

Cloud Based Decision Making for Multi-Agent Production Systems

Hamood Ur Rehman^{1,2}, Terrin Pulikottil^{3,4}, Luis Alberto Estrada-Jimenez^{3,4},
Fan Mo¹, Jack C. Chaplin¹, Jose Barata^{3,4}, and Svetan Ratchev¹

¹ University of Nottingham, Nottingham, UK

{Hamood.Rehman, Fan.Mo, Jack.Chaplin, Svetan.Ratchev}@nottingham.ac.uk

² TQC Ltd., Nottingham, UK

³ Centre of Technology and Systems, UNINOVA Instituto Desenvolvimento de Novas
Tecnologias, Caparica, Portugal

{tpulikottil,lestrada,jab}@uninova.pt

⁴ NOVA University of Lisbon, Caparica, Portugal

Abstract. The use of multi-agent systems (MAS) as a distributed control method for shop-floor manufacturing control applications has been extensively researched. MAS provides new implementation solutions for smart manufacturing requirements such as the high dynamism and flexibility required in modern manufacturing applications. MAS in smart manufacturing is becoming increasingly important to achieve increased automation of machines and other components. Emerging technologies like artificial intelligence, cloud-based infrastructures, and cloud computing can also provide systems with intelligent, autonomous, and more scalable solutions. In the current work, a decision-making framework is proposed based on the combination of MAS cloud computing, agent technology, and machine learning. The framework is demonstrated in a quality control use case with vision inspection and agent-based control. The experiment utilizes a cloud-based machine learning pipeline for part classification and agent technology for routing. The results show the applicability of the framework in real-world scenarios bridging cloud service-oriented architecture with agent technology for production systems.

Keywords: Cloud Computing · Multi-Agent · Machine Learning · Production Systems.

1 Introduction

The increasing demand for small batch sizes and customized products, combined with a high level of market fluctuations, is requiring manufacturing industries to change their traditional production methods. In some areas, fixed lines with centralized control are being replaced by autonomous modules with distributed and decentralized control, with the goal of increasing their level of agility and flexibility, but uptake is slow.

To achieve this transition, several manufacturing paradigms have been suggested and have successfully showcased applications that enable adaptable and re-configurable manufacturing solutions. Examples of these emerging paradigms are the Evolvable Production Systems (EPS) [1], Bionic Manufacturing Systems [2], Holonic Manufacturing Systems (HMS) [3], and Reconfigurable Manufacturing Systems (RMS) [4]. Each takes a different approach to move traditional mass production towards an era of embedded system intelligence capable of mass customization and high product personalization.

Emerging technologies such as artificial intelligence, multi-agent technologies, service-based infrastructures, and cloud computing [5] are supporting the implementation of these new paradigms and enabling the necessary infrastructure to develop new levels of interoperability, integration, and seamless data exchange. These are critical requirements for the transition to the fourth industrial revolution where high levels of digitization of resources are expected.

However, these technological enablers are recent developments, and one of the common challenges is the lack of methodologies that showcase their implementation and integration in real scenarios, leaving such works in very theoretical and abstract terms.

This article proposes a multi-agent framework capable of reconfiguring and monitoring manufacturing operations in response to data from a cloud-based analysis pipeline. The framework was defined to be generic and useful for various environments, and here is instantiated in a specific use case based on product testing. A multi-agent infrastructure runs distributed on the shop floor and grants intelligence to products, which can hence communicate their required operations with transport elements and machines. The introduction of a monitoring entity allows the system to be constantly checking for faults. A cloud-based machine learning platform provides intelligence “as a service” that allows the product agents to store, train and predict machine learning models that support decision making and feedback.

This paper is organized as follows; Section two introduces the background of the related technology for this paper. Section three details the decision-making framework based on a cloud platform. Section four is about experimentation and deployment. Section five explains final conclusions and future works.

2 Background

Machine learning (ML) allows the software to learn over time from data and make decisions and predictions that improve over time. ML is being used to improve decision-making, improve operations and customer experiences in a wide range of sectors.

Many real-world problems have high complexity and unknown underlying models which makes them excellent candidates for the application of ML. ML can be applied to various areas of computing to design and programming explicit algorithms with high-performance output, such as in the manufacturing industry,

robotics [6], e-commerce, medical applications [7], scientific visualization [8] and fault diagnosis [9].

Cloud technologies applied to manufacturing enable the conversion of manufacturing resources and capabilities into entities capable of being virtualized, combined, and enhanced [10] established the cloud concept in detail and presented a system that was service-oriented and interoperable. The system revolved around the customer/cloud user and enterprise user.

Cloud technologies often enable on-demand use of resources that follow a deployed-as-you-need model so that resources may be used with common interfaces. The cloud services can be mainly divided into Platform-as-a-service(PaaS), Infrastructure as a service(IaaS), Software as a service(SaaS). PaaS cloud-based solutions offer connectivity of a user's applications to the application resources, web services, or storage infrastructure of a cloud. IaaS, the major component in a PaaS solution, is a platform where services provided by other platform providers. While Software as a service(SaaS) is an application software delivery model. In the SaaS delivery model, the application software is delivered via the Internet to end-users over an Internet Protocol (IP) connection.

Distributed artificial intelligence (DAI) has attracted research interest because it can solve complex computing problems by breaking them into simpler tasks. DAI algorithms can be divided into three categories: parallel AI, distributed problem solving (DPS), and multi-agent systems (MAS)[11]. Parallel AI involves the development of parallel algorithms, languages, and architectures to improve the efficiency of classic AI algorithms by taking advantage of task parallelism. DPS involves dividing a task into several subtasks, and each subtask is assigned to one of a group of cooperating nodes (called computing entities). Computing entities have shared knowledge or resources and predefined communications with other entities, which limits their flexibility [12].

MAS divides the components of the system into autonomous and 'selfish' software agents, each aiming to achieve its own goals by collaborating with other agents. MAS supports complex applications where many components with conflicting objectives need to interact by breaking them into independent simpler entities. They require distributed and parallel data [13].

Agent technology is recognised as a powerful tool for the 21st century manufacturing system. Researches are on-going for utilising agent technology in manufacturing enterprises, production process planning and scheduling, workshop control, and re-configurable manufacturing systems [14].

The applications of agents in manufacturing industry have seen wider acceptance, such as: process and manufacturing control[15–18], manufacturing simulation and execution[19–21], reconfiguration and self-adaptation in manufacturing [22, 3, 23]. Monitoring, quality control, and diagnostics in manufacturing[24–27] can be greatly benefited by agent integration coupled with other technologies.

Machine vision and image processing techniques utilised in manufacturing applications are used for integrated inspections to detect defects and improve product quality in the process [28]. In many cases, traditional machine learning has made great progress and produced reliable results [29], but different

preprocessing methods are required, including structure-based, statistical-based, filter-based, and model-based techniques. To enhance performance for quality control these techniques can be combined with expert knowledge to extract representative features [30, 31] that influence quality. Most of the previous research about manufacturing quality monitoring with agent application does not relate with cloud computing technologies and machine learning. This work addresses that area.

3 Decision Making Framework

The framework for cloud-based decision-making in manufacturing is divided into two components, the multi-agent system, and the cloud computing services. A Graphical representation of the framework is given in Figure 1.

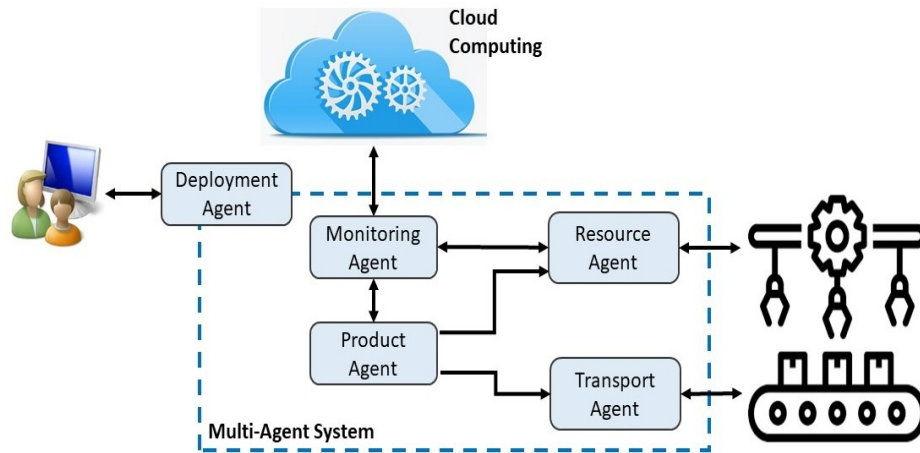


Fig. 1: Framework for MAS and cloud-intelligence integration

3.1 Multi-Agent System (MAS) Component

The framework requires a set of agents with defined responsibilities that are each instantiated when required. The agents developed in PROSA [32] and PRIME [25] projects served as an inspiration for the development of this component. Some agents may each have a physical asset associated with it and provides an interface to the virtualized "skills" performed by the physical asset. The list of agents and their functionalities are given in Table 1.

Table 1: Agent types and their functionalities

Agent Type	Functionalities
Deployment Agents (DAs)	Launches and kills the agents according to the task being performed. Keeps track of the list of agents being deployed.
Product Agents (PAs)	Represents each product on which tasks are being performed (one agent per product). Requests skills from other agents to be performed on the product. Updates the product properties (e.g. faulty or not) as task progresses. The agent is removed when the product is finished.
Transport Agents (TAs)	Performs the skills of moving the product from one place to another. Performs skills monitoring the development of bottlenecks in the system. Examples include conveyors, pick & place robots, AGVs.
Monitoring Agents (MAs)	Collects data from other agents and provides information. Utilizes the cloud computing platform for decision making. Skills being performed includes quality prediction, prognostics.
Resource Agents (RAs)	Represents a resource on the shop floor. Provides the skills provided by the resource. Extracts data from the resource it is abstracting.

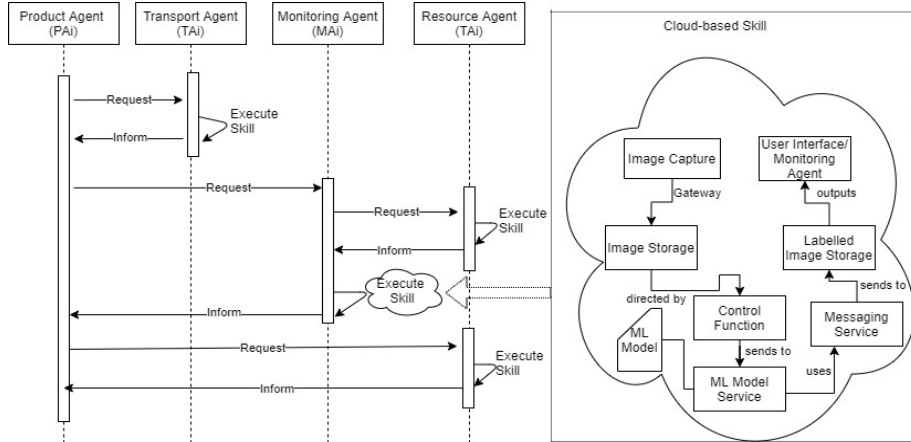


Fig. 2: Sequence diagram for the framework

The sequence diagram (Figure 2) shows the interaction between products, transportation assets, monitoring elements, and resources. It assumes that the agents have been already launched. This means that their skills and services have been already identified by the Deployment Agents (therefore, DAs are not included in the sequence diagram).

Product Agents (PA) guides the sequence of the process creating a product by requesting the skills from other agents to be performed on the physical product instance. This request of skills could vary depending on the product

properties and the skills required to create the product.

Transport Agents (TA) execute the skills requested by the PA to move the product to the required resources and to inform the current position of the product to PA. The TA also considers the buffers and potential bottlenecks that might arise before executing its skills.

Resource Agents (RA) represent shop-floor resources (such as machining centers, robots, sensors, cameras). RAs can provide their availability, task status info, and resource information as per request.

Monitoring Agents (MA) doesn't have a physical entity associated with it but utilizes the skills provided by RAs if required (e.g. image capturing skills from a camera RA). The MA offers the cloud computing functionality as skills if required for quality prediction and prognostics. For example, PA requests MA to perform a skill of quality prediction of the product associated with it. MA if required, requests RA to perform its skill (e.g. take a photo of the product). MA then uses a cloud platform for decision-making and informs PA about the quality of the product.

3.2 Cloud Computing Component

Cloud computing is used to enhance the capability of the multi-agent framework by bridging it with service-oriented architecture. The MA looks for certain events that act as a trigger for it to execute its functionality (requesting RAs and informing PA). Each of the services housed in the cloud can be instantiated by means of event trigger functions. The agent and the cloud platform rely on a gateway to realize their functionality. The event trigger functions are used by the MA to activate the capture of images and send them to cloud storage. This population of images in cloud storage triggers additional functionality and decision-making by the ML pipeline. The insight generated by the ML pipeline, based on the captured images in the cloud platform, is sent to the MA which then uses it to execute an operation. MA either triggers TAs or RAs which are responsible for transportation and production skills respectively. The agent interaction with cloud computing platform along with details on cloud service deployment is elaborated in more detail in the experimentation and deployment section.

The algorithm for ML processing used for image detection and classification is Neural Architecture Search (NAS) where a dataset and task (image detection and classification) is provided. This is used to find the design of machine learning model, that performs best among all other models for a given task as the model is trained under the provided dataset. NAS uses search strategy to find best model from all possible models that maximises performance (figure 3).

The three constituent of NAS include search space, search strategy and performance estimation. Search space defines the neural architecture selection basis like chain or multi-branch network, micro/macro-search or cell-search [33]. Search strategy and performance estimation employ multiple methods selected on the search space selected previously [33] such as random search, reinforcement



Fig. 3: NAS algorithm for determination of image detection and classification ML Model

learning, and evolutionary algorithms. The model derived from this approach can be used directly for the purpose. Google Cloud Platform (GCP) based its AutoML service on a novel architecture NASNET that uses NAS for image classification. NASNET redesigns the search space so best layer could be found and stacked multiple times in flexible manner for final network. This network was used to perform search strategy on image datasets and best learned architecture was selected for image detection and classification. More detail on the work can be found in the work done by Google Research Team [34].

4 Experimentation and Deployment

The experiment carried out in this work includes the implementation of cloud-based decision-making and the multi-agent-based simulation of the proposed shop floor. The demonstrator used as a basis for this experiment is shown in (Figure 4) and includes conveyors, three drilling stations, and one camera module. At this point, the Factory I/O environment was used specifically to demonstrate a use case of a plant layout and to provide an objective vision of the implementation of the proposed framework.

Once a product's order has been launched into the system, the component moves forward through the first conveyor until the camera module is reached. Immediately, it takes a picture of the component and compares it with an ML cloud-based classifier. The part is labelled and routed as per decision. A conveyor is used to direct the component to the rework/reject station in case that the part is defective. In other cases, it looks for the other production resource stations where the part could be routed. This routing is decided on the condition of station busyness. The part is routed to a less busy station and the station executes its function or skill and the process is finalized.

The experimental setup includes services employed in a cloud environment, agent resources, and deployed physical resources. The camera module connected to Raspberry Pi acting as a gateway device to the Cloud-Based Machine Learning Pipeline. The functionality employed by cloud-based services is of visual quality inspection for defect-free production and process routing.

4.1 Cloud-based decision making

Cloud-based machine learning (ML) model is trained on images and deployed for analysis on an end-point (Figure 6). As more and more classifications take

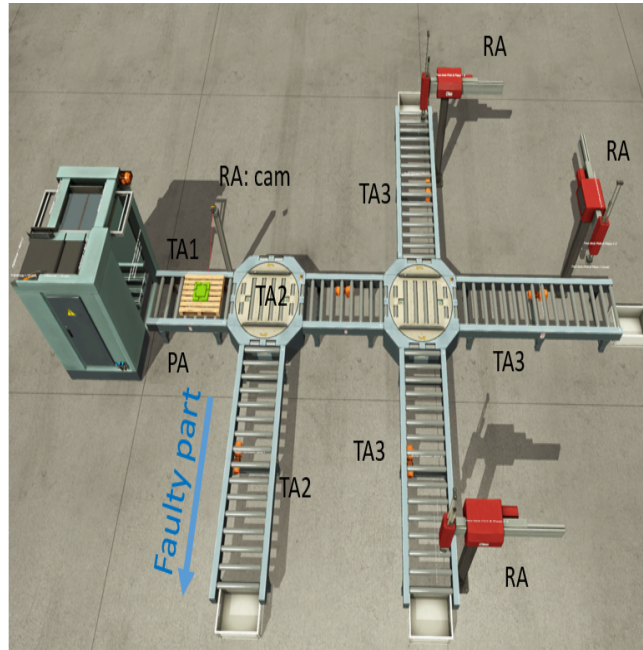


Fig. 4: Shop-floor demonstrator environment in Factory IO

place the model is improved. The classified images are sent to the pipeline for training with every iteration. The cloud-based deployment of quality inspection for mechanical components are used as the basis for agent-based decision-making. The images for classification are taken from a public dataset [35], and are trained to a high level of confidence.

Images ingested from the gateway device obtained by the camera module are stored in the storage housed by the cloud platform. This event of image storage acts as an event trigger that executes a script sending the stored image at the machine learning service endpoint. The endpoint houses the model that is determined by using NASNET, trained for the task and on the dataset in GCP. At this endpoint, the image is labelled as per the classification obtained by the trained, tested, and validated model. The label assigned to the image falls within the category of 'ok', 'defect', and 'uncertain'.

The labelled image is written to the message topic (MQTT publish/subscribe service). This message topic triggers another event that moves the labelled image to the predicted cloud storage offering separate storage services for the categories. The labelled images are moved into each respective service as per category. The uncertain image requires human intervention. An Application Protocol Interface (API) is incorporated that takes the uncertain image and inquires the accurate (ok) or defective status from the human operator. The image is then classified as per human input.

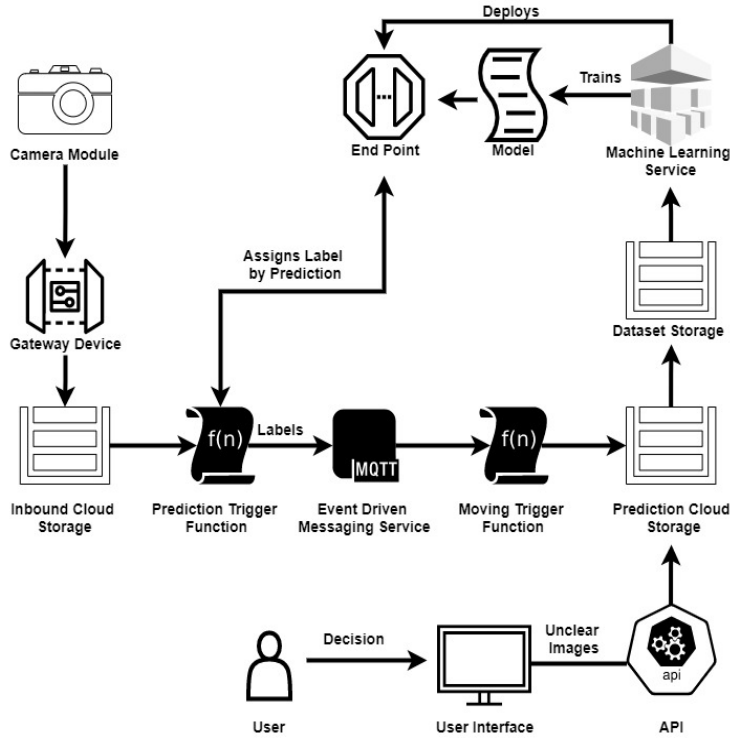


Fig. 5: Cloud Based Machine Learning Pipeline for Part Detection

4.2 Multi-agent based simulation

The agent-based framework presented in previous section is implemented and supported by the cloud-based platform.

Agent programming is implemented with the JADE (Java Agent Development Environment) platform. The agents deployed in JADE use cloud based classification input generated by vision model trained on the dataset, as input leading to further actions as per the proposed framework discussed above. All of the actors in the production environment are controlled by means of agents.

In the current use case, the agent execution and simulation is performed after the deployment of six agents: PA, MA, TA1, TA2, TA3 and RA. TA3 and RA represent the set of stations and their required transportation respectively. To account for similarity in resource skill and identical negotiation steps, TA3 and RA were not deployed individually rather resources 1, 2 and 3 are represented by agent RA and conveyors by the agent TA3. The simulation is performed within two variants, based on cloud based decision process, set by the MA if the part is defective (variant 1) or accurate (ok) (variant 2). The simulation process can be seen in Figure 7. Figure 7(a) presents the sequential model generated when a part is defective. In this case the TA2 is activated routing the product to a

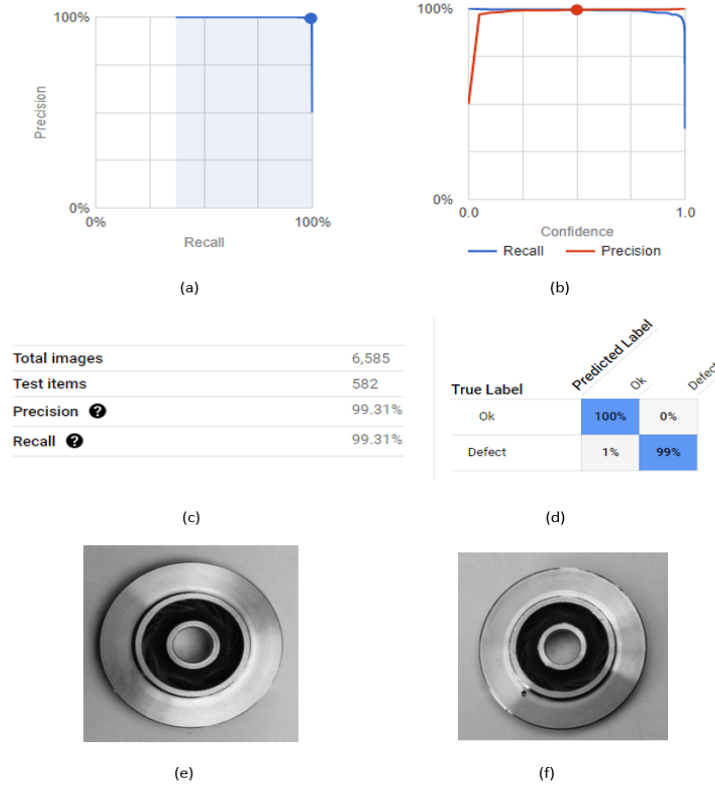


Fig. 6: ML Pipeline Deployment. (a) Precision vs Recall against identified part labels. Precision is measurement of positive label assignment (ratio between the True Positives and all the Positives) and recall is the measure of model correctly identifying True Positives. (b) Maximum confidence at Threshold = 0.5. (c) Dataset size (total and test images). (d) Confusion Matrix representing True and Predicted Labels. (e) Accepted (OK) part. (f) Defective Part.

storage place for defective parts. Finally, in Figure 7(b) the sequential model generated starts a negotiation process with RA followed by TA3 with the aim of performing the respective job.

5 Conclusion and Future Work

The research presents an elaborate framework on the application of cloud and agent technologies on quality control by vision inspection and agent-based control in a production system. The approach developed is suitable to achieve a quality testing-driven production that compliments the 'no-faults-forwards' approach in manufacturing. It reduces the risk of accepting defective parts and

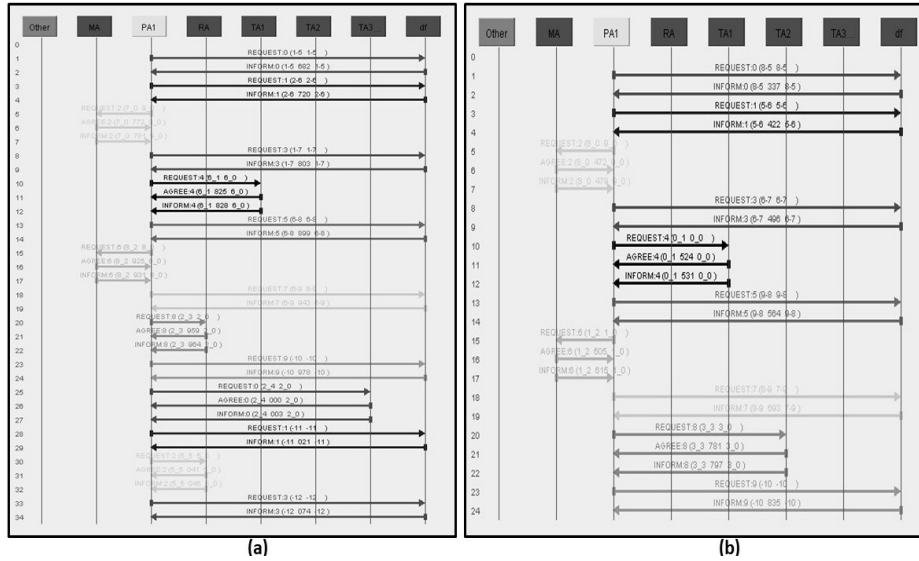


Fig. 7: Sequential diagram for (a) accurate parts (b) defective parts

rejecting good parts (Type 1 and 2 errors). This approach benefits in reducing costs as well as maintaining quality standards. Future work for this approach involves expanding the Machine Learning models on the cloud, developing inferences from structured data along vision models. This will enable integration with data generated on n shop floor for better control in production applications. The current-use case considers diverting of parts to less busy stations; however, a methodology to define an optimal routing resource has to be developed in future works. The multi-agent system capability will be enhanced to enable routing to stations having different processing capabilities. Deploying the test-driven production approach to multiple physical use-cases will also be a part of future activities.

Currently, the approach is constrained by the size of the data set i.e. a large number of pictures are required by the system to train the model for accurate prediction, which is a limitation of the service. A limitation however is a gap in proper integration of cloud services with agent technology for effective coordination and control. Future works will be looking into different ways the limitation can be overcome while keeping data set size to minimum and developing mechanism for effective service deployment and integration. Other mechanisms for deploying the cloud-based testing control will be implemented and compared with the multi-agent approach. Another limitation observed in the research is related to interoperability, common ontology, semantics, and protocols across the whole production line. Multiple APIs need to be deployed for communication between cloud pipeline, gateway device, agent system, and production system devices. A solution to this problem will be discussed in future works. Finally, fu-

ture developments will include the creation of the necessary interfaces to link the simulated environment in Factory I/O with JADE and with the cloud based ML infrastructure. This will clearly showcase the advantages of the framework and will be a step ahead towards its implementation in an industrial environment.

6 Acknowledgement

This work is carried out under DiManD Innovative Training Network (ITN) project funded by the European Union through the Marie Skłodowska-Curie Innovative Training Networks (H2020-MSCA-ITN-2018) under grant agreement number no. 814078.

References

1. Onori, M., Barata, J.: Evolvable production systems: New domains within mechatronic production equipment. In: 2010 IEEE International Symposium on Industrial Electronics. pp. 2653–2657. IEEE (2010)
2. Tharumarajah, A.: Comparison of the bionic, fractal and holonic manufacturing system concepts. *International Journal of Computer Integrated Manufacturing* 9(3), 217–226 (1996)
3. Van Brussel, H., Wyns, J., Valckenaers, P., Bongaerts, L., Peeters, P.: Reference architecture for holonic manufacturing systems: Prosa. *Computers in industry* 37(3), 255–274 (1998)
4. Bi, Z.M., Lang, S.Y., Shen, W., Wang, L.: Reconfigurable manufacturing systems: the state of the art. *International journal of production research* 46(4), 967–992 (2008)
5. Leitão, P., Colombo, A.W., Karnouskos, S.: Industrial automation based on cyber-physical systems technologies: Prototype implementations and challenges. *Computers in industry* 81, 11–25 (2016)
6. Wang, L., Du, Z., Dong, W., Shen, Y., Zhao, G.: Hierarchical human machine interaction learning for a lower extremity augmentation device. *International Journal of Social Robotics* 11(1), 123–139 (2019)
7. Wu, D., Zhang, Y., Ourak, M., Niu, K., Dankelman, J., Vander Poorten, E.B.: Hysteresis modeling of robotic catheters based on long short-term memory network for improved environment reconstruction. *IEEE Robotics and Automation Letters* pp. 1–1 (2021)
8. Torayev, A., Schultz, T.: Interactive classification of multi-shell diffusion mri with features from a dual-branch cnn autoencoder. In: EG Workshop on Visual Computing for Biology and Medicine (2020)
9. Tang, T., Hu, T., Chen, M., Lin, R., Chen, G.: A deep convolutional neural network approach with information fusion for bearing fault diagnosis under different working conditions. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science* p. 0954406220902181 (2020)
10. Vincent Wang, X., Xu, X.W.: An interoperable solution for Cloud manufacturing. *Robotics and Computer-Integrated Manufacturing* 29(4), 232–247 (2013)
11. Wooldridge, M.: An introduction to multiagent systems. John wiley & sons (2009)
12. Bond, A.H., Gasser, L.: Readings in distributed artificial intelligence. Morgan Kaufmann (2014)

13. Botti, V., Omicini, A., Mariani, S., Julian, V.: Multi-agent systems. MDPI-Multidisciplinary Digital Publishing Institute (2019)
14. Adeyeri, M.K., Mpofu, K., Olukorede, T.A.: Integration of agent technology into manufacturing enterprise: A review and platform for industry 4.0. In: 2015 International Conference on Industrial Engineering and Operations Management (IEOM). pp. 1–10. IEEE (2015)
15. Li, Z., Jiang, X., Yao, S., Li, D.: Research on collaborative control method of manufacturing process based on distributed multi-agent cooperation. In: 2018 11th International Symposium on Computational Intelligence and Design (ISCID). vol. 2, pp. 41–46. IEEE (2018)
16. Li, D., Jiang, X., Wei, X.: Research on manufacturing process control based on multi-agent-system. In: 2018 IEEE 4th Information Technology and Mechatronics Engineering Conference (ITOEC). pp. 1306–1309. IEEE (2018)
17. Răileanu, S., Borangiu, T., Morariu, O.: Multi-agent solution for automated part supply in robotized holonic manufacturing. In: International Conference on Robotics in Alpe-Adria Danube Region. pp. 211–218. Springer (2016)
18. Vatankhah Barenji, A., Vatankhah Barenji, R.: Improving multi-agent manufacturing control system by indirect communication based on ant agents. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering* 231(6), 447–458 (2017)
19. Gwiazda, A., Sekala, A., Banaś, W.: Modeling of a production system using the multi-agent approach. In: IOP Conference Series: Materials Science and Engineering. vol. 227, p. 012052. IOP Publishing (2017)
20. Blesing, C., Luensch, D., Stenzel, J., Korth, B.: Concept of a multi-agent based decentralized production system for the automotive industry. In: International Conference on Practical Applications of Agents and Multi-Agent Systems. pp. 19–30. Springer (2017)
21. Büth, L., Broderius, N., Herrmann, C., Thiede, S.: Introducing agent-based simulation of manufacturing systems to industrial discrete-event simulation tools. In: 2017 IEEE 15th International Conference on Industrial Informatics (INDIN). pp. 1141–1146. IEEE (2017)
22. Leitao, P., Karnouskos, S., Ribeiro, L., Lee, J., Strasser, T., Colombo, A.W.: Smart agents in industrial cyber-physical systems. *Proceedings of the IEEE* 104(5), 1086–1101 (2016)
23. Barata, J., Camarinha-Matos, L.M.: Coalitions of manufacturing components for shop floor agility-the cobasa architecture. *International journal of networking and virtual organisations* 2(1), 50–77 (2003)
24. Peres, R.S., Rocha, A.D., Leitao, P., Barata, J.: Idarts—towards intelligent data analysis and real-time supervision for industry 4.0. *Computers in industry* 101, 138–146 (2018)
25. Rocha, A.D., Peres, R.S., Flores, L., Barata, J.: A multiagent based knowledge extraction framework to support plug and produce capabilities in manufacturing monitoring systems. In: 2015 10th International Symposium on Mechatronics and its Applications (ISMA). pp. 1–5. IEEE (2015)
26. Baer, S., Bakakeu, J., Meyes, R., Meisen, T.: Multi-agent reinforcement learning for job shop scheduling in flexible manufacturing systems. In: 2019 Second International Conference on Artificial Intelligence for Industries (AI4I). pp. 22–25. IEEE (2019)
27. Rokhforoz, P., Gjorgiev, B., Sansavini, G., Fink, O.: Multi-agent maintenance scheduling based on the coordination between central operator and decentralized producers in an electricity market. *arXiv preprint arXiv:2002.12217* (2020)

28. Xie, X.: A review of recent advances in surface defect detection using texture analysis techniques. *ELCVIA: electronic letters on computer vision and image analysis* pp. 1–22 (2008)
29. Neogi, N., Mohanta, D.K., Dutta, P.K.: Review of vision-based steel surface inspection systems. *EURASIP Journal on Image and Video Processing* 2014(1), 1–19 (2014)
30. Pernkopf, F., O’Leary, P.: Visual inspection of machined metallic high-precision surfaces. *EURASIP Journal on Advances in Signal Processing* 2002(7), 1–12 (2002)
31. Wang, J., Ma, Y., Zhang, L., Gao, R.X., Wu, D.: Deep learning for smart manufacturing: Methods and applications. *Journal of Manufacturing Systems* 48, 144–156 (2018)
32. Wyns, J.: Reference architecture for holonic manufacture: The key to support evolution and reconfiguration. Unpublished PhD thesis, Katholieke Universiteit Leuven, Leuven (1999)
33. Elsken, T., Metzen, J.H., Hutter, F., et al.: Neural architecture search: A survey. *J. Mach. Learn. Res.* 20(55), 1–21 (2019)
34. Zoph, B., Vasudevan, V., Shlens, J., Le, Q.: Automl for large scale image classification and object detection. *Google AI Blog* 2, 2017 (2017)
35. Dabhi, R.: casting product image data for quality inspection. In: <https://www.kaggle.com/ravirajsinh45/real-life-industrial-dataset-of-casting-product> (2020)