IoT and Fog Analytics for Industrial Robot Applications

Anders Lager[1,2], Alessandro Papadopoulos[2], Thomas Nolte[2]
[1]ABB AB, Västerås, Sweden
[2]Mälardalen University, Västerås, Sweden

***Abstract*—The rapid development of IoT, cloud and fog computing has increased the potential for developing smart services for IoT devices. Such services require not only connectivity and high computing capacity, but also fast response time and throughput of inferencing results. In this paper we present our ongoing work, investigating the potential for implementing smart services in the context of industrial robot applications with focus on analytic inferencing on fog and cloud computing platforms. We review different use cases that we have found in the literature and we divide them into two suggested categories, "distributed deep models" and "distributed interconnected models". We analyze the characteristics of IoT data in industrial robot applications and present two concrete use cases of smart services where inferencing in a fog and a cloud architecture, respectively, is needed. We also reason about important considerations and design decisions for the development process of analytic services.**

I. Introduction

During the last years, the usage of deep learning for big data analysis has expanded tremendously [1]. Deep learning is based on layered algorithms with numerous parameters that are used to process various types of input data in order to generate output data for regression or classification. Since training and inferring with these algorithms require high computing capacity, offloading such computation to remote servers is the natural choice — should response time or throughput not be a critical issue. The increased device connectivity has paved the way for Internet of Things (IoT) where an ever increasing number of devices can be connected to the internet [2]. IoT devices constitute vast sources of data that can be processed for various purposes to create new services or improved services with more intelligence.

Deep learning analytics [1] is used in a number of application domains, e.g., Smart Home, Smart City, Industry, Agriculture and Retail to mention a few. Deep learning provides a toolbox to implement services in these application domains. These services will often include one or more basic services such as image recognition, voice and speech recognition, localization, physiological detection, psychological detection, security and privacy. The basic services are often based on deep learning and can be used as building blocks for the application specific services.

Industrial robot applications, i.e. industrial robots performing processing tasks, e.g., picking, palletizing, drilling, assembly or grinding, can be used for various tasks in most of these application domains. As connected IoT devices, they are capable to generate a lot of data originating from the robots, external sensors (e.g., cameras), process equipment and potentially also from parts to be processed. The recent development of fog architectures [3] has brought analytic computing closer to the data sources, i.e., the IoT devices, thereby reducing response times. Preprocessing of input data from IoT devices in fog nodes also decreases the network load, since the amount of data being transferred to the cloud for further processing is reduced.

Infrastructures for fog and cloud services with deep learning support, e.g., [4], are in constant progress and there is a huge potential to create additional value for industrial robot applications. Deep learning analytics can be used to improve robot applications in many different ways and enable completely new services. However, different use cases will have very different requirements on, e.g., response times and throughput. The main challenge addressed in this work is how to create new value by developing new robot application services with deep learning analytics that fulfill use case requirements on, e.g., computational accuracy, response time and throughput.

In this paper, we investigate the data characteristics for industrial robot applications in an IoT Big Data perspective. We present two different use cases of deep learning analytics and discuss their requirements in a fog/cloud architecture. We review different approaches proposed in the literature, for performing distributed analytics with fog architectures. Here we have identified two different categories of such approaches; *distributed deep models* and *distributed interconnected models*. Finally, we discuss design aspects for deep learning analytics with deployment into fog computing platforms.

Section II is a review of different use cases that we have found in the literature, where analytics is performed in a fog and cloud environment. Section III discusses the IoT big data characteristics for an industrial robot application. Section IV provides two examples of concrete use cases where fog or cloud analytics can be used to improve industrial robot applications. Section V discusses design aspects for analytic services in industrial robot applications. Finally, Section VI concludes the paper and outlines current work-in-progress along with future research directions.

II. Related work

The following sections list different use cases that we have found in the literature, covering different approaches

This work was funded by The Knowledge Foundation (KKS), project ARRAY, and by The Swedish Foundation for Strategic Research (SSF).

concerning how to distribute analytic inferencing over fog and cloud architectures. The works are separated in two suggested categories, i.e., *distributed deep models* and *distributed interconnected models*.

a) Distributed deep models: This category separates the layers of a deep model onto different nodes. During inferencing, the first layers are executed by node 1. This node takes input from the IoT devices and generates intermediate output from the first set of layers. The intermediate output from node 1 becomes input data for node 2, which generates output from the second set of layers etc. The last layers on the last node generates the final output, e.g., a classification of an image. The nodes may be distributed over the fog, the cloud or both.

In one experimental setup, a Convolutional Neural Network (CNN) is used to classify images in a fog architecture [5]. The network layers are split into two parts that are deployed on two different nodes. The raw input data in the form of image files are processed by the lower layers of the CNN at the first node to generate an intermediate output. The output is transferred to the second node for further processing by the higher layers of the CNN to generate a final output with inferencing results. Best throughput, i.e., number of analyzed images per second, is achieved if the complexity of the two model parts are equal. On the other hand, the communication overhead is reduced if the first node gets a bigger part of the model. However, the total response time for analyzing one image is not evaluated or compared, e.g, with single node execution.

The need to minimize response times was partly addressed by [6]. They use a CNN to perform machine vision inspection of parts in the manufacturing industry. The CNN is deployed as a Distributed Deep Neural Network [7] into a fog and cloud architecture. The lower layers are computed on the fog architecture and the higher layers are computed in the cloud. To improve the response time, an early exit branch [8] onto the lower layers gives preliminar inspection results. This early exit branch is jointly trained with the final exit branch and provides outputs for the same inference parameters but with a lower accuracy. If the early exit inspection results meets application requirements in terms of accuracy, they can be used without waiting for continuous processing in the cloud, thereby significantly reducing the response time.

b) Distributed interconnected models: For this category, data from IoT devices are processed in several steps, where each step uses one analytic model to calculate output data that represents some known properties. This is different from intermediate layers in a deep model where the output data, the "features", can not be directly interpreted to any concrete information. To increase throughput and reduce communication overhead, the analytic models are distributed in a sequence over different nodes in the fog and the cloud. In [5], a crowdedness detection application is presented that is split into three processing steps: image collection, face recognition and a crowdedness monitor. In [9], an investigation of a use case for a smart shopping mall is presented. Video cameras are monitoring the entrances of the shops in the mall. The cameras generate video data that are analyzed by face detection analytics on multiple first level fog nodes. The face information is sent to a second level fog node that analyzes detected faces and generates age and gender estimations. The age and gender estimations are sent to the cloud where additional analytic processing generates appropriate advertisements to be displayed on electronic billboards in the shopping mall.

A pipeline for deep learning processing over an architecture with three tiers, i.e. edge, cloudlet and cloud is proposed in [10]. The data processing is made in consecutive stages that can be distributed over the nodes in the different tiers. Each stage can also potentially be parallelized. A running example is the processing of video images that can be separated in consecutive stages: video loading, video decoding, motion detection, video frame enhancement, video frame scaling, object detection and object recognition. Experiments show how the throughput is improved when these stages are distributed in different ways. The pipeline also serves the purpose of overcoming limitations in network load by reducing the amount of data that is communicated to the nodes that handle the later stages.

III. Data characteristics

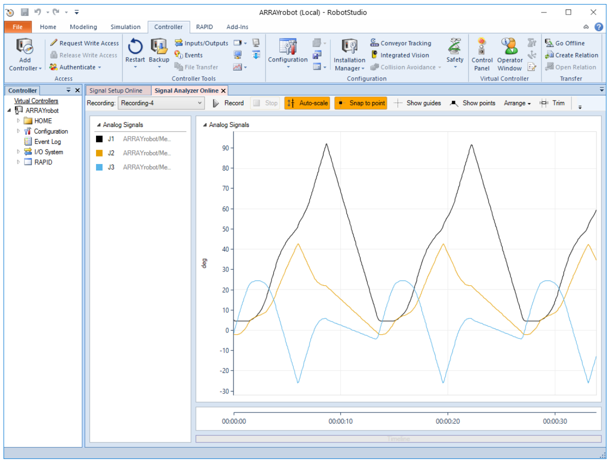
IoT Big Data can be characterized [1] by 6 V's, i.e., Volume, Velocity, Variety, Veracity, Variability, and Value. As analyzed below, the data generated from industrial robot applications matches these characteristics.

Volume. This means a high volume of data is generated. Industrial robot applications often runs 24/7 and will accumulate a lot of data over time.

Velocity. This means data will not only be generated in large volumes, but also with a high rate. Industrial robot applications will not only generate data at discrete events, but also continuously with high frequencies, e.g., an industrial robot from ABB can generate sensor feedback continuously every 4 ms for the built-in arm servo control [11]. A camera may generate images continuously or at discrete events, e.g., only when an object appears.

Variety. This means data will be generated in different forms and types. Industrial robot applications generate data from heterogeneous devices, e.g., robots, sensors and different process equipment. The data from the different devices will also be heterogeneous, e.g., in form of text or images, and will represent various properties, e.g., sensor readings. The variety of data is illustrated by robot data of joint position references for axes 1-3 in Fig. 1 and an image taken by a camera of chips with printed characters in Fig. 2.

Veracity. This means data is consistent and can be trusted. The data from industrial robot applications are generated from trusted devices that are installed to perform predefined tasks, e.g., in a production facility. The quality of the devices is expected to be high compared to consumer products, since they fulfill a number of industry standards and regulations regarding different relevant aspects, e.g., safety, Electro Magnetic Compatibility (EMC), Ingress Protection (IP) and cyber security. They are also dimensioned for a long life, operating 24 hours per day, 7 days per week (24/7) with low failure rates.



Recording: Recording-4

Fig. 1. Joint position references for axes 1-3 from a robot controller.

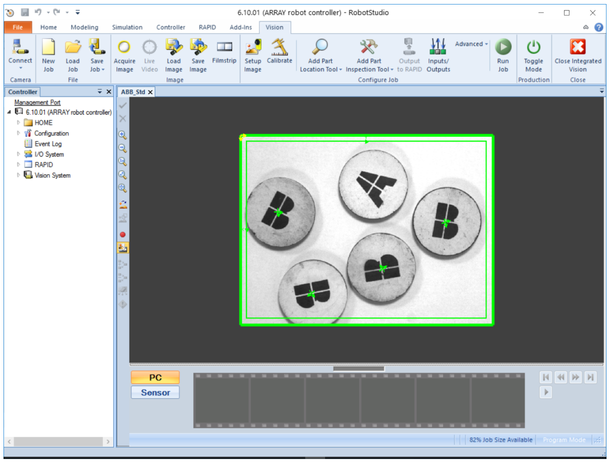


Fig. 2. Camera image with detected characters.

Variability. This means data is generated with different rates. The heterogeneous devices in industrial robot applications will generate data with different rates, both continuously with certain frequencies and at discrete events.

Value. This means the data has a value that can be transformed to improved or new valuable services for robot applications. The use cases presented in the next section gives concrete examples on how value can be created from the data. Typically, the end customer, i.e., the owner of the industrial robot application installation, also owns the generated data. The customer needs an incentive, e.g., improved performance or reliability, to share data with, e.g., the supplier of the industrial robots.

IV. Use Cases

This section presents two use cases with very different requirements and characteristics for industrial robot applications where data can be turned into *Value* by analytic computations in the fog or the cloud. Both these example may be realized with assistance from deep learning models.

A. *Predictive maintenance*

Predictive maintenance, or Condition Based Monitoring (CBM) [12], can be used to diagnose failure states and prognosticate the Remaining Useful Life of components and devices in industrial robot applications. Sensor data recorded from failing scenarios can be used to train a deep model to estimate the remaining time until a certain failure will occur. As part of CBM, inference is performed on real time production data. The output data from the deep model can be used to plan and schedule appropriate maintenance activities, e.g. replacement of parts, in time before a failure state occurs.

To enable CBM, it is necessary to select the parameters to be monitored [13]. For industrial robot applications, some parameters are available in every system since they support basic system functions, e.g., motor current for robot axes. Other parameters are available in a subset of systems, e.g., images from machine vision equipment. To predict some failure states, additional sensors may be required for this purpose and if so, they need to be reliable and the additional cost must be motivated by the consequences of having a failure. As an example, the selected data can be continuous, high frequency measurements of motor currents, positions and speeds for each axis of an industrial robot. A predicted failure can be, e.g., a gear box breakdown and the corresponding planned maintenance action can be an exchange of gear box oil and seal.

For the predictive maintenance use case, the inference time of the deep model is not a critical factor. The accuracy and the time horizon of the model predictions can be expected to be more important, considering that a maintenance action from a service technician may require a few days waiting time. Inferencing can be run using cloud computing and the result can be monitored by a centralized system to schedule actions for service technicians. Using fog computing to speed up the analysis would make little sense.

B. *Reactive replanning*

Industry is recently transitioning from traditional caged robots working in fairly static environments towards more collaborative robots working physically closer to humans and other actors without separating fences. The environment is more dynamic and less predictable, and entails an increased risk of having a failure when following a predetermined plan of robot movements and process interactions.

To improve efficiency and robustness while preserving safety, industrial robot applications need to become reactive by design to handle unforeseen events, e.g., grasping failures or unexpected movements of humans. In [14], an architecture is proposed addressing *reactive replanning* of industrial robot applications. As part of this architecture, a *supervisor* component performs continuous supervision of both the robot's plan and the predicted movement of objects and actors in it's surrounding environment, see Fig. 3.

Whenever a conflict is detected that may cause a failure, or an opportunity is detected for improving the plan to a more efficient one, the *supervisor* component activates a reactive

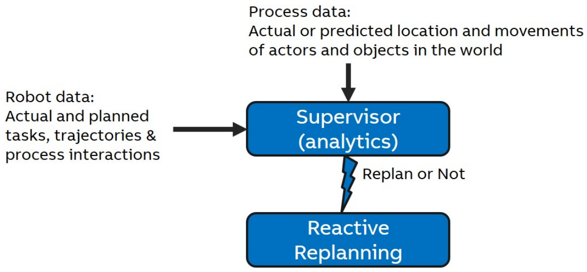

Process data:
Actual or predicted location and movements of actors and objects in the world
Robot data:
Actual and planned tasks, trajectories & process interactions
Supervisor (analytics)
Replan or Not
Reactive Replanning

Fig. 3. Supervisor component.

replanning event. Then, a new plan is generated by a *planning* component, that replaces the current one in a reactive way.

The *supervisor* needs to receive data originating from different sources, i.e., robot controllers, data bases and cameras/sensors. The analytic computations of a *supervisor* component can become quite heavy, due to the high volume and velocity of data to be processed. Required computation rate depends on the speed of the robot, the speed of objects and other actors, the accuracy requirements of the processing of parts and the desired overall system reactiveness to avoid collisions, correct for failures or find more efficient paths or actions. Considering these factors, it is suggested that the *supervisor* is offloaded to a fog architecture. The required response time of the *supervisor* is less than 0.1s, making cloud computing a less feasible solution.

V. Design aspects and Work-in-Progress

Development of new and improved services for industrial robot applications, using deep learning analytics deployed into fog and cloud computing platforms, require a number of important considerations and design decisions. Among them:

- The requirements of the use case, e.g., accuracy, response time and throughput.
- What data is needed and what characteristics (6 Vs) of this data is needed.
- What are the requirements of sensors and other IoT devices that shall provide the data.
- How the computational hardware architecture shall be designed. For example, availability, computational capacity and organisation of fog and cloud nodes.
- Required networking capacity.
- What types of hardware to use. To accelerate deep learning analytics, a number of hardwares specialized in accelerating the inferencing and training of deep neural networks have been developed recently [15]. In general, they are based on parallel computing using different processing units, e.g., CPUs, GPUs, ASICs and FPGAs and they leverage from the inherent parallelism of deep models.
- How analytic models shall be designed to reach the required accuracy.
- How analytic models shall interact to generate desired results.
- Distribution of deep models, e.g., deciding the cutting points between layers and where to deploy them.
- Distribution of interconnected models, e.g., how to define stages of execution and decide where to deploy them.

Our work-in-progress target finding efficient and accurate solutions to the above considerations and design decisions.

VI. Conclusions and future work

In this paper we have presented out ongoing work exploring the potential for creating new and improved services for industrial robot applications by using analytic computations in fog and cloud computing platforms. We have analyzed the characteristics of IoT data in industrial robot applications and provided two concrete use cases. We have reviewed different approaches that we have found in the literature, to perform analytic computations in fog and cloud architectures, and we have divided them into two categories, i.e., *distributed deep models* and *distributed interconnected models*. Finally, we have discussed design aspects for analytic services. Along with current work-in-progress, one potential area of future research is the development of tools that can help to simplify this process. In addition we will address the realization of reactive replanning in a fog architecture.

References

[1] M. Mohammadi *et al.*, “Deep learning for IoT big data and streaming analytics: A survey,” *IEEE Comm. Surv. Tut.*, vol. 20, no. 4, pp. 2923–2960, 2018.

[2] I. Lee and K. Lee, “The Internet of Things (IoT): Applications, investments, and challenges for enterprises,” *Business Horizons*, vol. 58, no. 4, pp. 431–440, 2015.

[3] Cisco. (2015) Fog computing and the internet of things: Extend the cloud to where the things are. [Online]. Available: https://www.cisco.com/c/dam/en_us/solutions/trends/iot/docs/computing-overview.pdf

[4] D. Chappel. (2015) Introducing azure machine learning.

[5] P. Tsai *et al.*, “Distributed analytics in fog computing platforms using tensorflow and kubernetes,” in *Asia-Pacific Network Operations and Management Symp. (APNOMS)*, 2017, pp. 145–150.

[6] L. Li *et al.*, “Deep learning for smart industry: Efficient manufacture inspection system with fog computing,” *IEEE Trans. on Ind. Inf.*, vol. 14, no. 10, pp. 4665–4673, 2018.

[7] S. Teerapittayanon *et al.*, “Distributed deep neural networks over the cloud, the edge and end devices,” in *IEEE Int. Conf. on Distr. Comp. Syst. (ICDCS)*, 2017, pp. 328–339.

[8] ——, “Branchynet: Fast inference via early exiting from deep neural networks,” in *Int. Conf. on Pattern Recognition (ICPR)*, 2016, pp. 2464–2469.

[9] K. Lun Cai and F. Joseph Lin, “Distributed artificial intelligence enabled by onem2m and fog networking,” in *IEEE Conf. on Standards for Comm. and Netw. (CSCN)*, 2018, pp. 1–6.

[10] M. Ali *et al.*, “Edge enhanced deep learning system for large-scale video stream analytics,” in *IEEE Int. Conf. on Fog and Edge Comp. (ICFEC)*, 2018, pp. 1–10.

[11] A. Blomdell *et al.*, “Extending an industrial robot controller: implementation and applications of a fast open sensor interface,” *IEEE Robotics Automation Magazine*, vol. 12, no. 3, pp. 85–94, 2005.

[12] J.-H. Shin and H.-B. Jun, “On condition based maintenance policy,” *Journal of Comp. Design and Eng.*, vol. 2, no. 2, pp. 119–127, 2015.

[13] A. Tsang, “Condition-based maintenance: Tools and decision making,” *Journal of Quality in Maintenance Engineering*, vol. 1, pp. 3–17, 1995.

[14] A. Lager *et al.*, “Towards reactive robot applications in dynamic environments,” in *IEEE Int. Conf. on Emerging Tech. and Factory Automation (ETFA)*, 2019, pp. 1603–1606.

[15] K. Abdelouahab *et al.*, “Accelerating CNN inference on FPGAs: A survey,” 2018.