Toward Evaluation of Deployment Architecture of ML-based Cyber-Physical Systems

Lucas Gisselaire, Florian Cario, Quentin Guerre-berthelot,
Bastien Zigmann, Lydie du Bousquet
University of Grenoble Alpes
CNRS, Grenoble INP, LIG 38000 Grenoble, France
lucas.gisselaire@etu.univ-grenoble-alpes.fr
lydie.du-bousquet@univ-grenoble-alpes.fr

Masahide Nakamura
Kobe University, Riken AIP
1-1 Rokkodai-cho, Nada-ku, Kobe 657-8501, Japan
1-4-1 Nihon-bashi, Chuo-ku, Tokyo 103-0027, Japan
masa-n@cs.kobe-u.ac.jp

***Abstract*—In this position paper, we explore a way to assess deployment architectures for intelligent Cyber-Physical Systems (iCPS). Our long-term goal is to establish a method that allows a software architect of iCPS to choose an appropriate deployment architecture, based on requirements on quality, cost and security. As the first step towards the goal, this paper first presents an abstract model of iCPS to capture the possible deployment configurations. We then propose a method that assesses cost-related attributes for the configurations. Finally, we investigate security threats within different configurations, to help the engineer to achieve the configuration choice.**

***Index Terms*—Assessment Model, Software Engineering, Software Architecture, Cyber-Physical Systems, Machine Learning**

I. Introduction

Nowadays, more and more software systems and solutions are making use of *Machine Learning (ML)* to make complex decisions towards the end-goal of undertaking some actions. *Cyber-Physical Systems (CPS)* (integration of computation, networking and physical processes) are impacted by this trend. Indeed, thanking to emerging Internet of Things (IoT), a huge amount of data can be collected from the physical world, and fed to ML algorithms to implement *intelligent* CPS (we call iCPS). The trend of iCPS is spread to various areas, including health, energy, and transportation [1]–[3].

To implement an iCPS application with a (supervised) ML algorithm, it is necessary to train a *prediction model*. This training usually requires a large amount of memory and high-performance resources, as well as complex libraries. Therefore, a reasonable solution to perform the model development is to use a *ML platform* on a cloud (such as Microsoft Azure, Google AI platform, Amazon AWS platform). The trained prediction model is then deployed as a Web service within a cloud. On the other hand, the software component that uses the prediction model is deployed on the *edge* side. The component takes input data from a physical space, and makes intelligent decisions based on the prediction derived by the model.

However, such a deployment configuration is not always best, due to a variety of requirements of individual iCPS, such as security, latency, and cost. Data might contain personal and confidential information, making cloud computing sensitive without special pre-treatment. After the model development, moving computation closer to the physical space could reduce latency and overhead, which is especially important for iCPS with realtime requirements (such as self-driving cars) [4]. Thus, there is no obvious way for a software architect to determine the best deployment configuration of iCPS, even if all the requirements and constraints are known.

Our long-term goal is to establish a method that allows the software architect of iCPS to choose an appropriate deployment architecture, based on requirements on quality, cost and security. As the first step towards the goal, this paper especially focuses on assessing the costs and security of iCPS posed by the deployment configuration. We first present an abstract model of iCPS to capture the possible deployment configurations. We then propose a rule-based method that assesses cost-related attributes for the configurations. Finally, we investigate security threats within different configurations, to help the architect to choose appropriate configuration based on given security requirements.

II. Deployment Architecture of ML-Based CPS

A. Intelligent Cyber-Physical System (iCPS)

Cyber-physical systems (CPS) are systems that are built from the seamless integration of computation and physical components. Emerging cloud computing and IoT technologies allow a wide range of system configuration to meet individual requirements and use cases [3]. Figure 1 shows a typical architecture of CPS, in which the cloud computing is exploited. In the figure, devices in a physical world collect data, and send the data to an *edge* component (i.e., local server). The edge component can communicate with *cloud* servers via the Internet, in order to use any computational resource *as a service*. Using the data and the resources on the cloud, the edge commands actions for some devices [5].

Integrating machine-learning (ML) technologies with CPS may implement more intelligent and challenging applications, where decisions and behaviors cannot be specified by explicit rules. We call such applications *intelligent CPS* (iCPS).

Figure 2 depicts a data flow diagram (DFD), showing how an iCPS application typically works. The first step to implement iCPS is to build a *Predictive Model (PM)* from *Training Data-Sets (DTS)*. This *Model Learning (ML)* process

978-1-7281-4136-7/19/$31.00 ©2019 IEEE
DOI 10.1109/ASEW.2019.00036

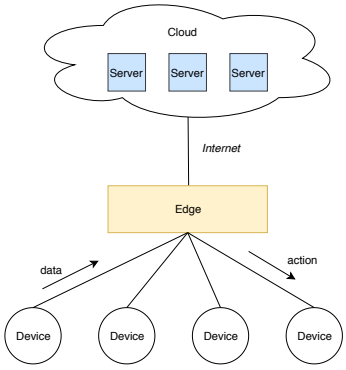

Fig. 1. Architecure of CPS with cloud and edge systems

is supposed to be performed once (or periodically with a low frequency) based on manual experiments.

Once a predictive model is built, it is deployed on the application to be used for intelligent decision making, as shown in the *Decision Taking (DT)* process. The process collects the data from a device, and requests the predictive model for a prediction based on the data. Based on the prediction returned, the application takes a decision to command an action to a device. This process is executed whenever there is a request, with a frequency from few millisecond to several hours/days, depending on the application.

B. Considering Deployment Configurations of iCPS

In order to simplify the problem, let us suppose that any iCPS can be abstracted by the four components (DTS, ML, PM, and DT) and physical devices, as shown in Figure 2. Then, each of the four components can theoretically be deployed either on the *edge* or on the *cloud*, considering the architecture of CPS (see Figure 1).

Since there exist two different deployments for each of the four, there are thus $16 (= 2^4)$ possible deployment configurations, as illustrated in Figure 3. In each configuration, two boxes are depicted where the upper box represents a cloud, and the lower box represents an edge. For example, configuration (4) represents that TDS and ML are deployed on the cloud, whereas PM and DT are deployed on the edge. An arrow between components represents a data flow, as defined in the DFD in Figure 2.

C. The Problem: How to Choose the Best Configuration?

A major challenge of designing iCPS lies in the fact that it is not obvious for software architects which deployment configuration best suits a given set of requirements and constraints. The best configuration should be determined based on various attributes of the target application. However, in this paper, we narrow our focus on the *cost* and the *security* only.

1) Cost: The cost of machines, computation, networking, operation, and maintenance in both edge (local) and cloud (remote) infrastructures is an important factor to take into account. Cloud computing can provide on-demand and elastic

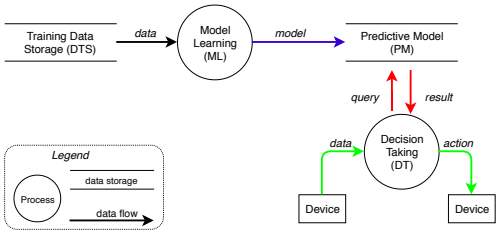

Fig. 2. Abstract data flow diagram of iCPS applications

computation resources. Hence, using the cloud for machine learning or deploying applications/services on the cloud produces the cost merit. The maintenance is carried out by the platform provider. In the cloud-based machine-learning platform, various kinds of algorithms and processes are ready to use, and they are facilitated by application program interfaces (API). To reproduce such an environment locally on an edge server, the time and expertise are required as well as huge memory and high performance computers.

2) Security: The security of iCPS takes different dimensions among the deployment configurations. It depends on whether or not the configuration allows appropriate protection against common threats and attacks to the processes, as well as the confidentiality and authenticity of the data.

When an iCPS application is connected to the Internet, the security becomes a crucial factor. The tight interaction between the software and the physical components in CPS enables cyber-attacks to have catastrophic physical consequences [6]. For instance, over a half million pacemakers have been recalled by the American Food and Drug Administration, due to fears that hackers could exploit cyber-security flaws to modify the patient's heartbeat [7]. In case of ML algorithms, the *poisoning attack* for the training data can modify the prediction model [8]. Theoretically, if the data doesn't traverse the network, the security is supposed to be increased. The less data is in the network, the less the data is in a breach or leak.

Just considering these factors, it is quite confusing to choose an appropriate configuration. In the next section, we try to propose a method that can *assess* the cost and the security, for the sixteen deployment configurations.

III. Towards Configuration Evaluation

Our long-term goal is to establish a method that allows software architects of iCPS to quantitatively *evaluate* the deployment configuration for a given set of requirements and constraints. Towards the goal, this paper presents brief assessment methods with respect to the cost and security.

A. Assessing the Cost

Every iCPS application requires to create a predictive model by machine learning, before the application is in operation. The resources and the time needed for the *model learning phase* are quite different from those in the *operation phase*. In addition, the predictive model should be updated during

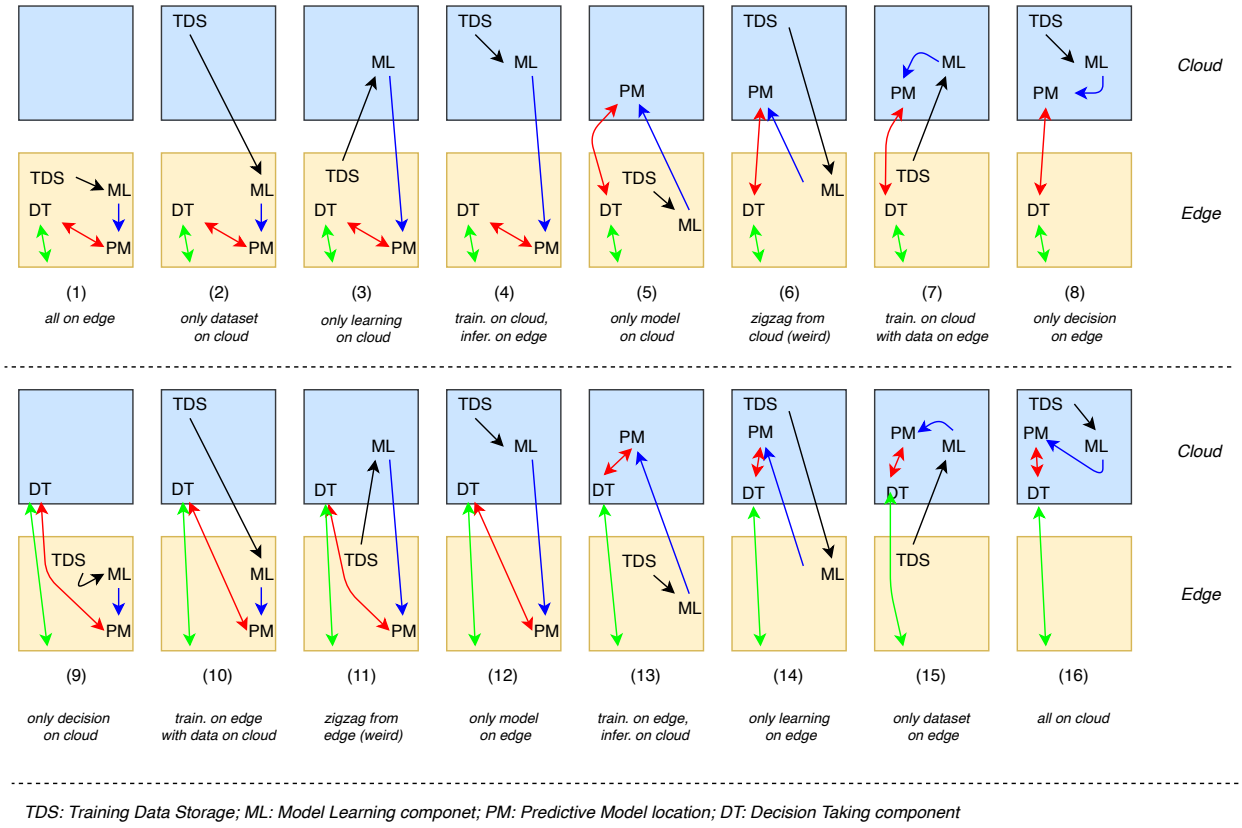

Fig. 3. The 16 possible deployment configurations for iCPS

the *maintenance phase*. For these reasons, we decided to distinguish the cost of the three phases.

The cost at the edge side and the cost at the cloud side are considered separately. The cost at one side is highly dependent on the components that are deployed. In the learning phase, the locations of TDS, ML, and PM are considered. In the operation phase, the locations of PM and DT are considered. In the maintenance phase, all the four components are considered. To compare different configurations by cost, we try to *rate* every configuration, depending on where each component is present. As an example, we chose to rate 1.5 when TDS is present on a side; 2.5 for ML; 0.5 for PM; 0.5 for DT. These values were chosen because; a huge capacity of memory is required for storing the training data; a large memory, high performance computing, and many libraries are required for the machine-learning task; the predictive model is usually smaller than the training data set; the size of decision-taking component depends on the application but usually requires considerably smaller resources than the machine learning task.

Note that TDS should be on the same side as ML to execute the learning phase. Therefore, if TDS exists at another side of ML, it has to be moved, and the resources for TDS should be counted for both side. Similarly, PM is generated at the same side as ML. Therefore, if the configuration deploys PM at another side, then PM has to be moved and the resources should be counted for both sides.

Table I presents the values obtained with the proposed method. For example, Configuration 4 requires a small cost on the edge during the learning phase. In the operation phase, however, the cost is maximized on the edge, and is zero on the cloud. In Configuration 8, the operation costs are balanced on both sides. The actual cost will be derived by multiplying each value by the unit price of the platform, and the time spent for each phase, which depend on individual iCPS applications.

B. Assessing the Security

The security threats to iCPS are mainly related to two activities: (1) preventing the system from working correctly, or (2) stealing the primary data. In the first case, an attacker inserts incorrect data or modifies correct data in DTS to produce a wrong PM. They are called poisoning attack or evasion attack [9]. Also, communications between PM and DT can be altered or subject to denial-of-service (DoS) attacks. Moreover, DT can be a source of attacks (e.g., service manipulation). Data can be stolen at any levels. For instance, the literature [8], indicates that it is possible to infer a part of original model or dataset on the basis of input/output pairs.

Of course, the deployment configuration influences the security issues [10], [11]. If PM and DT are deployed on different side, a DoS attack to PM increases the reponse time

TABLE I
ASSESSING THE COST IN DIFFERENT CONFIGURATIONS

Configuration	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Learning cost (edge)	4.5	4.5	2	0.5	4.5	4.5	1.5	0	4.5	4.5	2	0.5	4.5	4.5	1.5	0
Learning cost (cloud)	0	1.5	4.5	4.5	0.5	2	4.5	4.5	0	1.5	4.5	4.5	0.5	2	4.5	4.5
Operation cost (edge)	1	1	1	1	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0	0	0	0
Operation cost (cloud)	0	0	0	0	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	1	1	1	1
Maintenance cost (edge)	5	5	2.5	1	5	5	2	0.5	4.5	4.5	2	0.5	4.5	4.5	1.5	0
Maintenance cost (cloud)	0	1.5	4.5	4.5	0.5	2	4.5	4.5	0.5	2	5	5	1	2.5	5	5

of the prediction, and thus declines the performance. If DT is on the cloud, an attacker may attempt to re-size the cloud resources, to increase the operation cost [11].

Also, the edge side is not free of threats [12]. Privacy leakage, service manipulation, injection of information, and even physical damage can be achieved on the edge side, facilitated by the usage of vulnerable equipment or inappropriate system administration. We are currently in the process to derive rules to assess the security threats for the deployment configuration.

C. Limitations

We assumed the simplified architecture and model for iCPS (see Figures 1 and 2). We did not count the *fog computing*, nor the case where DT is spread on the cloud and edge.

IV. CONCLUSION AND FUTURE PERSPECTIVES

In the context of intelligent CPS with machine-learning technologies (iCPS), data and software components can be deployed either on the cloud or on the edge. Although such deployment configuration influences the quality, cost, and security of the target application, it is not obvious for software architects how to determine the best configuration for a given set of requirements and constraints. In this paper, we investigated a method that briefly assesses the deployment configuration of iCPS with respect to the cost and the security. Introducing the architecture and data flow models, we first enumerated 16 possible configurations. We then defined methods to assess the configurations for the cost and the security.

Our future perspective is to extend the method for other quality attributes, including reliability, compatibility, maintainability, and portability, as defined in the SQuaRE [13]. A second perspective is to introduce quantitative functions associated to the attribute assessment. For instance, the network usage is correlated to the size of DTS in the learning phase, as well as the volume of communication between DT and PM in the operation phase. Complex models have been proposed in the literature to evaluate the cost at both side [14]. The idea is to re-investigate those models to derive a more abstract and universal view. Our final goal is to recommend the best configuration. For this, we need to develop a method that collects requirements and constraints from the user (e.g. the size of the training data, the location of the data, the edge capabilities). In the workshop, we hope to discuss these issues with interested researchers and practitioners.

Acknowledgment: This research was supported by JSPS KAKENHI, Grant Numbers JP19H01138, JP17H00731, JP18H03242, JP18H03342, JP19H04154. It was also supported by Région Rhône-Alpes in France: Kouno-Tori project of the SCUSI'18 program (18006667 01-41024).

REFERENCES

[1] V. Varadharajan, V. Subramaniyaswamy, J. H. Abawajy, and L. Yang, "Intelligent, smart and scalable cyber-physical systems," *Journal of Intelligent and Fuzzy Systems*, vol. 36, no. 5, pp. 3935–3943, 2019. [Online]. Available: https://doi.org/10.3233/JIFS-179108

[2] O. Rajabi Shishvan, D. Zois, and T. Soyata, "Machine intelligence in healthcare and medical cyber physical systems: A survey," *IEEE Access*, vol. 6, pp. 46 419–46 494, 2018.

[3] A. Darwish and A. E. Hassanien, "Cyber physical systems design, methodology, and integration: the current status and future outlook," *Journal of Ambient Intelligence and Humanized Computing*, vol. 9, no. 5, pp. 1541–1556, Oct 2018. [Online]. Available: https://doi.org/10.1007/s12652-017-0575-4

[4] L. Stojanovic, "Intelligent edge processing," in *Machine Learning for Cyber Physical Systems*, J. Beyerer, A. Maier, and O. Niggemann, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2020, pp. 35–42.

[5] W. Yu, F. Liang, X. He, W. G. Hatcher, C. Lu, J. Lin, and X. Yang, "A survey on the edge computing for the internet of things," *IEEE Access*, vol. 6, pp. 6900–6919, 2018.

[6] D. Ratasich, F. Khalid, F. Geissler, R. Grosu, M. Shafique, and E. Bartocci, "A roadmap toward the resilient internet of things for cyber-physical systems," *IEEE Access*, vol. 7, pp. 13 260–13 283, 2019.

[7] A. Hern, "Hacking risk leads to recall of 500000 pacemakers due to patient death fears," London, U.K., Aug. 2017, https://www.theguardian.com/technology/2017/aug/31/hacking-risk-recall-pacemakers-patient-death-fears-fda-firmware-update [Retrieved July, 2019].

[8] N. Baracaldo, B. Chen, H. Ludwig, J. A. Safavi, and R. Zhang, "Detecting poisoning attacks on machine learning in iot environments," in *2018 IEEE International Congress on Internet of Things, ICIOT 2018, San Francisco, CA, USA, July 2-7, 2018*. IEEE Computer Society, 2018, pp. 57–64. [Online]. Available: https://doi.org/10.1109/ICIOT.2018.00015

[9] H. Bae, J. Jang, D. Jung, H. Jang, H. Ha, and S. Yoon, "Security and privacy issues in deep learning," *CoRR*, vol. abs/1807.11655, 2018. [Online]. Available: http://arxiv.org/abs/1807.11655

[10] A. Bonguet and M. Bellaïche, "A survey of denial-of-service and distributed denial of service attacks and defenses in cloud computing," *Future Internet*, vol. 9, no. 3, p. 43, 2017. [Online]. Available: https://doi.org/10.3390/fi9030043

[11] S. Dong, K. Abbas, and R. Jain, "A survey on distributed denial of service (ddos) attacks in sdn and cloud computing environments," *IEEE Access*, vol. 7, pp. 80 813–80 828, 2019.

[12] R. Roman, J. López, and M. Mambo, "Mobile edge computing, fog et al.: A survey and analysis of security threats and challenges," *Future Generation Comp. Syst.*, vol. 78, pp. 680–698, 2018. [Online]. Available: https://doi.org/10.1016/j.future.2016.11.009

[13] ISO/IEC, "Iso/iec 25010 system and software quality models," Tech. Rep., 2010.

[14] F. A. Zaman, A. Jarray, and A. Karmouch, "Software defined network-based edge cloud resource allocation framework," *IEEE Access*, vol. 7, pp. 10 672–10 690, 2019.