[16] Barba, L.A., Thiruvathukal, G.K., 2017. Reproducible Research for Computing in Science Engineering. Computing in Science Engineering 19, 85–87.

[17] Bellendorf, J., Mann, Z.Á., 2020. Classification of optimization problems in fog computing. Future Generation Computer Systems 107, 158–176.

[18] Bendechache, M., Svorobej, S., Takako Endo, P., Lynn, T., 2020. Simulating resource management across the cloud-to-thing continuum: A survey and future directions. Future Internet 12, 95.

[19] Bhat, S.A., Huang, N.F., 2021. Big data and ai revolution in precision agriculture: Survey and challenges. IEEE Access .

[20] Bolze, R., Cappello, F., Caron, E., Dayde, M., Desprez, F., Jeannot, E., Jégou, Y., Lanteri, S., Leduc, J., Melab, N., Mornet, G., Namyst, R., Primet, P., Quétier, B., Richard, O., Talbi, E.G., Touche, I., 2006. Grid'5000: A Large Scale And Highly Reconfigurable Experimental Grid Testbed. International Journal of High Performance Computing Applications 20, 481–494. URL: https://hal.inria.fr/hal-00684943, doi:10.1177/1094342006070078.

[21] Bonawitz, K., Eichner, H., Grieskamp, W., Huba, D., Ingerman, A., Ivanov, V., Kiddon, C., Konečnỳ, J., Mazzocchi, S., McMahan, H.B., et al., 2019. Towards federated learning at scale: System design. arXiv preprint arXiv:1902.01046 .

[22] Bouckaert, S., Vandenberghe, W., Jooris, B., Moerman, I., Demeester, P., 2010. The w-ilab. t testbed, in: International Conference on Testbeds and Research Infrastructures, Springer. pp. 145–154.

[23] Breiman, L., 2001. Random forests. Machine learning 45, 5–32.

[24] Brewer, E.A., 2015. Kubernetes and the path to cloud native, in: Proceedings of the sixth ACM symposium on cloud computing, pp. 167–167.

[25] Brogi, A., Forti, S., 2017. Qos-aware deployment of iot applications through the fog. IEEE Internet of Things Journal 4, 1185–1192. doi:10.1109/JIOT.2017.2701408.

[26] Cai, Z., Li, Q., Li, X., 2017. Elasticsim: A toolkit for simulating workflows with cloud resource runtime auto-scaling and stochastic task execution times. Journal of Grid Computing 15, 257–272.

[27] Caida, July 28, 2021. About caida. https://www.caida.org/about/.

[28] Calheiros, R.N., Ranjan, R., Beloglazov, A., De Rose, C.A., Buyya, R., 2011. Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. Software: Practice and experience 41, 23–50.

[29] Chao, L., Peng, X., Xu, Z., Zhang, L., 2019. Ecosystem of things: Hardware, software, and architecture. Proceedings of the IEEE 107, 1563–1583.

[30] Chen, J., Ran, X., 2019. Deep learning with edge computing: A review. Proceedings of the IEEE 107, 1655–1674.

[31] Chen, Y., Zhao, K., Li, B., Zhao, M., 2019. Exploring the use of synthetic gradients for distributed deep learning across cloud and edge resources, in: 2nd {USENIX} Workshop on Hot Topics in Edge Computing (HotEdge 19).

[32] Cherrueau, R.A., Delavergne, M., van Kempen, A., Lebre, A., Pertin, D., Balderrama, J.R., Simonet, A., Simonin, M., 2021. Enoslib: A library for experiment-driven research in distributed computing. IEEE Transactions on Parallel and Distributed Systems .

[33] Claesen, M., De Moor, B., 2015. Hyperparameter search in machine learning. arXiv preprint arXiv:1502.02127 .

[34] Coutinho, A., Greve, F., Prazeres, C., Cardoso, J., 2018. Fogbed: A rapid-prototyping emulation environment for fog computing, in: 2018 IEEE International Conference on Communications (ICC), IEEE. pp. 1–7.

[35] Das, S., Mullick, S.S., Suganthan, P.N., 2016. Recent advances in differential evolution–an updated survey. Swarm and Evolutionary Computation 27, 1–30.

[36] Dautov, R., Distefano, S., 2020. Stream processing on clustered edge devices. IEEE Transactions on Cloud Computing .

[37] Dautov, R., Distefano, S., Bruneo, D., Longo, F., Merlino, G., Puliafito, A., 2017. Pushing intelligence to the edge with a stream processing architecture, in: 2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), IEEE. pp. 792–799.

[38] Dautov, R., Distefano, S., Bruneo, D., Longo, F., Merlino, G., Puliafito, A., 2018. Data processing in cyber-physical-social systems through edge computing. IEEE Access 6, 29822–29835.

[39] Dautov, R., Distefano, S., Buyya, R., 2019. Hierarchical data fusion for smart healthcare. Journal of Big Data 6, 1–23.

[40] Debauche, O., Mahmoudi, S., Manneback, P., Lebeau, F., 2021. Cloud and distributed architectures for data management in agriculture 4.0: Review and future trends. Journal of King Saud University-Computer and Information Sciences .

[41] Demeester, P., Van Daele, P., Wauters, T., Hrasnica, H., 2016. Fed4fire: the largest federation of testbeds in europe, in: Building the future internet through FIRE, pp. 87–109.

[42] Deng, S., Zhao, H., Fang, W., Yin, J., Dustdar, S., Zomaya, A.Y., 2020. Edge intelligence: The confluence of edge computing and artificial intelligence. IEEE Internet of Things Journal 7, 7457–7469.

[43] Dey, S., Mondal, J., Mukherjee, A., 2019. Offloaded execution of deep learning inference at edge: Challenges and insights, in: 2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), IEEE. pp. 855–861.

[44] Diez-Olivan, A., Del Ser, J., Galar, D., Sierra, B., 2019. Data fusion and machine learning for industrial prognosis: Trends and perspectives towards industry 4.0. Information Fusion 50, 92–111.

[45] Du, K.L., Swamy, M., 2016. Particle swarm optimization, in: Search and optimization by metaheuristics. Springer, pp. 153–173.

[46] Duan, Y., Edwards, J.S., Dwivedi, Y.K., 2019. Artificial intelligence for decision making in the era of big data–evolution, challenges and research agenda. International Journal of Information Management 48, 63–71.

[47] Elsken, T., Metzen, J.H., Hutter, F., 2019. Neural architecture search: A survey. The Journal of Machine Learning Research 20, 1997–2017.

[48] Endo, P.T., Rodrigues, M., Gonçalves, G.E., Kelner, J., Sadok, D.H., Curescu, C., 2016. High availability in clouds: systematic review and research challenges. Journal of Cloud Computing 5, 1–15.

[49] ETP4HPC, April 29, 2020. Etp4hpc strategic research agenda. https://www.etp4hpc.eu/sra.html.

[50] Fafoutis, X., Marchegiani, L., Elsts, A., Pope, J., Piechocki, R., Craddock, I., 2018. Extending the battery lifetime of wearable sensors with embedded machine learning, in: 2018 IEEE 4th World Forum on Internet of Things (WF-IoT), IEEE. pp. 269–274.

[51] FAR-EDGE, July 5, 2021. Far-edge vision. http://www.faredge.eu/.

[52] Fei, X., Shah, N., Verba, N., Chao, K.M., Sanchez-Anguix, V., Lewandowski, J., James, A., Usman, Z., 2019. Cps data streams analytics based on machine learning for cloud and fog computing: A survey. Future Generation Computer Systems 90, 435–450.

[53] Fernández-Cerero, D., Fernández-Montes, A., Jakóbik, A., Kołodziej, J., Toro, M., 2018. Score: Simulator for cloud optimization of resources and energy consumption. Simulation Modelling Practice and Theory 82, 160–173.

[54] Ferro, N., Kelly, D., 2018. SIGIR Initiative to Implement ACM Artifact Review and Badging, in: ACM SIGIR Forum, ACM New York, NY, USA. pp. 4–10.

[55] Fiuczynski, M.E., 2006. Planetlab: overview, history, and future directions. ACM SIGOPS Operating Systems Review 40, 6–10.

[56] Fu, X., Ghaffar, T., Davis, J.C., Lee, D., 2019. Edgewise: a better stream processing engine for the edge, in: 2019 {USENIX} Annual Technical Conference ({USENIX}{ATC} 19), pp. 929–946.

[57] Ghosh, A.M., Grolinger, K., 2020. Edge-cloud computing for internet of things data analytics: Embedding intelligence in the edge with deep learning. IEEE Transactions on Industrial Informatics 17, 2191–2200.

[58] Gill, S.S., Tuli, S., Xu, M., Singh, I., Singh, K.V., Lindsay, D., Tuli, S., Smirnova, D., Singh, M., Jain, U., et al., 2019. Transformative effects of iot, blockchain and artificial intelligence on cloud computing: Evolution, vision, trends and open challenges. Internet of Things 8, 100118.

[59] Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep learning. MIT press.

[60] Grzenda, M., Kwasiborska, K., Zaremba, T., 2020. Hybrid short term prediction to address limited timeliness of public transport data streams. Neurocomputing 391, 305–317.

[61] Grzywaczewski, A., 2017. Training ai for self-driving vehicles: the challenge of scale. Available from Internet: https://devblogs. nvidia. com/training-self-driving-vehicles-challenge-scale .

[62] Guo, P., Hu, B., Hu, W., 2021. Mistify: Automating dnn model porting for on-device inference at the edge., in: NSDI, pp. 705–719.

[63] Gupta, H., Vahid Dastjerdi, A., Ghosh, S.K., Buyya, R., 2017. ifogsim: A toolkit for modeling and simulation of resource management techniques in the internet of things, edge and fog computing environments. Software: Practice and Experience 47, 1275–1296.

[64] Hamdan, S., Ayyash, M., Almajali, S., 2020. Edge-computing architectures for internet of things applications: A survey. Sensors 20, 6441.

[65] Hasenburg, J., Werner, S., Bermbach, D., 2018. Supporting the evaluation of fog-based IoT applications during the design phase, in: Proceedings of the 5th Workshop on Middleware and Applications for the Internet of Things (M4IoT 2018), ACM.

[66] Hauswirth, M., Le-Phuoc, D., 2020. Autonomous rdf stream processing for iot edge devices, in: Semantic Technology: 9th Joint International Conference, JIST 2019, Hangzhou, China, November 25–27, 2019, Proceedings, Springer Nature. p. 304.

[67] Hong, R., Chandra, A., 2019. Dlion: Decentralized distributed deep learning in micro-clouds, in: 11th {USENIX} Workshop on Hot Topics in Cloud Computing (HotCloud 19).

[68] Huang, Z., Lin, K.J., Tsai, B.L., Yan, S., Shih, C.S., 2018. Building edge intelligence for online activity recognition in service-oriented iot systems. Future Generation Computer Systems 87, 557–567.

[69] Kaur, K., Singh, J., Ghumman, N.S., 2014. Mininet as software defined networking testing platform, in: International Conference on Communication, Computing & Systems (ICCCS), pp. 139–42.

[70] Keahey, K., Anderson, J., Zhen, Z., Riteau, P., Ruth, P., Stanzione, D., Cevik, M., Colleran, J., Gunawi, H.S., Hammock, C., Mambretti, J., Barnes, A., Halbach, F., Rocha, A., Stubbs, J., 2020. Lessons learned from the chameleon testbed, in: Proceedings of the 2020 USENIX Annual Technical Conference (USENIX ATC '20). USENIX Association.

[71] Keele, S., et al., 2007. Guidelines for performing systematic literature reviews in software engineering. Technical Report. Citeseer.

[72] Khayyam, H., Javadi, B., Jalili, M., Jazar, R.N., 2020. Artificial intelligence and internet of things for autonomous vehicles, in: Nonlinear approaches in engineering applications. Springer, pp. 39–68.

[73] Kołodziej, J., González-Vélez, H., 2019. High-Performance Modelling and Simulation for Big Data Applications: Selected Results of the COST Action IC1406 cHiPSet. Springer Nature.

[74] Kourtellis, N., Herodotou, H., Grzenda, M., Wawrzyniak, P., Bifet, A., 2021. S2ce: a hybrid cloud and edge orchestrator for mining exascale distributed streams, in: Proceedings of the 15th ACM International Conference on Distributed and Event-based Systems, pp. 103–113.

[75] Kukreja, N., Shilova, A., Beaumont, O., Huckelheim, J., Ferrier, N., Hovland, P., Gorman, G., 2019. Training on the edge: The why and the how, in: 2019 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), IEEE. pp. 899–903.

[76] Kumar, A., Goyal, S., Varma, M., 2017. Resource-efficient machine learning in 2 kb ram for the internet of things, in: International Conference on Machine Learning, PMLR. pp. 1935–1944.

[77] Kumar, D., Ramkumar, A.A., Sindhu, R., Chandra, A., 2019. Decaf: Iterative collaborative processing over the edge, in: 2nd {USENIX} Workshop on Hot Topics in Edge Computing (HotEdge 19).

[78] Lee, J., Singh, J., Azamfar, M., Pandhare, V., 2020. Industrial ai and predictive analytics for smart manufacturing systems, in: Smart Manufacturing. Elsevier, pp. 213–244.

[79] Lera, I., Guerrero, C., Juiz, C., 2019. Yafs: A simulator for iot scenarios in fog computing. IEEE Access 7, 91745–91758.

[80] Li, E., Zeng, L., Zhou, Z., Chen, X., 2019. Edge ai: On-demand accelerating deep neural network inference via edge computing. IEEE Transactions on Wireless Communications 19, 447–457.

[81] Liang, N.Y., Huang, G.B., Saratchandran, P., Sundararajan, N., 2006. A fast and accurate online sequential learning algorithm for feedforward networks. IEEE Transactions on neural networks 17, 1411–1423.

[82] Liaw, R., Liang, E., Nishihara, R., Moritz, P., Gonzalez, J.E., Stoica, I., 2018. Tune: A research platform for distributed model selection and training. arXiv preprint arXiv:1807.05118 .

[83] Liu, F., Tang, G., Li, Y., Cai, Z., Zhang, X., Zhou, T., 2019. A survey on edge computing systems and tools. Proceedings of the IEEE 107, 1537–1562.

[84] Lobo, J.L., Del Ser, J., Bifet, A., Kasabov, N., 2020. Spiking neural networks and online learning: An overview and perspectives. Neural Networks 121, 88–100.

[85] Loghin, D., Cai, S., Chen, G., Dinh, T.T.A., Fan, F., Lin, Q., Ng, J., Ooi, B.C., Sun, X., Ta, Q.T., et al., 2020. The disruptions of 5g on data-driven technologies and applications. IEEE transactions on knowledge and data engineering 32, 1179–1198.

[86] Lu, S., Yao, Y., Shi, W., 2019. Collaborative learning on the edges: A case study on connected vehicles, in: 2nd {USENIX} Workshop on Hot Topics in Edge Computing (HotEdge 19).

[87] Luckow, A., Rattan, K., Jha, S., 2021. Exploring task placement for edge-to-cloud applications using emulation, in: 2021 IEEE 5th International Conference on Fog and Edge Computing (ICFEC), IEEE. pp. 79–83.

[88] L'heureux, A., Grolinger, K., Elyamany, H.F., Capretz, M.A., 2017. Machine learning with big data: Challenges and approaches. Ieee Access 5, 7776–7797.

[89] Mahmood, Z., 2018. Fog Computing: Concepts, Frameworks and Technologies. Springer.

[90] Malik, A.W., Qayyum, T., Rahman, A.U., Khan, M.A., Khalid, O., Khan, S.U., 2020. Xfogsim: A distributed fog resource management framework for sustainable iot services. IEEE Transactions on Sustainable Computing 6, 691–702.

[91] Mayer, R., Graser, L., Gupta, H., Saurez, E., Ramachandran, U., 2017. Emufog: Extensible and scalable emulation of large-scale fog computing infrastructures, in: 2017 IEEE Fog World Congress (FWC), IEEE. pp. 1–6.

[92] Medina, A., Lakhina, A., Matta, I., Byers, J., 2001. Brite: An approach to universal topology generation, in: MASCOTS 2001, Proceedings Ninth International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems, IEEE. pp. 346–353.

[93] Midya, S., Roy, A., Majumder, K., Phadikar, S., 2018. Multi-objective optimization technique for resource allocation and task scheduling in vehicular cloud architecture: A hybrid adaptive nature inspired approach. Journal of Network and Computer Applications 103, 58–84.

[94] Mijuskovic, A., Chiumento, A., Bemthuis, R., Aldea, A., Havinga, P., 2021. Resource management techniques for cloud/fog and edge computing: An evaluation framework and classification. Sensors 21, 1832.

[95] Mirjalili, S., 2019. Genetic algorithm, in: Evolutionary algorithms and neural networks. Springer, pp. 43–55.

[96] Mohammadi, M., Al-Fuqaha, A., Sorour, S., Guizani, M., 2018. Deep learning for iot big data and streaming analytics: A survey. IEEE Communications Surveys & Tutorials 20, 2923–2960.

[97] Mrozek, D., Koczur, A., Małysiak-Mrozek, B., 2020. Fall detection in older adults with mobile iot devices and machine learning in the cloud and on the edge. Information Sciences 537, 132–147.

[98] Mwase, C., Jin, Y., Westerlund, T., Tenhunen, H., Zou, Z., 2022. Communication-efficient distributed ai strategies for the iot edge. Future Generation Computer Systems .

[99] Nair, L.R., Shetty, S.D., Shetty, S.D., 2018. Applying spark based machine learning model on streaming big data for health status prediction. Computers & Electrical Engineering 65, 393–399.

[100] Nandan Jha, D., Alwasel, K., Alshoshan, A., Huang, X., Naha, R.K., Battula, S.K., Garg, S., Puthal, D., James, P., Zomaya, A.Y., et al., 2019. Iotsim-edge: A simulation framework for modeling the behaviour of iot and edge computing environments. arXiv e-prints , arXiv–1910.

[101] Nguyen, G., Dlugolinsky, S., Bobák, M., Tran, V., García, Á.L., Heredia, I., Malík, P., Hluchỳ, L., 2019. Machine learning and deep learning frameworks and libraries for large-scale data mining: a survey. Artificial Intelligence Review 52, 77–124.

[102] Nikouei, S.Y., Chen, Y., Song, S., Choi, B.Y., Faughnan, T.R., 2019. Toward intelligent surveillance as an edge network service (isense) using lightweight detection and tracking algorithms. IEEE Transactions on Services Computing .

[103] Nussbaum, L., 2019. An overview of fed4fire testbeds–and beyond?, in: GEFI-Global Experimentation for Future Internet Workshop.

[104] Ometov, A., Molua, O.L., Komarov, M., Nurmi, J., 2022. A survey of security in cloud, edge, and fog computing. Sensors 22, 927.

[105] ORBIT, P., 2016. Open-access research testbed for next-generation wireless networks.

[106] Pääkkönen, P., Pakkala, D., 2020. Extending reference architecture of big data systems towards machine learning in edge computing environments. Journal of Big Data 7, 1–29.

[107] Pan, S.J., Yang, Q., 2009. A survey on transfer learning. IEEE Transactions on knowledge and data engineering 22, 1345–1359.

[108] Parashar, M., 2022. Eic editorial–advancing reproducibility in parallel and distributed systems research. IEEE Transactions on Parallel & Distributed Systems 33, 2010–2010.

[109] Pathak, A.R., Pandey, M., Rautaray, S., 2018. Construing the big data based on taxonomy, analytics and approaches. Iran Journal of Computer Science 1, 237–259.

[110] Pechlivanidou, K., Katsalis, K., Igoumenos, I., Katsaros, D., Korakis, T., Tassiulas, L., 2014. Nitos testbed: A cloud based wireless experimentation facility, in: 2014 26th International Teletraffic Congress (ITC), IEEE. pp. 1–6.

[111] Pérez, J.L., Gutierrez-Torre, A., Berral, J.L., Carrera, D., 2018. A resilient and distributed near real-time traffic forecasting application for fog computing environments. Future Generation Computer Systems 87, 198–212.

[112] Peuster, M., Kampmeyer, J., Karl, H., 2018. Containernet 2.0: A rapid prototyping platform for hybrid service function chains, in: 2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft), IEEE. pp. 335–337.

[113] Pham, D., Karaboga, D., 2012. Intelligent optimisation techniques: genetic algorithms, tabu search, simulated annealing and neural networks. Springer Science & Business Media.

[114] Prabhu, C., 2019. Fog Computing, Deep Learning and Big Data Analytics-Research Directions. Springer.

[115] Qayyum, T., Malik, A.W., Khattak, M.A.K., Khalid, O., Khan, S.U., 2018. Fognetsim++: A toolkit for modeling and simulation of distributed fog environment. IEEE Access 6, 63570–63583.

[116] RADICAL-DREAMER, Feb 12, 2022. Radical-dreamer: Dynamic runtime and execution adaptive middleware emulator (rd). https://github.com/radical-project/radical.dreamer/.

[117] Ranjan, R., Rana, O., Nepal, S., Yousif, M., James, P., Wen, Z., Barr, S., Watson, P., Jayaraman, P.P., Georgakopoulos, D., et al., 2018. The next grand challenges: Integrating the internet of things and data science. IEEE Cloud Computing 5, 12–26.

[118] Rao, T.R., Mitra, P., Bhatt, R., Goswami, A., 2019. The big data system, components, tools, and technologies: a survey. Knowledge and Information Systems 60, 1165–1245.

[119] Rocha Neto, A.F., Delicato, F.C., Batista, T.V., Pires, P.F., 2020. Distributed machine learning for iot applications in the fog. Fog Computing: Theory and Practice , 309–345.

[120] Rong, G., Xu, Y., Tong, X., Fan, H., 2021. An edge-cloud collaborative computing platform for building aiot applications efficiently .

[121] Rosendo, D., Costan, A., Antoniu, G., Simonin, M., Lombardo, J.C., Joly, A., Valduriez, P., 2021a. Reproducible performance optimization of complex applications on the edge-to-cloud continuum. arXiv preprint arXiv:2108.04033 .

[122] Rosendo, D., Costan, A., Antoniu, G., Valduriez, P., 2021b. E2clab: Reproducible analysis of complex workflows on the edge-to-cloud continuum, in: IEEE 35th International Parallel and Distributed Processing Symposium (IPDS 2021).

[123] Rosendo, D., Silva, P., Simonin, M., Costan, A., Antoniu, G., 2020. E2clab: Exploring the computing continuum through repeatable, replicable and reproducible edge-to-cloud experiments, in: 2020 IEEE International Conference on Cluster Computing (CLUSTER), IEEE. pp. 176–186.

[124] Sadiq, S., Orlowska, M., Sadiq, W., Foulger, C., 2004. Data Flow and Validation in Workflow Modelling, in: Proceedings of the 15th

Australasian database conference-Volume 27, pp. 207–214.

[125] Samie, F., Bauer, L., Henkel, J., 2019. From cloud down to things: An overview of machine learning in internet of things. IEEE Internet of Things Journal 6, 4921–4934.

[126] Sanchez, L., Muñoz, L., Galache, J.A., Sotres, P., Santana, J.R., Gutierrez, V., Ramdhany, R., Gluhak, A., Krco, S., Theodoridis, E., et al., 2014. Smartsantander: Iot experimentation over a smart city testbed. Computer Networks 61, 217–238.

[127] Sankaranarayanan, S., Rodrigues, J.J., Sugumaran, V., Kozlov, S., et al., 2020. Data flow and distributed deep neural network based low latency iot-edge computation model for big data environment. Engineering Applications of Artificial Intelligence 94, 103785.

[128] Sarabia-Jácome, D., Lacalle, I., Palau, C.E., Esteve, M., 2019. Efficient deployment of predictive analytics in edge gateways: Fall detection scenario, in: 2019 IEEE 5th World Forum on Internet of Things (WF-IoT), IEEE. pp. 41–46.

[129] Sarabia-Jácome, D., Usach, R., Palau, C.E., Esteve, M., 2020. Highly-efficient fog-based deep learning aal fall detection system. Internet of Things 11, 100185.

[130] Sergeev, A., Del Balso, M., 2018. Horovod: fast and easy distributed deep learning in tensorflow. arXiv preprint arXiv:1802.05799 .

[131] Sezer, O.B., Dogdu, E., Ozbayoglu, A.M., 2017. Context-aware computing, learning, and big data in internet of things: a survey. IEEE Internet of Things Journal 5, 1–27.

[132] Sharma, S.K., Wang, X., 2017. Live data analytics with collaborative edge and cloud processing in wireless iot networks. IEEE Access 5, 4621–4635.

[133] Shin, H., Lee, J.K., Kim, J., Kim, J., 2017. Continual learning with deep generative replay. arXiv preprint arXiv:1705.08690 .

[134] Simpson, T.W., Mauery, T.M., Korte, J.J., Mistree, F., 2001. Kriging models for global approximation in simulation-based multidisciplinary design optimization. AIAA journal 39, 2233–2241.

[135] Sitton Candanedo, I.X., et al., 2020. Geca: A global edge computing architecture .

[136] Snoek, J., Larochelle, H., Adams, R.P., 2012. Practical bayesian optimization of machine learning algorithms. arXiv preprint arXiv:1206.2944 .

[137] Sonmez, C., Ozgovde, A., Ersoy, C., 2018. Edgecloudsim: An environment for performance evaluation of edge computing systems. Transactions on Emerging Telecommunications Technologies 29, e3493.

[138] Spataru, A., 2021. A review of blockchain-enabled fog computing in the cloud continuum context. Scalable Computing: Practice and Experience 22, 463–468.

[139] Sreerangaraju, S., 2020. Emulation vs. simulation. https://www.perfecto.io/blog/emulation-vs-simulation.

[140] Stodden, V., Miguez, S., 2013. Best Practices for Computational Science: Software Infrastructure and Environments for Reproducible and Extensible Research. Available at SSRN 2322276 .

[141] Struye, J., Braem, B., Latré, S., Marquez-Barja, J., 2018. The citylab testbed—large-scale multi-technology wireless experimentation in a city environment: Neural network-based interference prediction in a smart city, in: IEEE INFOCOM 2018-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), IEEE. pp. 529–534.

[142] Sutton, R.S., Barto, A.G., 2018. Reinforcement learning: An introduction. MIT press.

[143] Svorobej, S., Takako Endo, P., Bendechache, M., Filelis-Papadopoulos, C., Giannoutakis, K.M., Gravvanis, G.A., Tzovaras, D., Byrne, J., Lynn, T., 2019. Simulating fog and edge computing scenarios: An overview and research challenges. Future Internet 11, 55.

[144] Talagala, N., Sundararaman, S., Sridhar, V., Arteaga, D., Luo, Q., Subramanian, S., Ghanta, S., Khermosh, L., Roselli, D., 2018. {ECO}: Harmonizing edge and cloud with ml/dl orchestration, in: {USENIX} Workshop on Hot Topics in Edge Computing (HotEdge 18).

[145] Ulusar, U.D., Ozcan, D.G., Al-Turjman, F., 2020. Open source tools for machine learning with big data in smart cities, in: Smart Cities Performability, Cognition, & Security. Springer, pp. 153–168.

[146] Vanhove, T., Van Seghbroeck, G., Wauters, T., De Turck, F., Vermeulen, B., Demeester, P., 2015. Tengu: An experimentation platform for big data applications, in: 2015 IEEE 35th International Conference on Distributed Computing Systems Workshops, IEEE. pp. 42–47.

[147] Verma, P., Fatima, S., 2020. Smart healthcare applications and real-time analytics through edge computing, in: Internet of Things Use Cases for the Healthcare Industry. Springer, pp. 241–270.

[148] Verma, S., Kawamoto, Y., Fadlullah, Z.M., Nishiyama, H., Kato, N., 2017. A survey on network methodologies for real-time analytics of massive iot data and open research issues. IEEE Communications Surveys & Tutorials 19, 1457–1477.

[149] Vermesan, O., Bahr, R., Ottella, M., Serrano, M., Karlsen, T., Wahlstrøm, T., Sand, H.E., Ashwathnarayan, M., Gamba, M.T., 2020. Internet of robotic things intelligent connectivity and platforms. Frontiers in Robotics and AI 7, 104.

[150] Véstias, M.P., Duarte, R.P., de Sousa, J.T., Neto, H.C., 2020. Moving deep learning to the edge. Algorithms 13, 125.

[151] Wang, D., Chen, D., Song, B., Guizani, N., Yu, X., Du, X., 2018. From iot to 5g i-iot: The next generation iot-based intelligent algorithms and 5g technologies. IEEE Communications Magazine 56, 114–120.

[152] Wang, J., Li, D., 2018. Adaptive computing optimization in software-defined network-based industrial internet of things with fog computing. Sensors 18, 2509.

[153] Wang, X., Chen, B., Qian, G., Ye, F., 2000. On the optimization of fuzzy decision trees. Fuzzy Sets and Systems 112, 117–125.

[154] Wette, P., Dräxler, M., Schwabe, A., Wallaschek, F., Zahraee, M.H., Karl, H., 2014. Maxinet: Distributed emulation of software-defined networks, in: 2014 IFIP Networking Conference, IEEE. pp. 1–9.

[155] Wiering, M.A., Van Otterlo, M., 2012. Reinforcement learning. Adaptation, learning, and optimization 12.

[156] Xia, Y., Etchevers, X., Letondeur, L., Lebre, A., Coupaye, T., Desprez, F., 2018. Combining heuristics to optimize and scale the placement of iot applications in the fog, in: 2018 IEEE/ACM 11th International Conference on Utility and Cloud Computing (UCC), IEEE. pp. 153–163.

[157] Xian, G., 2020. Parallel machine learning algorithm using fine-grained-mode spark on a mesos big data cloud computing software framework for mobile robotic intelligent fault recognition. IEEE Access 8, 131885–131900.

[158] Xiao, W., Xue, J., Miao, Y., Li, Z., Chen, C., Wu, M., Li, W., Zhou, L., 2017. Tux^2: Distributed graph computation for machine learning, in: 14th {USENIX} symposium on networked systems design and implementation ({NSDI} 17), pp. 669–682.

[159] Xiong, Y., Sun, Y., Xing, L., Huang, Y., 2018. Extend cloud to edge with kubeedge, in: 2018 IEEE/ACM Symposium on Edge Computing (SEC), IEEE. pp. 373–377.

[160] Xu, H., Yu, W., Griffith, D., Golmie, N., 2018. A survey on industrial internet of things: A cyber-physical systems perspective. IEEE Access 6, 78238–78259.

[161] Xu, L., Venkataraman, S., Gupta, I., Mai, L., Potharaju, R., 2021. Move fast and meet deadlines: Fine-grained real-time stream processing with cameo, in: 18th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 21), pp. 389–405.

[162] Xu, M., Liu, J., Liu, Y., Lin, F.X., Liu, Y., Liu, X., 2019. A first look at deep learning apps on smartphones, in: The World Wide Web Conference, pp. 2125–2136.

[163] Yang, S., 2017. Iot stream processing and analytics in the fog. IEEE Communications Magazine 55, 21–27.

[164] Yousefpour, A., Fung, C., Nguyen, T., Kadiyala, K., Jalali, F., Niakanlahiji, A., Kong, J., Jue, J.P., 2019. All one needs to know about fog computing and related edge computing paradigms: A complete survey. Journal of Systems Architecture 98, 289–330.

[165] Zhang, C., Patras, P., Haddadi, H., 2019. Deep learning in mobile and wireless networking: A survey. IEEE Communications surveys

& tutorials 21, 2224–2287.

[166] Zhang, M., Zhang, F., Lane, N.D., Shu, Y., Zeng, X., Fang, B., Yan, S., Xu, H., 2020. Deep learning in the era of edge computing: Challenges and opportunities. Fog Computing: Theory and Practice , 67–78.

[167] Zhang, X., Wang, Y., Shi, W., 2018. pcamp: Performance comparison of machine learning packages on the edges, in: {USENIX} Workshop on Hot Topics in Edge Computing (HotEdge 18).

[168] Zhang, Y., Yang, Q., 2017. A survey on multi-task learning. arXiv preprint arXiv:1707.08114 .

[169] Zhou, L., Wen, H., Teodorescu, R., Du, D.H., 2019. Distributing deep neural networks with containerized partitions at the edge, in: 2nd {USENIX} Workshop on Hot Topics in Edge Computing (HotEdge 19).

[170] Zhou, Z., Wu, B., Liang, Z., Sun, G., Xu, C., Luo, G., 2020. Saface: Towards scenario-aware face recognition via edge computing system, in: 3rd {USENIX} Workshop on Hot Topics in Edge Computing (HotEdge 20).