



Context-Based Resilience in Cyber-Physical Production System

Ada Bagozi¹ · Devis Bianchini¹ · Valeria De Antonellis¹

Received: 14 May 2021 / Revised: 3 August 2021 / Accepted: 6 September 2021 / Published online: 11 October 2021
© The Author(s) 2021

Abstract

Cyber-physical systems are hybrid networked cyber and engineered physical elements that record data (e.g. using sensors), analyse them using connected services, influence physical processes and interact with human actors using multi-channel interfaces. Examples of CPS interacting with humans in industrial production environments are the so-called cyber-physical production systems (CPPS), where operators supervise the industrial machines, according to the human-in-the-loop paradigm. In this scenario, research challenges for implementing CPPS resilience, promptly reacting to faults, concern: (i) the complex structure of CPPS, which cannot be addressed as a monolithic system, but as a dynamic ecosystem of single CPS interacting and influencing each other; (ii) the volume, velocity and variety of data (Big Data) on which resilience is based, which call for novel methods and techniques to ensure recovery procedures; (iii) the involvement of human factors in these systems. In this paper, we address the design of resilient cyber-physical production systems (R-CPPS) in digital factories by facing these challenges. Specifically, each component of the R-CPPS is modelled as a *smart machine*, that is, a cyber-physical system equipped with a set of recovery services, a Sensor Data API used to collect sensor data acquired from the physical side for monitoring the component behaviour, and an operator interface for displaying detected anomalous conditions and notifying necessary recovery actions to on-field operators. A *context-based mediator*, at shop floor level, is in charge of ensuring resilience by gathering data from the CPPS, selecting the proper recovery actions and invoking corresponding recovery services on the target CPS. Finally, *data summarisation and relevance evaluation* techniques are used for supporting the identification of anomalous conditions in the presence of high volume and velocity of data collected through the Sensor Data API. The approach is validated in a food industry real case study.

Keywords Resilient cyber-physical production system · Smart machine · Context-aware resilience · Data summarisation · Data relevance · Big Data · Human-in-the-loop · Digital factory

1 Introduction

The advent of digital technologies in modern factory is leading towards a new wave of manufacturing operations, where important decisions to drive efficient and responsive production systems are made based on facts, in turn extracted from data collected from the shop floor and surrounding environment. Information and communication technology promoted

the design and widespread integration of cyber-physical systems (CPS) with production lines. cyber-physical systems are hybrid networked cyber and engineered physical elements that record data (e.g. using sensors), analyse them using connected services, influence physical processes and interact with human actors using multi-channel interfaces. In industrial production environments, CPS interact each other and with operators who supervise the machines, according to the human-in-the-loop paradigm [1], in the so-called cyber-physical production systems (CPPS). Ensuring efficiency and full operability of such systems, to enforce their *resilience*, is a key challenge [2]. The Oxford English dictionary defines resilience as the ability of a system to recover quickly from a disruption that can undermine the normal operation and consequently the quality of the system. Resilience in CPPS is even more challenging since it must be addressed both on single CPS and at the shop floor level,

✉ Devis Bianchini
devis.bianchini@unibs.it

Ada Bagozi
a.bagozi@unibs.it

Valeria De Antonellis
valeria.deantonellis@unibs.it

¹ Department of Information Engineering, University of Brescia, Via Branze 38, 25123 Brescia, Italy

where interactions between different CPS and humans must be considered. In fact, in modern digital factories, single CPS are fully connected; therefore, changes in one of them may require recovery actions on the others. Moreover, volume, velocity and variety characteristics of sensors data (Big Data), collected from CPPS on which recovery actions are based, must be properly managed in order to ensure efficiency and effectiveness of a resilience approach for CPPS. Finally, implementing resilience for CPPS is made more difficult when on-field operators are engaged in the production process, to perform manual tasks according to the human-in-the-loop paradigm and to supervise the physical system.

In this paper, we propose an approach to design resilient cyber-physical production systems (R-CPPS) in the digital factory domain by addressing the three research challenges mentioned above, namely the complex structure of CPPS, the exploitation of Big Data to ensure resilience and the human-in-the-loop perspective. To this aim, the approach relies on three main pillars: (i) the modelling of each component of the R-CPPS as a *smart machine*, that is, a cyber-physical system equipped with a set of recovery services, a Sensor Data API used to collect sensor data acquired from the physical side for monitoring the component behaviour and an operator interface for displaying detected anomalous conditions and notifying necessary recovery actions to on-field operators nearby the involved smart machine; (ii) a *mediator* at shop floor level, that is, in charge of ensuring resilience by gathering data from the CPPS, selecting the proper recovery actions based on the context, that is, the type of the product, the current production process phase and the involved smart machines, and invoking corresponding recovery services; (iii) *data summarisation and relevance evaluation* techniques, aimed at supporting the identification of anomalous conditions on data collected through the Sensor Data API, dealing with Big Data issues. The mediator is based on a *context model*, which has been presented in [3], designed to capture the organisation of the single CPS within the CPPS and the association between each CPS and the services that have been designed to implement recovery actions. In [3] a service-oriented approach for modelling resilient CPPS has been sketched, in line with other existing approaches [4]. Services form a middleware at the cyber-side to implement recovery strategies, either on single CPS or on the whole CPPS. Failures and disruptions on one or more components of the CPPS and on the whole production process environment are monitored through proper parameters, while service outputs are displayed nearby the component on which recovery actions must be performed.

This paper extends the approach proposed in [3] with the following novel contributions: (i) the architectural model has been improved with the introduction of the concept of smart machine, aimed at encapsulating together recovery services, Sensor Data API and operator interface; smart machines are in turn composed into more complex structures corresponding to the whole CPPS; (ii) the context model on which the mediator is based has been enhanced, highlighting three main perspectives on product, process and smart machines; (iii) the implementation and validation of the overall architecture have been completed, by introducing the data summarisation and relevance evaluation techniques. The approach is validated in a food industry real case study. The goal of this paper, as key research contribution, is to demonstrate how the smart machine and the context models provide the necessary flexibility to face the complexity in the CPPS structure, including the engagement of human operators, who are still present in many industrial plants, and how Big Data issues can be fruitfully managed using the data summarisation and relevance evaluation techniques. In this sense, the smart machine and context models, as well as data summarisation and relevance evaluation techniques, can be conceived as general enough to be applied to different application domains, beyond the real case study used for validation in this paper. They represent the lesson learned to implement R-CPPS in digital factories given the research challenges highlighted above.

The paper is organised as follows: in Sect. 2 related work is discussed, together with cutting-edge features of this approach with respect to the state of the art; in Sect. 3 the approach overview is provided, with the help of the real case study in the food industry; Sect. 4 presents the conceptual modelling of context-aware recovery services; Sect. 5 describes data summarisation and relevance evaluation techniques; Sect. 6 details the procedure for relevance-based selection of recovery services; Sects. 7 and 8 address implementation and experimental evaluation issues, respectively; finally, Sect. 9 closes the paper and proposes some future work.

2 Related Work

To ensure timely recovery actions, efficient and effective anomaly detection techniques constitute a key enabling factor. Anomaly detection has been largely investigated in recent years, covering data from several domains [5–8], including healthcare domain [9, 10] and Industry 4.0 [11]. In [12], we integrated an anomaly detection approach with a CMMS (Computerised Maintenance Management System)

to provide time-limited maintenance interventions. Such approach has been further generalised in [13] and is used here to enable CPPS resilience.

However, resilience also requires techniques and strategies to perform recovery actions and ensure continuity in the system operability also in case of disruptions. A recent ever-growing interest has been devoted to resilience challenges [2, 14, 15], often related to the notion of self-adaptation, as witnessed by an increasing number of surveys on this topic [16–19]. Targets of recent approaches addressing CPS resilience vary from cyber-security [20–23], to cyber-physical power systems [24, 25] and to cyber-physical production systems [26–30]. The first ones are focused on security issues at runtime on the communication layer or to implement resource balancing strategies between the edge and cloud computing layers. In [31], a model-free, quantitative and general-purpose evaluation methodology is given, to evaluate the systems resilience in case of cyber-attacks, by extracting indexes from historical data. On the other hand, cyber-physical power systems approaches [24, 25] are focused on investigating proper recovery strategies to enhance the system resilience in the interconnected power network. System resilience in such applications is focused on minimising the impact of disasters and reconstruction costs. Cyber-security and power systems approaches address research issues that are complementary with respect to the scope of this paper. Indeed, our approach is focused on ensuring CPPS resilience in the digital factory domain, and in the following, a comparison against resilience/self-adaptation approaches, specifically designed for these kinds of CPPS, will be discussed. Moreover, the research challenges highlighted in the introduction have been considered here.

In [26, 27], ad hoc solutions are provided, focusing on single work centres or on the entire production line, without considering the effects of resilience across connected components. In [28], a decision support system is proposed to automatically select the best recovery action based on KPIs (e.g. overall equipment efficiency) measured on the whole production process.

In [29], a failure predictive tool to ensure resilience in CPPS has been proposed. The predictive mechanism is a combination of the dynamic principal component analysis (DPCA) and the gradient boosting decision trees (GBDT), a supervised machine learning algorithm. Thanks to such combination, the proposed tool is able to predict the production failure and has been tested on a real-world CPPS, involved in milling process.

Authors in [30] propose a digital twin (DT) and reinforcement learning (RL)-based production control method in order to ensure resilience in a micro-smart factory (MSF).

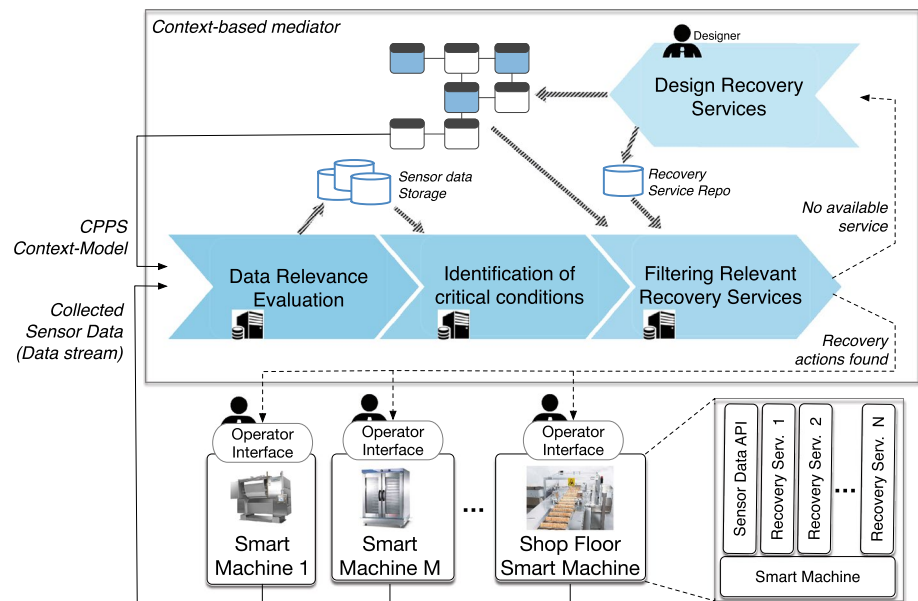
The DT is used for simulations in order to provide the virtual event logs for supporting the learning process of the RL policy network. This method establishes and adjusts the required parameters for production control in MSF when they need to be revised. The RL supports the parameters adjustment through the learning process. Authors in [32] present a reconfiguration algorithm based on first-order logic, which can be integrated directly into the automation software. The goal of reconfiguration is to identify the necessary changes to a system in the presence of faults. Therefore, authors rely on a logical formalism for modelling CPPS, suitable to perform reconfiguration. Moreover, the system dynamics are integrated using gradient information: the sign of the gradient of state variables is used to predict how the system will evolve given an input. Thus, reconfiguration identifies an adaption to the inputs that restore a valid state.

Table 1 reports a comparison with the state-of-the-art approaches on CPPS resilience in the digital factory domain. Compared to these approaches, our solution contributes to the state of the art by introducing a context model, apt to relate recovery services with a fully connected hierarchy of smart machines (from connected devices up to the whole production line at shop floor level). Smart machines are in turn associated with supervising operators. This model enables, on the one hand, to take into account the propagation of recovery effects throughout the hierarchy of connected work centres and, on the other hand, to personalise the visualisation of recovery actions nearby the involved work centres, supporting the human operators who supervise the production line. Another advantage of the proposed approach is its applicability in a human-in-the-loop scenario, making effective the resilience actions even with the involvement of human operators. This situation is very common in some production lines, when the knowledge brought by operators is still of paramount importance.

The adoption of context awareness to implement resilient CPS has been investigated in the cyber-physical Systems (CAR) Project [33]. In the project, resilience patterns have been identified via the empirical analysis of practical CPS systems and implemented as the combination of recovery services. Our approach enables a better flexibility, given by the adoption of service-oriented technologies, and a continuous evolution of the service ecosystem is realised through the design of new services in case of unsuccessful recovery service identification. Authors in [4] share with us the same premises regarding the adoption of service-oriented technologies, modelling processes as composition of services that can be invoked to ensure resilience. This also enables to extend the approach to different kinds of services, beyond the re-configuration ones. Sometimes, re-configuration is

Table 1 Cutting edge features of R-CPPS approach with respect to state-of-the-art approaches on CPPS resilience in the digital factory domain

	Service oriented	Anomaly detection techniques	Propagation throughout the CPPS structure	Types of recovery actions
Barenji et al. [26]	Yes	Multi-agent system	No	Re-configuration
Vogel-Heuser et al. [27]	Yes	Multi-agent system	No	Re-configuration
Galaske et al. [28]	No	Based on OEE KPI	Yes	Re-configuration
Zhang et al. [29]	No	Statistical method (DPCA) combined with machine learning (GBDT)	No	Re-configuration
Park et al. [30]	Yes	Digital twin combined with reinforcement learning	No	Re-configuration
Balzereit et al. [32]	No	N/A	Yes	Re-configuration
Bicocchi et al. [4]	Yes	N/A	No	Re-configuration component substitution
Our Approach	Yes	Distance-based relevance evaluation	Yes	Re-configuration component substitution
	Context-model	Big Data issues	Human-in-the-loop	
Barenji et al. [26]	No	No	No	
Vogel-Heuser et al. [27]	No	No	No	
Galaske et al. [28]	No	No	No	
Zhang et al. [29]	No	No	No	
Park et al. [30]	No	No	No	
Balzereit et al. [32]	No	No	No	
Bicocchi et al. [4]	No	No	No	
Our Approach	Yes	Yes	Yes	

Fig. 1 Overview of the approach for resilient CPPS

not applicable and the operators working nearby the smart machine must be guided during component substitution phases. Finally, the integration of proper methods and techniques to face Big Data issues has not been considered in the compared approaches.

3 Approach Overview

Figure 1 presents the overview of the proposed approach. The designer, who has the domain knowledge about the production plant, is in charge of preparing the context model,

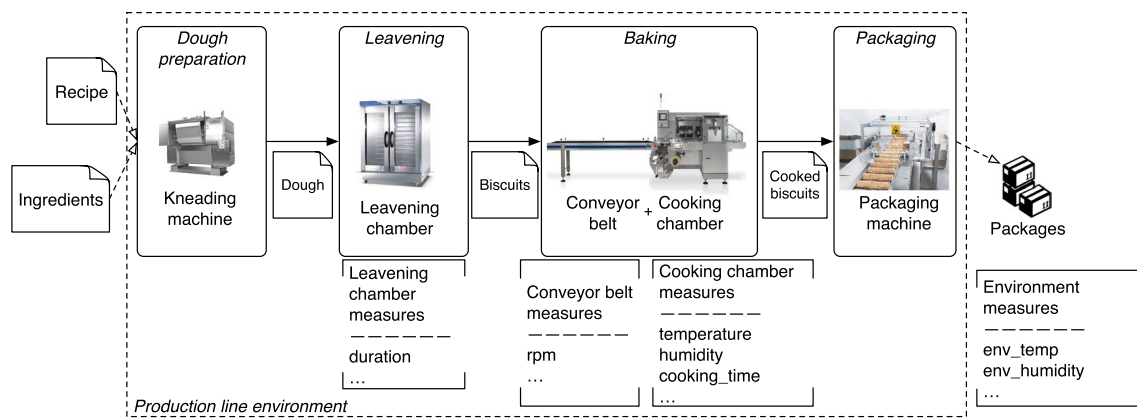


Fig. 2 Production process for the food industry case study

as well as an initial repository of recovery services. In a nutshell, the context model is organised over the three perspectives of the product that is being produced, of the production process and of the structure of smart machines in the CPPS. Each smart machine is associated with services that have been designed to implement recovery actions. The CPPS and its components, modelled as smart machines, are monitored by relying on parameters whose measures are collected from the CPPS. Parameters refer to the status of the product (after quality control), the production process and the status of smart machines. In particular, measures collected on smart machines are gathered as data streams through Sensor Data API and are processed with IDEaaS system [13], a novel solution for data stream exploration and anomaly detection. Anomalies are detected by applying data relevance evaluation techniques, where relevance here has to be intended as a measurable distance from normal working conditions. Critical conditions have to be discovered to trigger recovery services. Therefore, detected anomalies are propagated over the hierarchy of connected components of the CPPS and the corresponding relevant recovery services are filtered among all the available ones defined by the designer. Selected services are executed or, if human intervention is required, are suggested to the on-field operators through the smart machine operator interface, to be triggered according to the operator's on-field experience. The selected services (either automatically or manually triggered) are invoked using the corresponding API of the smart machine. If no recovery services are found, the design of new services in the future is planned.

3.1 Motivating Example

Let's now consider a real case study in the food industry that will be used to validate the proposed approach. The

production process of this case study is shown in Fig. 2. Specifically, we consider a company that bakes biscuits, starting from the recipe and the ingredients up to the finished product (ready-to-sell biscuits). In the production process, the dough is prepared by a kneading machine and let rise in a leavening chamber. Once the biscuits are ready to be baked, they are placed in the oven. The oven is composed of a conveyor belt, mounting a rotating engine and the cooking chamber. By regulating the velocity of the belt through the rpm of the rotating engine, it is possible to setup the cooking time of the biscuits. The temperature and the humidity of the cooking chamber can be regulated as well. Finally, other measures can be gathered at the shop floor level, such as the temperature and the humidity of the whole production line environment. The considered case study presents the characteristics of any fully connected digital factory. In fact, according to the RAMI 4.0 reference architectural model [34], modern digital factories present a hierarchical organisation, from connected devices up to the fully connected work centres and factory (IEC62264/IEC61512 standards). Measures can be gathered from components in this hierarchy and from the operating environment at shop floor level; this allows to detect anomalous working conditions, which may propagate over the whole hierarchy. In the proposed approach, components are modelled as smart machines, such as the conveyor belt and the leavening chamber. Some of them are directly connected (e.g. the conveyor belt and the cooking chamber), and the other ones are part of the whole production line at shop floor level. Note that smart machines are used to model either a single work centre (e.g. the cooking chamber, from which temperature, humidity, cooking time measures are collected through the corresponding Sensor Data API) or the whole production line (from which environment measures are gathered). In Fig. 2, only a subset of measures is showed.

Identification of anomalous working conditions on a smart machine may trigger recovery actions either on the machine itself or on a different one. Let's consider, for

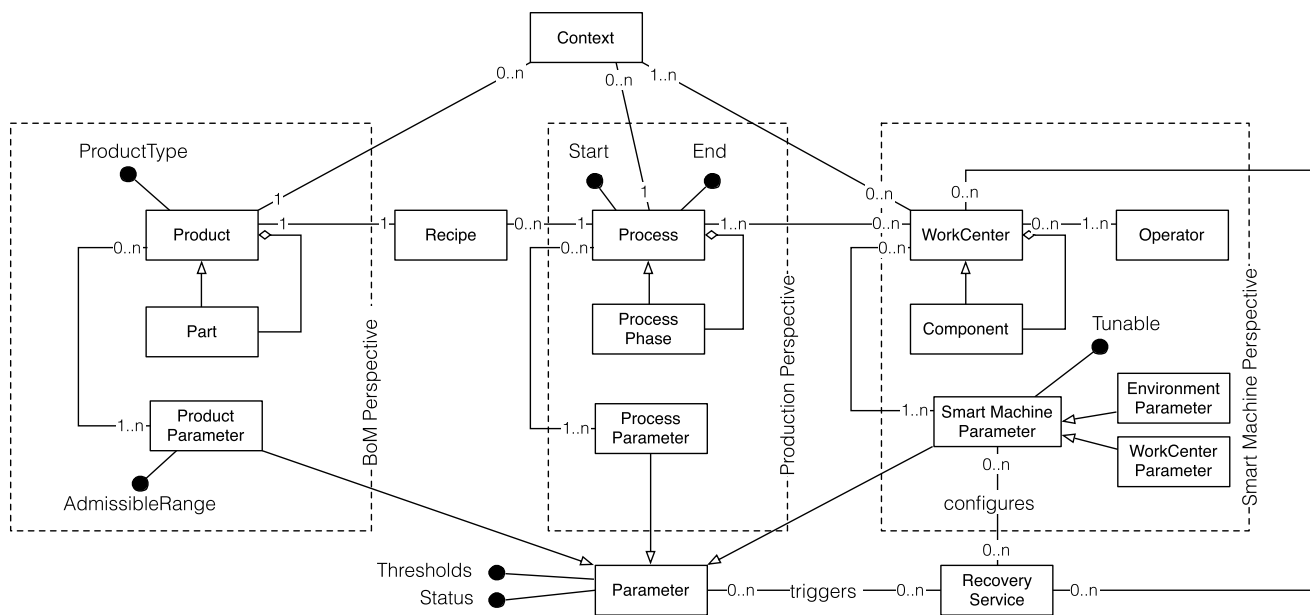


Fig. 3 Resilient CPPS context model

example, an anomaly on the rotating speed of the conveyor belt, which can be detected by measuring the rpm of the rotating engine. Since this anomaly might cause the cookies to burn, a possible recovery action can be triggered to modify the temperature of the cooking chamber to face a longer cooking time. But at the same time the effects of environment temperature must be considered on the dough entering the oven. To this aim, smart machines are equipped with an edge computing layer to connect them to the rest of the digital factory. Recovery actions to be performed on smart machines are implemented as services. Services may represent functions that relate the controlling variables as service inputs (e.g. the rpm of the rotating engine) with controlled ones (e.g. the temperature of the cooking chamber). Services may also be used to send messages to the operator interface of the machine, to support on-field operators, e.g. in suggesting manual operations on the machine such as configurations or substitutions of corrupted parts. In fact, some recovery actions, such as the substitution of some parts in the production line, are costly and should be carefully acknowledged by operators who know the monitored machines. Furthermore, in some machines, parameters to be modified (according to the output of re-configuration services) could be set only manually. In these cases, recovery actions must be modelled as a support for operators. In particular, the proper suggestion must be visualised nearby the involved smart machine, in order to let operators manage the complexity of the system and speed up recovery actions also in those cases where the human intervention is unavoidable.

In the following sections, the main parts of the approach, namely context-aware recovery services and relevance

identification techniques, are detailed, before explaining how to use them for implementing R-CPPS.

4 Context-Aware Recovery Services

4.1 Three-Perspective Context Model

In Fig. 3, we report the context model to support resilience in CPPS. In the figure, only main attributes of the model are shown for the sake of clarity. The model is organised over three perspectives, namely the *Bill of Material (BoM)*, the *Production* and the *Smart Machine* perspectives. The three perspectives are aimed at grouping different kinds of anomalies, which may arise when either quality issues are detected on the product during quality control (BoM perspective), delays occur during the execution of production phases (production perspective) or faults are detected/predicted on the CPPS (smart machine perspective). All the perspectives are modelled in a hierarchical way. Specifically, a *Product* is made of parts (either raw materials or semi-finished parts), which in turn can be composed in a recursive way, according to the bill of material. A product is classified through a *Product Type*. According to the model, also raw materials or semi-finished parts are in turn considered as products (from their suppliers' point of view). For example, the product cocoa shortbread belongs to the biscuits type and is obtained by combining the package (also reporting the product trademark) and cooked dough, in turn made of milk, sugar, chocolate and so on. In order to obtain the ready-to-sell product, the production *Process* (e.g. cocoa

shortbread biscuits preparation) is performed through a set of *Process Phases*, which can be recursively decomposed in further sub-phases as well (according to the production perspective). For example, to prepare cocoa shortbread biscuits, the production process starts from the dough preparation and proceeds with leavening, baking, packaging. In the smart machine perspective, work centres are made of their components and are aggregated in a hierarchy to form the whole production line. For example, as shown in Fig. 2, the production line for the food industry case study is composed of a kneading machine, a leavening chamber, an oven and a packaging machine. The oven is in turn composed of the conveyor belt and the cooking chamber. In this model, the hierarchy includes at the topmost level the entire CPPS. A work centre can be supervised by at least one *Operator* working nearby the machine and for whom the smart machine operator interface has been conceived.

Furthermore, relationships are modelled across different perspectives. In particular, products are crafted through the instantiation of process phases, according to the so-called *Recipe* in manufacturing industry. For example, a recipe is composed of the list of phases that describe how the cocoa shortbread biscuits are prepared and how the ingredients should be mixed. Each phase of the production process is performed by relying on one or more work centres (e.g. the biscuits baking process involves a kneading machine to prepare the dough, a leavening chamber to prepare biscuits, an oven to bake the biscuits and so on). The following definition provides a formalisation of multi-dimensional context model, which will be used in the rest of the paper.

Definition 1 (Context model) We denote a multi-dimensional context model as a pair $CM = \langle D, H \rangle$, where:

- $D = D_{\text{bom}} \cup D_p \cup D_{\text{sm}}$ is a finite set of dimensions, corresponding to the three perspectives illustrated above, namely bill of material (D_{bom}), production process (D_p) and smart machine perspective (D_{sm}); furthermore, $D_{\text{bom}} \cap D_p = \emptyset$, $D_{\text{bom}} \cap D_{\text{sm}} = \emptyset$ and $D_p \cap D_{\text{sm}} = \emptyset$;
- $H = \{h_{\text{bom}}, h_p, h_{\text{sm}}\}$ are three finite hierarchies of dimensions; each hierarchy is defined over one of the three sets D_{bom} , D_p and D_{sm} , respectively; in each hierarchy, a total order $\succeq_{h_i}$ on D_i is established.

4.2 Parameters for Resilient CPPS

Different kinds of parameters are used to monitor the behaviour of a CPPS according to the three perspectives of the context model. On each product or part in the bill of materials, some *Product Parameters* are measured, concerning the quality properties of the product or the part. These parameters are measured when quality controls are performed and their values must stay within an acceptable range. On each

process phase, proper *Process Parameters* are measured as well, concerning the phase duration, which must be compliant with the end timestamp of each phase, as established by the production schedule. Finally, smart machines parameters are gathered to monitor the working conditions of each work centre or the production environment. These parameters are continuously collected as a stream of sensor data through the smart machine Sensor Data API. Their monitoring is performed through a novel data stream anomaly detection approach detailed in Sect. 5. Some smart machine parameters are identified as *tunable*, that is, they are not measured to monitor the status of the work centre, but they can be set to control its behaviour. In all cases, parameters are associated with proper bounds or thresholds, according to the following definition.

Definition 2 (Parameter) A parameter represents a monitored variable that can be measured to ensure CPPS resilience. A parameter P_i is described as a tuple

$$P_i = \langle n_{P_i}, u_{P_i}, \tau_{P_i}, \text{down}_{P_i}^w, \text{up}_{P_i}^w, \text{down}_{P_i}^e, \text{up}_{P_i}^e \rangle \quad (1)$$

where: (i) n_{P_i} is the parameter name; (ii) u_{P_i} represents the unit of measure; (iii) τ_{P_i} is the parameter type, either product, process, work centre or environment parameter; (iv) $\text{down}_{P_i}^w$, $\text{up}_{P_i}^w$, $\text{down}_{P_i}^e$, $\text{up}_{P_i}^e$ are the parameter bounds, which delimit acceptable values for P_i measurements. Let's denote with $\mathbf{P}$ the overall set of parameters.

Parameters bounds are used to establish if a critical condition has occurred on the monitored CPPS. In particular, we distinguish between two kinds of critical conditions: (a) *warning* conditions, which may lead to breakdown or damage of the monitored system or production process failure in the near future; (b) *error* conditions, in which the system cannot operate. Depending on the parameter type τ_{P_i} , parameter bounds in Definition (2) are differently used. For product parameters, measured during product quality control, only a range of acceptable values is provided, therefore $\text{down}_{P_i}^w$ and $\text{up}_{P_i}^w$ are not specified. Process parameters concern delays of production phases; therefore, a comparison is made against phase scheduled ending time only: $\text{up}_{P_i}^w$ and $\text{up}_{P_i}^e$ are the only two specified bounds. For smart machine parameters, all the four bounds are specified, but as explained in Sect. 5, comparison against bounds is not performed on single measures, but on a lossless, summarised representation of measures clusters to face their volume. Parameters bounds are established by domain experts, who possess the knowledge about the production process, or by specific tools, such as the production scheduler, which is used to setup the process parameters bounds.

4.3 Recovery Services

If either quality issues are found on the product or delays are discovered during the execution of a production phase or anomalies are detected on work centres, recovery actions are identified to be performed on a certain smart machine and invoked as *Recovery Services*. In the model, a “trigger” relationship between *Parameter* and *Recovery Service* is established to specify which parameters must exceed the warning and error bounds to trigger the recovery service. A recovery service is associated with a smart machine (see the overview in Fig. 1) and displayed on the operator interface of that machine.

In the current version of the approach, we are considering two kinds of recovery services: (i) *re-configuration services*, which implement resilience by modifying tunable smart machine parameters; (ii) *component substitution services*, which ensure resilience by suggesting possible alternative smart machines. A re-configuration service is used to set differently the tunable parameters on the smart machine, which the service is associated with. It implements a function that maps values of one or more input parameters to values of parameters to be tuned. Formally, a re-configuration service is defined as follows.

Definition 3 (Re-configuration recovery service) A re-configuration recovery service S_j is described as a tuple

$$S_j = \langle n_{S_j}, IN_{S_j}, OUT_{S_j}, SM_{S_j}, ACT_{S_j} \rangle \quad (2)$$

where: (i) n_{S_j} is the service name; (ii) IN_{S_j} is the set of input parameters of S_j ; (iii) OUT_{S_j} is the set of output parameters tuned through the service execution; (iv) SM_{S_j} is the smart machine associated with the recovery service; (v) ACT_{S_j} is the service activation condition.

The activation condition can be one of the following: (a) a product parameter with value that is out of range; (b) a delayed process phase (according to the observation of process parameters); (c) an anomaly detected on a work centre (through work centre parameters monitoring) or at shop floor level (through environment parameters monitoring); (d) a combination of one or more activation conditions mentioned above. An example of re-configuration service is `setOvenTemperature`, to set the cooking chamber temperature of the oven. The input of the service includes the conveyor belt rpm. When an anomaly is detected on the values of conveyor belt rpm, the service properly sets the cooking chamber temperature, to avoid cookies overheating. We remark here that the service is an applicable solution if it returns a cooking chamber temperature that is within acceptable ranges of values.

When a re-configuration is not an applicable solution, other recovery actions must be applied, such as to replace or repair the conveyor belt. A “component substitution” service is used in these cases to suggest the substitution of a component with other alternative smart machines. The alternatives are displayed on the operator interface of the Smart Machine associated with the recovery service. Formally, a substitution service is defined as follows.

Definition 4 (Substitution recovery service) A substitution recovery service S_j is described as a tuple

$$S_j = \langle n_{S_j}, IN_{S_j}, SM_{S_j}^{subst}, SM_{S_j}, ACT_{S_j} \rangle \quad (3)$$

where: (i) n_{S_j} is the service name; (ii) IN_{S_j} is the set of input parameters of S_j ; (iii) $SM_{S_j}^{subst}$ is the set of alternative smart machines that can be used instead of the one associated with the service; (iv) SM_{S_j} is the smart machine associated with the recovery service; (v) ACT_{S_j} is the service activation condition (as defined for re-configuration services).

An example would be the `replaceConveyorBeltRotatingEngine`, which has no output parameters to tune and whose input is the conveyor belt rpm. This service is selected when the rpm is out of predefined ranges of acceptable values. This service is associated with the conveyor belt. The function implemented within the re-configuration services, as well as other information (e.g. execution cost, time) that can be used to choose among different kinds of services, is based on the knowledge of the domain and is set when the service is added. The examples of recovery service types considered here are not exhaustive and may be extended [35].

Let's denote with S the overall set of recovery services, with $S^c \subseteq S$ the set of re-configuration recovery services and with $S^s \subseteq S$ the set of substitution recovery services. In our model, for a Smart Machine it is possible to define one or more, possibly alternative, recovery services. Recovery services can be exposed in different ways, for example as web services, invoked from a local library, etc.

5 Data Relevance Evaluation for Resilience

Measures of smart machine parameters gathered through Sensor Data API form data streams, that is, continuously evolving flows of measures. These parameters are characterised by high volume and velocity and are different from product parameters (whose measures are gathered at a lower frequency when a quality control is performed) and from process parameters (whose measures are collected when a delay with respect to the established schedule is detected

on a process phase). Anomaly detection in data streams has been extensively investigated in recent years [36] and must address research issues such as fast incremental processing of incoming measures, efficient storage of high volume of measures, which accumulate over time, partial knowledge about the data stream, limited to the current flow of records that has been processed and might change in the (near) future.

To face these issues, smart machine parameters are monitored by relying on IDEAAAS [13], a novel approach for data stream exploration and anomaly detection on cyber-physical production systems. The IDEAAAS approach is structured over three main elements, namely: (a) a *multi-dimensional model* that is used to partition the incoming data stream; (b) an *incremental clustering* algorithm, to reduce the volume of incoming data stream through the generation of *lossless data syntheses* that evolve over time; (c) *data relevance evaluation techniques*, which aim at focusing the exploration on the syntheses corresponding to anomalous working conditions of the monitored system. In [13] the IDEAAAS system has been tested on single CPS or work centres. Here, we will describe how to adapt the approach to the R-CPDS design problem.

Context-aware data streams partitioning. In this paper, partitioning of data streams is performed by relying on the context model. The objective of partitioning is to process data streams that are collected in the same context, namely, on the same smart machine (smart machine perspective), engaged in the same type of process phase (production perspective) for the production of the same type of product (BoM perspective). The combination of the smart machine, the process phase and the product type is denoted as *exploration context* and is defined as follows.

Definition 5 (Exploration context) Given a context model $CM = \langle D, H \rangle$, we denote as exploration context γ the combination of dimension instances, one for each $h_j \in H$, used to partition the stream of measures for smart machine parameters. Let's denote with Γ the set of exploration contexts.

With reference to the food industry case study, an example of exploration context is composed of the “biscuit” product type, during the “leavening” phase of the process, using the “leavening chamber” smart machine.

Data summarisation. Data summarisation is used to provide a representation over a set of measures using a reduced amount of information. Moreover, it allows to depict the

behaviour of the system better than single data, which might be affected by noise and false outliers. In the IDEAAAS approach, data summarisation is based on an incremental clustering algorithm that is applied to partition the data streams [37]. Given an exploration context $\gamma \in \Gamma$, the application of the clustering algorithm to the stream of measures incrementally produces a set of micro-clusters $\mu C^\gamma(t) = \{\mu c_1, \mu c_2, \dots, \mu c_n\}$, where n represents the number of micro-clusters in $\mu C^\gamma(t)$. A micro-cluster μc_i is a lossless representation of a set of closed measures in the stream and conceptually represents a working behaviour of the monitored system. For each micro-cluster, the centroid and the radius are calculated as described in [37]. The clustering algorithm has an incremental nature and the set $\mu C^\gamma(t)$ is calculated at a given time t starting from measures collected from timestamp $t - \Delta t$ to timestamp t and updated every Δt seconds. Moreover, the set $\mu C^\gamma(t)$ is built on top of the previous set of micro-clusters $\mu C^\gamma(t - \Delta t)$ for the observed set of parameters in the same exploration context $\gamma \in \Gamma$. For each iteration, the clustering algorithm can be divided in two parts: (i) firstly, each measure corresponds to a new micro-cluster; (ii) existing micro-clusters are updated considering the new collected measures. In particular, whenever a new measure is collected, if the maximum number of allowed micro-clusters q has not been reached yet, a new micro-cluster is created, containing only the new measure. Otherwise, the algorithm determines whether the new measure is absorbed by an existing micro-cluster, according to the distance of the measure from the centroid of the micro-cluster, or a new micro-cluster must be created. In the latter case, before creating a new micro-cluster, an existing one must be deleted or the two closest micro-clusters must be merged together. The algorithm identifies the micro-clusters to be removed by considering the micro-cluster *relevance stamp*, defined as the $\frac{m}{2n_i}$ -th percentile of the timestamps of measures in the micro-cluster, assuming for them a normal distribution. If the relevance stamp of a micro-cluster is older than a threshold τ , then the micro-cluster can be removed. Otherwise, the two closest micro-clusters are merged. The maximum number of allowed micro-clusters q and threshold τ are determined by the amount of main memory available to store the micro-clusters, in order to prevent memory overflow. More details about the incremental clustering algorithm in the IDEAAAS approach are provided in [13].

Figure 4 reports three different steps of the incremental clustering for three smart machine parameters, given an exploration context $\gamma \in \Gamma$. Coloured micro-clusters in

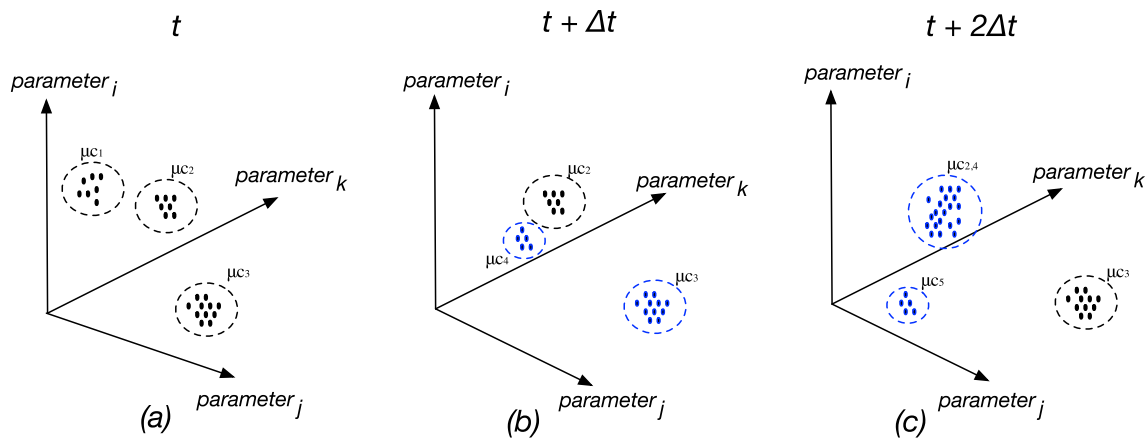


Fig. 4 Evolution of micro-clusters over time considering three smart machine parameters. The exploration context is not shown in the figure

the figure are the ones that have changed from $\mu C^\gamma(t)$ to $\mu C^\gamma(t + \Delta t)$. In Fig. 4b, with respect to Fig. 4a, the micro-cluster μc_1 has been deleted and a new micro-cluster μc_4 has been created; moreover, the micro-cluster μc_3 has been re-positioned. In Fig. 4c the micro-clusters μc_2 and μc_4 are merged and a new micro-cluster μc_5 has been created.

Data relevance evaluation. Once the set of micro-clusters has been produced, data relevance evaluation techniques are used to highlight relevant micro-clusters only. Data relevance is defined as the *distance* of the physical system behaviour from an *expected status*. The status corresponds to the normal working conditions of the system and is denoted here with the set of micro-clusters $\mu \hat{C}^\gamma(t_0)$. This set is tagged as “normal” by the domain expert while observing the monitored system when operates normally. Data relevance evaluation at time t is based on the computation of the distance $\Delta(\mu C^\gamma(t), \mu \hat{C}^\gamma(t_0))$ between the set of micro-clusters $\mu C^\gamma(t) = \{\mu c_1, \mu c_2, \dots, \mu c_n\}$ and $\mu \hat{C}^\gamma(t_0) = \{\mu \hat{c}_1, \mu \hat{c}_2, \dots, \mu \hat{c}_m\}$, where n and m represent the number of micro-clusters in $\mu C^\gamma(t)$ and $\mu \hat{C}^\gamma(t_0)$, respectively, and do not necessarily coincide. The distance $\Delta(\mu C^\gamma(t), \mu \hat{C}^\gamma(t_0))$ is computed as:

$$\frac{\sum_{\mu \hat{c}_i \in \mu \hat{C}^\gamma(t_0)} D(\mu \hat{c}_i, \mu C^\gamma(t)) + \sum_{\mu c_j \in \mu C^\gamma(t)} D(\mu c_j, \mu \hat{C}^\gamma(t_0))}{m + n} \quad (4)$$

where $D(\mu \hat{c}_i, \mu C^\gamma(t)) = \min_{j=1, \dots, n} d(\mu \hat{c}_i, \mu c_j)$ is the minimum distance between $\mu \hat{c}_i \in \mu \hat{C}^\gamma(t_0)$ and micro-clusters in $\mu C^\gamma(t)$. Similarly, the distance $D(\mu c_j, \mu \hat{C}^\gamma(t_0)) = \min_{i=1, \dots, m} d(\mu c_j, \mu \hat{c}_i)$ is the minimum distance between $\mu c_j \in \mu C^\gamma(t)$ and micro-clusters in $\mu \hat{C}^\gamma(t_0)$.

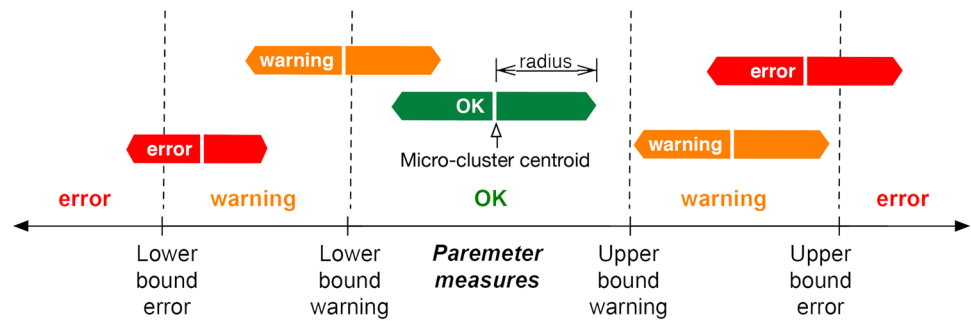
To compute the distance $d(\mu c_a, \mu c_b)$ between two micro-clusters, different factors are combined: (i) the Euclidean distance between micro-clusters centroids $d_{X0}(\mu c_a, \mu c_b)$, to verify if μc_b moved with respect to μc_a and (ii) the difference between micro-clusters radii $d_R(\mu c_a, \mu c_b)$, to verify whether there has been an expansion or a contraction of micro-cluster μc_b with respect to μc_a . Formally:

$$d(\mu c_a, \mu c_b) = \alpha \cdot d_{X0}(\mu c_a, \mu c_b) + \beta \cdot d_R(\mu c_a, \mu c_b) \quad (5)$$

where $\alpha, \beta \in [0, 1]$ are weights such that $\alpha + \beta = 1$, used to balance the impact of terms in Eq. (5). In the experiments described in [13] the two terms of Eq. (5) are equally weighted, that is, $\alpha = \beta = \frac{1}{2}$. To better understand the rationale behind data relevance evaluation, in Fig. 4 coloured micro-clusters are deemed as relevant due to centroid re-positioning from Fig. 4a, b (see μc_3), micro-cluster creation (see μc_4), micro-cluster removal (see μc_1) and micro-cluster merging from Fig. 4b, c (i.e. variation of the number of micro-clusters, which has an impact on the denominator of Eq. (4), see μc_2 and μc_4).

Roughly speaking, the relevance techniques allow to identify what are the micro-clusters that changed over time (namely, appeared, have been merged or removed) for a specific exploration context $\gamma \in \Gamma$. The set of relevant micro-clusters is denoted with $\overline{\mu C}^\gamma(t) = \{\overline{\mu c}_1, \overline{\mu c}_2, \dots, \overline{\mu c}_k\}$ at time t , where $k \leq n$ and n is the number of micro-clusters $\in \mu C^\gamma(t)$. Relevant micro-clusters are considered in order to speed up the identification of critical conditions and therefore the selection of recovery services, as explained in the next section.

Fig. 5 Comparison against warning and error bounds for smart machine parameters (after data summarisation and relevance evaluation)



6 Relevance-Based Selection of Recovery Services

6.1 Identification of Critical Conditions

The goal of the anomaly detection is to identify critical conditions and send alerts concerning the *CPPS status*. Three different options for the status are considered: (a) *ok*, when the CPPS operates normally; (b) *warning*, when the CPPS operates in anomalous conditions that may lead to breakdown or damage (proactive anomaly detection); (c) *error*, when the CPPS operates in unacceptable conditions or does not operate at all (reactive anomaly detection). The warning status is used to perform an early detection of a potential deviation towards an error status. The identification of the CPPS status is based on the monitoring of smart machine parameters, product parameters during quality control procedure, production process parameters. In particular, smart machine parameters are analysed after performing the incremental clustering algorithm described in Sect. 5. Comparison against warning/error bounds is not performed on each single measure, but at micro-clusters level, considering the micro-cluster centroid and radius, as shown in Fig. 5. This comparison is performed to classify the status of each smart machine for every possible exploration context. Given an exploration context $\gamma \in \Gamma$ at a given time t , its status is classified as: (i) *ok*, if all micro-clusters in $\mu C^\gamma(t)$ are located in the *ok* area as shown in Fig. 5; (ii) *warning*, if there is at least one micro-cluster in $\mu C^\gamma(t)$ that is located in the *warning* area, according to Fig. 5; (iii) *error*, if there is at least one micro-cluster in $\mu C^\gamma(t)$ that is located in the *error* area, according to Fig. 5. The status of a Smart Machine at a time t is classified as:

- *ok*, if the status for all its exploration contexts is *ok*;
- *warning*, if the status of at least one exploration context is *warning*;

- *error*, if the status of at least one exploration context is *error*.

Since the overall number of exploration contexts and the sets of $\mu C^\gamma(t)$ can be very large, considering that a new set of micro-clusters is identified every Δt seconds, the relevance evaluation techniques are used to prune the number of comparisons. Specifically, the above conditions are checked on the set $\bar{\mu C}^\gamma(t)$ of relevant micro-clusters only.

The status of the product is classified in a similar way by considering the values of product parameters with respect to the range of admitted values for each parameter. In this case, the comparison is performed on each single parameter measure when the quality control is performed. The status of product is therefore classified as: (i) *ok*, if the status for all product parameters is *ok*; (ii) *warning*, if the status of at least one product parameter is *warning*; (iii) *error*, if the status of at least one product parameter is *error*. Finally, the status of production process is classified as: (i) *ok*, if all process phases are on time; (ii) *warning*, if at least one process phase exceeds the warning delay time; (iii) *error*, if at least one process phase exceeds the error delay time.

Once the status of smart machines, of the product and of the production process have been classified, they are combined to obtain the overall CPPS status as follows:

- *ok*, if the status of all smart machines, of the product and of the production process is *ok*;
- *warning*, if the status of either a smart machine or the product or the production process is *warning*;
- *error*, if the status of either a smart machine or the product or the production process is *error*.

When the status of the CPPS changes towards *warning* or *error*, a recovery service must be selected to be executed as explained in the following.

6.2 Relevant Recovery Service Selection

Algorithm 1: Relevant recovery services selection

Input: the set $\mathcal{C}$ of CPPS status changes towards **error** or **warning**
Output: set $\bar{S}$ of relevant recovery services

```

1 Function selectRelevantServices( $\mathcal{C}$ ):
2    $\bar{S} \leftarrow \{\}$ 
3    $\mathcal{P}_{error} \leftarrow$  select parameters where (status == error)
4    $\mathcal{P}_{warning} \leftarrow$  select parameters where (status == warning)
5    $SM^e \leftarrow$  select smart machines where (status == error)
6    $SM^w \leftarrow$  select smart machines where (status == warning)
7    $S \leftarrow \text{GETSERVICES}(\mathcal{P}_{error}) \cup \text{GETSERVICES}(\mathcal{P}_{warning})$ 
8   foreach service  $s \in S$  do
9     if (TYPE( $s$ ) == “re-configuration” and OUTPUTSWITHINBOUNDS( $s$ ))
10      then
11         $\bar{S} \leftarrow \bar{S} \cup \{s\}$ 
12      else if TYPE( $s$ ) == “substitution” then
13         $SM_s^{subst} \leftarrow$  select alternative smart machines for  $s$ 
14         $SM_s^{subst} \leftarrow SM_s^{subst} \setminus SM^e \setminus SM^w$ 
15        if  $SM_s^{subst} \neq \emptyset$  then
16           $\bar{S} \leftarrow \bar{S} \cup \{s\}$ 
17        end if
18      end if
19    end foreach
20  return  $\bar{S}$ 
21 End Function

```

Algorithm (1) reports the pseudo-code to select relevant recovery services. Formally, a relevant recovery service is defined as follows.

Definition 6 (Relevant recovery service) A recovery service $S_k \in \mathbf{S}$ is defined as relevant if:

- the status of at least one parameter that is related to the service through the “triggers” relationship has been classified as **warning** or **error**;
- if S_k is a re-configuration service (i.e. $S_k \in \mathbf{S}^c$), then the value of all the output parameters OUT_{S_k} resulting from service execution must fall inside the allowed parameter bounds;
- if S_k is a substitution service (i.e. $S_k \in \mathbf{S}^s$), then there must exist a smart machine SM such that $SM \in SM_{S_k}^{subst}$ and $SM \neq SM_{S_k}$; SM must be available and not in a **warning** or **error** status; the machine SM_{S_k} will be substituted with SM .

Firstly, parameters are inspected to identify anomalous conditions that may trigger recovery services (lines 3–4). To this aim, the status of all CPPS parameters is checked, according to the criteria exposed in Sect. 6.1. These steps produce the set $\mathcal{P}_{error}$ of parameters in the **error** status and the set $\mathcal{P}_{warning}$ of parameters in the **warning** status. These sets are used to identify the set S of recovery services that are candidates to be selected, through the GETSERVICES

function (line 7). We remark here that the order followed to populate S is important, since services to recover from error conditions have highest priority. Similarly, the set SM^e of smart machines in the **error** status and the set SM^w of smart machines in the **warning** status are retrieved (lines 5–6). These sets will be used to select proper substitution recovery services.

Furthermore, before including each service of S among the relevant ones, the candidate service is checked to verify the following conditions (line 8–18): (a) if the service type is “re-configuration” (TYPE function) and the value of its output parameters does not exceed any parameter bound (OUTPUTSWITHINBOUNDS function), then the candidate service is selected as relevant (lines 9–10); (b) if the service type is “substitution” (line 11), the alternative smart machines associated with the service are identified (line 12) and among them those machines that are currently unavailable because they are in **warning** or **error** status (lines 5–6) are excluded (line 13); the candidate service is selected as relevant if the resulting set of smart machines suitable for the substitution is not empty (lines 14–16).

Example. The setOvenTemperature service described above is a candidate to be identified as relevant if an anomaly has been detected on the values of rotating engine rpm in the conveyor belt. Since the service type in this case is “re-configuration”, before proposing the service to the operator, the value of the service output must be

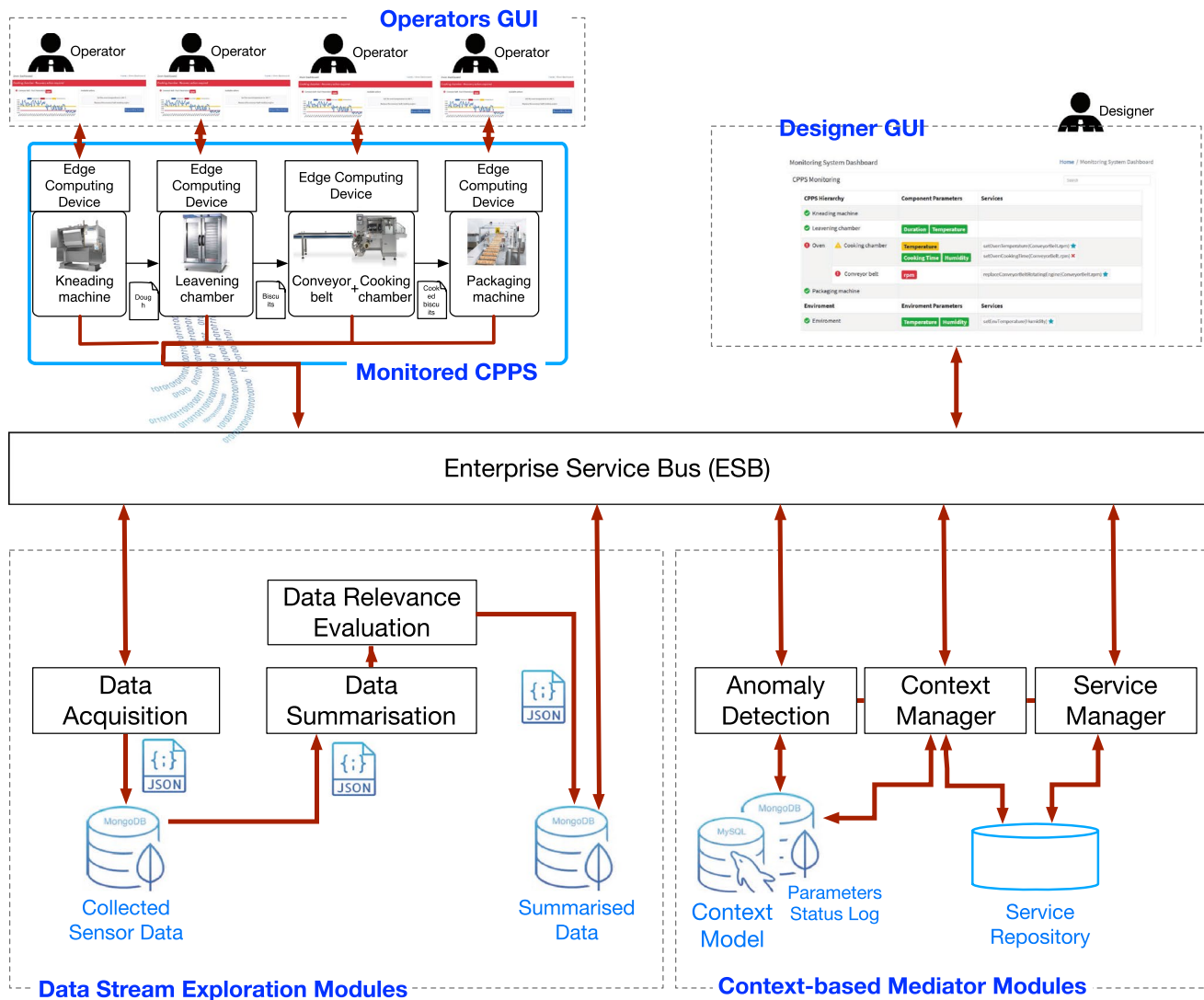


Fig. 6 Approach architecture (in the upper left part of the figure it is reported the cyber-physical production system introduced in Fig. 2, while GUIs are discussed in detail in Sect. 8.4)

compared against parameter bounds of the cooking chamber temperature.

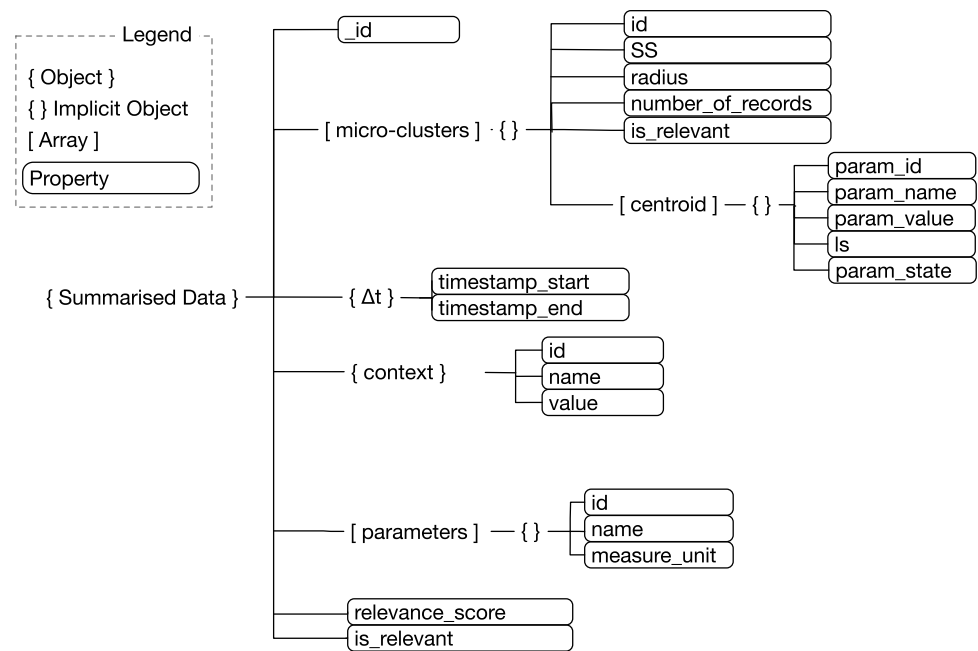
Once a relevant recovery service S_k has been identified, it is executed or suggested to the operator assigned to the SM_{S_k} in order to be confirmed and executed (for re-configuration services) or to proceed with a substitution of the affected part (for substitution services). In particular, the second type of services again highlights the role of human operators as the final actuators of recovery actions on the production line. The service-oriented architecture does not exclude that in the future some kinds of services (such as the re-configuration ones) that require the human intervention can be made fully automatic. Nevertheless, this is not the focus of the paper. We remark here that another feature of the approach is that the information about the recovery actions to undertake, as

results of recovery services execution, are proposed only to the operators supervising the involved smart machines and visualised on the operator interface. This means that “the right information is made available on the right place only”, avoiding useless data propagation if not necessary and information flooding towards operators that may hamper their working efficiency.

7 Implementation

Figure 6 reports the approach modular architecture. The architecture includes the following modules: (i) *Data Stream Exploration Modules*, aimed at acquiring sensor data from the CPPS and performing data summarisation and relevance

Fig. 7 The structure of the JSON document generated every Δt seconds after the application of data summarisation and relevance evaluation techniques



evaluation; (ii) the *Context-based Mediator Modules*, which implement the identification of critical conditions, the selection and management of recovery services and the management of the context model; (iii) the *Designer GUI* and the *Operator GUI*. The architecture is conceived as a service-oriented system, where an enterprise service bus (ESB) is adopted to manage the interactions between modules, as detailed in the following. The overall architecture is currently implemented in Java.

Data Stream Exploration Modules. As explained in Sect. 5, the data stream exploration is aimed at monitoring streams of measures acquired from smart machines through Sensor Data API. Measures are pre-processed by the *Data Acquisition* module, which is in charge of partitioning data streams by relying on the *Context Model*. To this aim, this module interacts with the *Context-based Mediator* through the ESB, to obtain the current exploration context, according to Definition (5).

Collected measures are saved within a NoSQL database (*Collected Data*) as JSON documents using MongoDB technology. Each JSON document represents a record of collected measures within a set of parameters and includes the timestamp and a reference to the exploration context in which the measures have been collected.

Collected measures are processed by the *Data Summarisation* module every Δt seconds. At each iteration, a JSON document is generated, containing the set of micro-clusters according to the analysed parameters and the exploration context, stored within the MongoDB database (*Summarised Data*). The *Data Relevance Evaluation* module implements the relevance evaluation techniques described in Sect. 5 and updates the above mentioned JSON file.

Figure 7 shows the structure of the JSON document defined to represent relevance-enriched summarised data. In particular, the document contains information about the set of micro-clusters, the time interval Δt over which the micro-clusters has been generated, the exploration context, information about the observed parameters and the relevance score computed by the *Data Relevance Evaluation* module as described in Sect. 5.

Moreover, labelled summarised data are visualised: (a) on the *Designer GUI* to let the designer monitor the overall evolution of the CPPS; (b) on the *Operator GUI* of the involved smart machine, to let the on-field operator to better understand the behaviour of the machine.

Context-based Mediator Modules. Labelled summarised data are fed to the *Anomaly Detection* module, which is in charge of identifying critical conditions, as presented in Sect. 6.1. This module is also in charge of being notified about product parameters values out of admitted range and process parameters values corresponding to delayed production phases. All this information is saved in the *Parameters Status Log* repository.

During anomaly detection, the *Context Manager* is invoked in order to contextualise the incoming data. To this purpose, the *Context Manager* will provide the following information: (i) an identifier for the context; (ii) a set of parameters to be monitored; (iii) the involved smart machine; (iv) the product type that is being produced; (v) the process phase. Such information is extracted from the *Context Model* database.

Once anomalous conditions are detected, the *Service Manager* module is notified with the identifier of the exploration context and the list of parameters whose status

Table 2 Summary of the characteristics of the experimental data set in the considered real case scenario

Highest data acquisition frequency	$0.1 \div 0.5$ s
Number of parameters	8
Average number of measures per month	105×10^6
Number of smart machines	4
Hierarchy depth of components (smart machine perspective)	$1 \div 3$

changed towards error or warning. The *Service Manager* will search for relevant recovery services, saved in the *Service Repository*, according to Algorithm (1). The result of the service selection is sent to the operator assigned to the smart machine associated with selected recovery services, on which recovery actions must be performed. The communication is once again managed by the ESB.

On the other hand, when the identification of relevant recovery services fails, the *Service Manager* reports the unsuccessful service selection to the designer, through the *Designer GUI*, including all data of the context and available services. The designer will take into account such report for future design of new services. The new services, when available, will be registered in the *Service repository* through an interaction with the *Service Manager*. The *Context Manager* is notified as well, in order to update the *Context model*.

8 Experimental Evaluation

8.1 Experimental Setup

This section describes the experimental evaluation performed to validate the resilience approach described in this

paper. The main goal of the experimental evaluation is to verify the ability of data summarisation and relevance evaluation techniques to efficiently and promptly detect anomalies on the monitored system in the presence of Big Data and the availability of dashboards to display suggested recovery services in a human-in-the-loop scenario, by relying on the smart machine and context models. In the former case, the volume, velocity and variety of data are a potential bottleneck that may hamper anomaly detection and the following aspects have been evaluated: (i) processing time required to promptly detect anomalies and activate recovery services; (ii) the quality of data relevance evaluation techniques. In the last part of this section, proof-of-concept dashboards to display suggested recovery services are discussed.

A data set has been built containing data collected on a real scenario, starting from the motivating example introduced in Sect. 3. The data set contains measures of 8 parameters, collected from 4 smart machines. In total, the data set contains $\sim 1,12$ billions of measures collected during a year. Table 2 provides a summary of the characteristics of the data set.

In the following, the performed experiments will be described. Experiments have been run on a MacBook Pro Retina, with an Intel Core i7-6700HQ processor, at 2.60 GHz, 4 cores, RAM 16GB.

8.2 Experiments on Data Summarisation and Relevance Evaluation Efficiency

The experiment on processing time has been run in order to prove that data summarisation and relevance evaluation techniques can be efficiently computed, thus facing high acquisition rates. Figure 8 reports the average response time for each collected measure, with respect to the Δt interval in

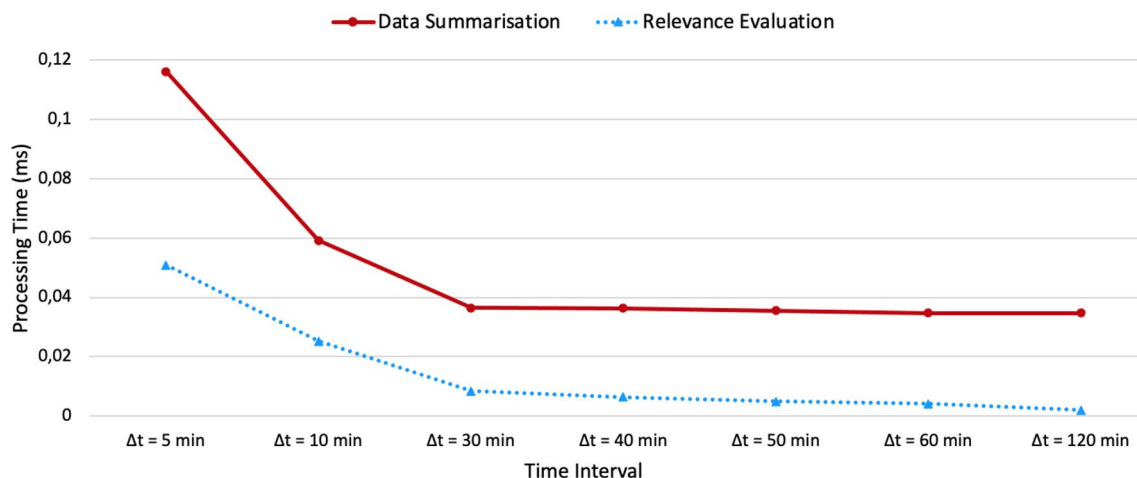


Fig. 8 Average response time to apply data summarisation and relevance evaluation techniques that are used to activate recovery services. Processing time is normalised with respect to the number of collected measures

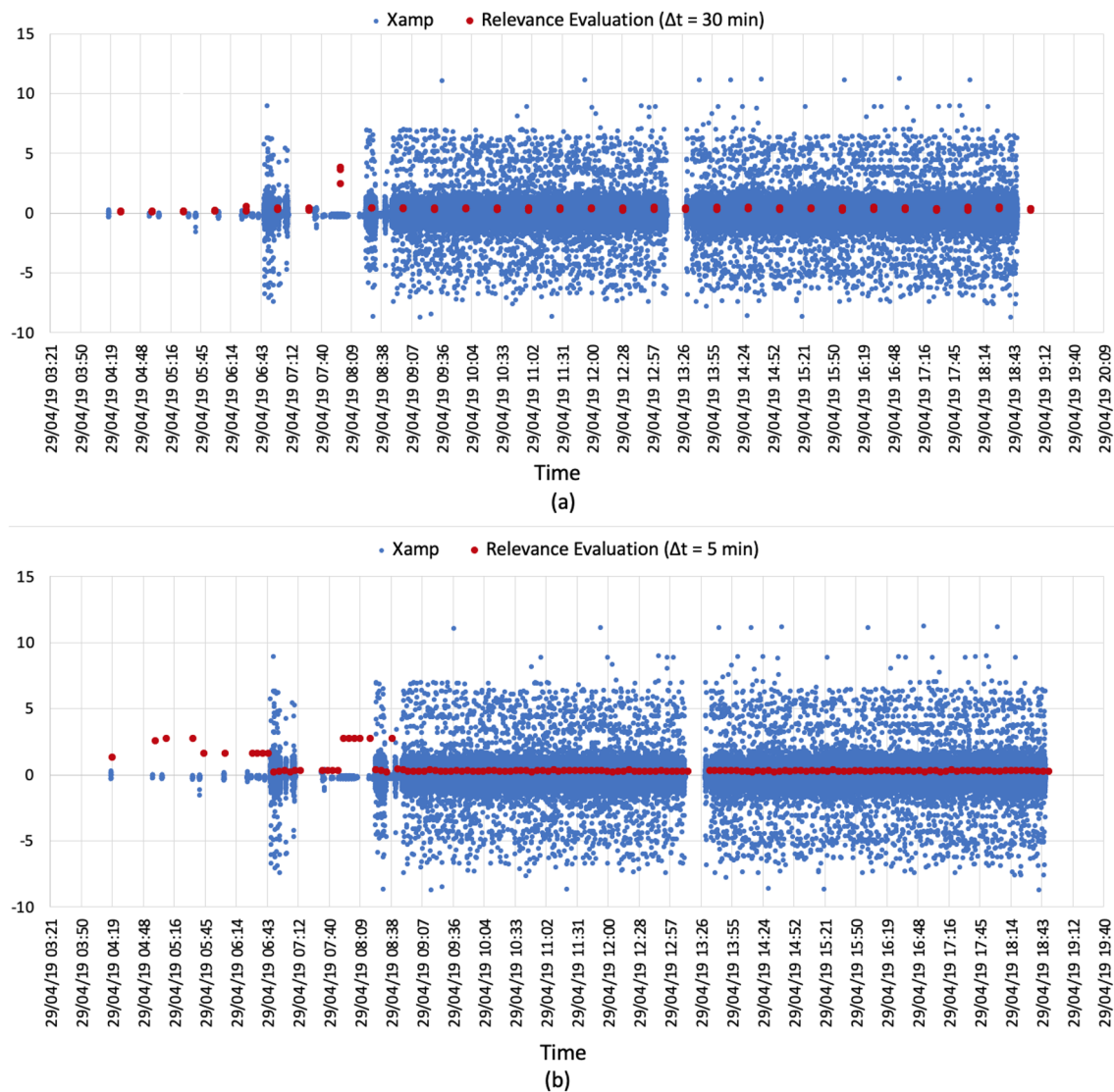


Fig. 9 Relevance evaluation for anomaly detection applied every 30 min ($\Delta t = 30$ min) (a) and every 5 min ($\Delta t = 5$ min) (b)

which data summarisation and relevance evaluation are performed. In a counter-intuitive way, lower Δt values require more time to process data. This is due to the nature of the incremental data summarisation algorithm. In fact, for each algorithm iteration (i.e. every Δt seconds), a certain amount of time is required for operations such as opening/closing connection to the data storage layer and retrieval of the set of micro-clusters previously computed. Therefore, increasing Δt values means better distribution of this overhead over more measures. On the other hand, higher Δt values decrease the promptness in identifying anomalous events, as the frequency with which the data relevance is evaluated is lower. However, even in the worst case ($\Delta t = 5$ min), ~ 2290 measures per second can be processed, thus demonstrating how the approach is able to face proper data acquisition rates for the application domains like the ones in which the approach

has been applied. Moreover, in such applications, the deterioration of monitored CPPS may increase the occurrence of anomalies, and for this reason the CPPS should be monitored at a higher frequency. Indeed, after applying re-configuration services, Δt should be set as lower as possible, based on the computational resources, in order to have a near real-time detection of anomalous events. On the other hand, after applying substitution services, Δt can be increased, in order to mitigate the cost of the computational resources. In fact, in the latter case, the probability to have anomalous events is lower, considering that the substituted smart machine is new. Results shown in Fig. 8, however, demonstrate how the anomaly detection techniques enable to quickly process sensor data collected from smart machines at an acquisition rate that is acceptable for a real case resilience application.

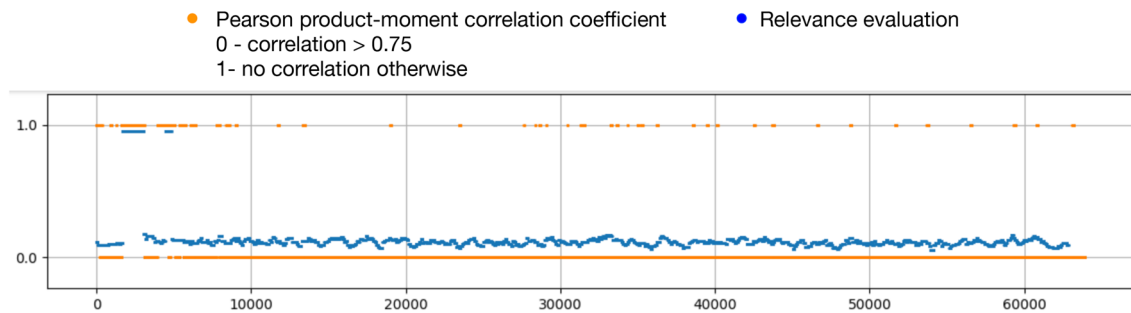


Fig. 10 Pearson product-moment correlation coefficient vs. relevance evaluation ($\Delta t=5\text{min}$)

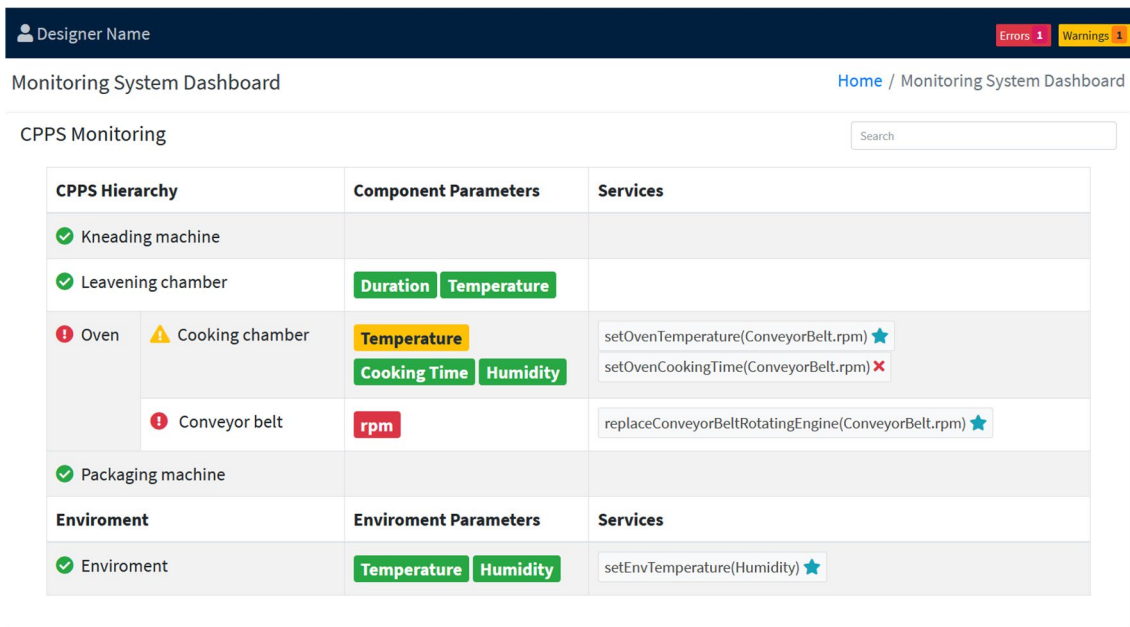
8.3 Experiments on Relevance Evaluation Effectiveness

The aim of this experiment has been to evaluate the ability of relevance evaluation techniques to detect anomalous behaviour in the real data set. In Fig. 9, the difference between original stream of measures and the use of data relevance evaluation result is showed. The considered parameter Xamp is the current absorption for one of the smart machines in the food industry case study. In Fig. 9a, data summarisation and relevance evaluation have been applied every 30 min ($\Delta t = 30\text{ min}$), blue points represent single data points in the stream, corresponding to single measures for Xamp parameter, collected every 500 msec. Exploration of such data for anomaly detection appears as really difficult for an operator who is observing the collected data, but also automatic algorithms are less effective due to the high presence of noise and false outliers. Red points represent the value of the cluster distance between the current set of micro-clusters and the set of micro-clusters computed when the monitored system was working in normal conditions, according to the relevance evaluation techniques described in Sect. 5. Operators who performed a maintenance intervention on the monitored machine confirmed that on the morning of April 29th, before 9am, some anomalies occurred on the machine. The lack of precision was due to the manual collection of maintenance reports, which is still very usual in production lines with an important involvement of human operators. Figure 9 shows how the relevance evaluation techniques have been able to detect anomalies on the real data stream. However, after setting $\Delta t = 30\text{ min}$, anomalies before 7am have not been detected. The reason can be attributed to the presence of more rapid changes in the data stream. In Fig. 9b, techniques have been applied every 5 minutes ($\Delta t = 5\text{ min}$). Figure shows how by reducing the Δt value, as expected, the promptness in identifying anomalous conditions increases and the visualisation of the cluster distance from stable working conditions is more evident with respect to Fig. 9a.

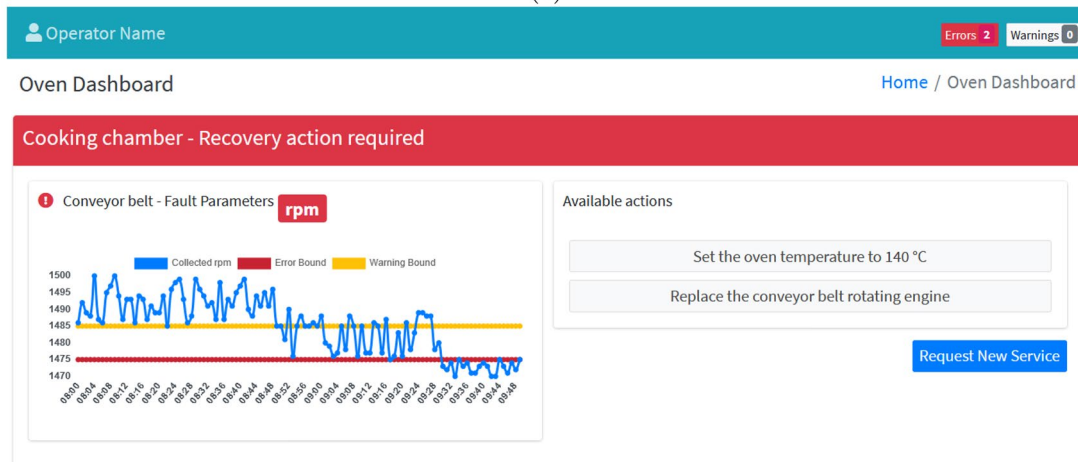
According to these considerations, we quantified the capability of detecting anomalies on the collected measures using the Pearson correlation coefficient (PCC) $\in [-1, +1]$, which estimates the correlation between the real variations and the detected ones. In the experiment, the best PCC value is higher than 0.85 in the case $\Delta t = 10\text{ minutes}$, which represents a strong correlation. Figure 10 reports a comparison between the relevance evaluation metric presented in this paper and a well-known correlation algorithm, the Pearson product-moment correlation coefficient, which searches for recurrent patterns in the considered data. In the figure, the value has been set to 1.0 if anomalies in the pattern are detected (that means that the coefficient is lower than 0.75), and to 0.0 otherwise. As shown in the figure, the correlation is quite sensible to outliers, and anomalies are detected in the whole set of data. On the other hand, thanks to the incremental clustering algorithm, the approach described in this paper is more robust to anomalies, and noise in the data is mitigated by the clustering, detecting only real anomalies and not leading to false alarms.

8.4 Proof-of-Concepts Dashboards

Proof-of-concepts dashboards have been designed, one for the designer (Monitored System Dashboard, MSD) and one for the operator who supervises the smart machine involved in the resilience process. Figure 11 reports the MSD (a) and the Oven Dashboard (b). The functions of dashboards are based on the organisation of smart machines in the CPPS and their relationships with recovery services, according to the context model. The effectiveness of dashboards to support recovery service selection, also in the presence of humans in the loop and given the complexity of CPPS structure, has been validated in the real scenario of food industry, where rotating engine rpm values of the conveyor belt, which exceed the error bounds, have been detected. To recover from the anomaly, three services have been selected as relevant ones, namely `setOvenTemperature`, `setOvenCookingTime` and



(a)



(b)

Fig. 11 Dashboards for context-driven monitoring and recovery phases: **a** designer view; **b** operator view

replaceConveyorBeltRotatingEngine. However, the setOvenCookingTime service is not applicable, because this service controls the cooking time by changing the rpm of the belt rotating engine, which is already in error state. The MSD displays the structure of the CPPS and, for each smart machine, the parameters and the available services. The measures of the parameters that exceed the error bounds are highlighted in red, the ones that exceed the warning bounds in yellow and the others in green. On the list of services, the blue stars identify services that have been proposed to recover from the anomaly and the red cross on the setOvenCookingTime means that the service is not applicable. On the other hand, on the Oven Dashboard shown in Fig. 11b, only warning or error events that require

recovery actions on the oven are displayed. Not applicable services are not shown to avoid hampering the efficiency of the operator. The operator can also reject the suggestions by pushing on “Request new service” button and, in that case, a notification “Service not found” will be properly displayed on the MSD for designing future countermeasures.

Although the validation of dashboards has been performed on specific recovery services in the considered real case scenario, the implementation of the MSD and the operator dashboard can be conceived as a proof of concept that demonstrates the usefulness of these tools in easing the task of human supervisors to enable CPPS resilience, when their presence is required because recovery actions must be manually performed or at least triggered, being their automatic

execution not feasible or not yet available for the level of digitalisation of industrial plants.

8.5 Discussion About Experimental Evaluation

As suggested by [38], ensuring resilience in the presence of human operators becomes a critical task and needs to be correctly addressed. Indeed, when dealing with operators, recovery actions may be slow and error prone. However, when dealing with small-medium enterprises (SMEs), such as the one considered in our case study, human operators are very often required to supervise the industrial machines and the operators' experiences is fundamental to monitor the production process. In such conditions, to support the operator and, at the same time, to guarantee effectiveness and efficiency of the resilience system, we rely on the data summarisation and relevance evaluation techniques. Considering their experimental validation, it is clear how data summarisation and relevance evaluation techniques help in reducing the processing time and attracting the operators' attention on relevant data only. Data summarisation and relevance evaluation enable to deploy resilience solutions in the presence of Big Data, which may hamper the final results also in case of fully automatic recovery strategies. Moreover, carefully modelling the CPPS structure as an ecosystem of tightly interconnected smart machines and adopting a context model that relates smart machines to different types of recovery services is crucial to guide the operator during the recovery actions. Thanks to the designed services, the operators' decisions are not arbitrary or solely based on their experience, but they take into account the state of the system when the services are executed. In the future, the approach will be used to design a knowledge base, which contains a proper formalisation of operators' experience and their behaviour when interacting with the CPPS [39].

9 Concluding Remarks

In this paper, we proposed an approach for resilient cyber-physical production systems (R-CPPS) in digital factories domain, taking into account the complex structure of CPPS (modelled as an ecosystem of interconnected CPS influencing each other), the presence of humans in the loop (who may hamper the efficiency and effectiveness of recovery actions) and the exploitation of (Big) sensor data collected from monitored CPPS and surrounding environment to enable resilience, in which volume, velocity and variety of data are addressed. Complex structure, humans in the loop and Big Data exploitation for CPPS resilience are the three main research challenges investigated in this paper. To this aim, the approach relies on: (i) the modelling of each component of the R-CPPS as a *smart machine*, that is, a cyber-physical

system equipped with a set of runtime recovery services, a Sensor Data API used to collect sensor data acquired from the physical side, for monitoring the component behaviour, and an operator interface for displaying detected anomalous conditions and notifying necessary recovery actions to on-field operators nearby the involved machine; (ii) a *context model*, used to put together smart machines and on-field operators, and a *context-based mediator* at shop floor level, that is, in charge of ensuring resilience by gathering sensor data from smart machines, selecting the proper recovery services, and invoking them on the smart machines by notifying the on-field operators; (iii) *data summarisation and relevance evaluation* techniques, aimed at supporting the identification of anomalous conditions on data collected through the Sensor Data API in the presence of Big Data characteristics. A validation of the approach has been performed in a real food industry case study as a possible proof of concept, but other scenarios will be investigated. Future efforts will be also devoted to the improvement of service selection criteria, for example defining a cost model and using simulation-based modules to predict the effects of recovery actions on the production process, like the ones described in [28]. To this aim, smart machines can be further enriched with the simulation facility, through which predictions can be carried on by combining both real-time data gathered on-field through the Sensor Data API and historical (summarised) data properly stored on the cyber side. Moreover, in order to protect data privacy while conserving original data without any transformation, vertical fragmentation techniques can be applied [40]. Finally, recovery services design and management will be further investigated in order to provide support to the service manager (e.g. mechanisms to classify missing recovery services and support the creation of new ones). Once anomalous situations are detected, recovery services should be applied to either bring the monitored CPPS back to normal working conditions (*backward recovery*) or automatically apply corrections to the current state, until an acceptable state is reached (*forward recovery*) [41].

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Nunes D, Silva JS, Boavida FA (2018) Practical introduction to human-in-the-loop cyber-physical systems. Wiley IEEE Press, Hoboken
- Bennaceur A, Ghezzi C, Tei K, Kehrer T, Weyns D, Calinescu R, Dustdar S, Hu Z, Honiden S, Ishikawa F, Jin Z, Kramer J, Litoiu M, Loreti M, Moreno G, Muller H, Nenzi L, Nuseibeh B, Pasquale L, Reisig W, Schmidt H, Tsigkanos C, Zhao H (2019) Modelling and analysing resilient cyber-physical systems. In: Proceedings of the 14th international symposium on software engineering for adaptive and self-managing systems (SEAMS), pp 70–76
- Bagozi A, Bianchini D, De Antonellis V (2020) Designing context-based services for resilient cyber physical production systems. In: Proceedings of 21th international conference on web information systems engineering (WISE), pp 474–488
- Bicocchi N, Cabri G, Mandreoli F, Mecella M (2019) Dynamic digital factories for agile supply chains: an architectural approach. *J Ind Inform Integ* 15:111–121
- Li H, Wang Y, Wang H, Zhou B (2017) Multi-window based ensemble learning for classification of imbalanced streaming data. *World Wide Web* 20:1507–1525
- Yin J, Tang M, Cao J, Wang H, You M, Lin Y (2020) Adaptive online learning for vulnerability exploitation time prediction. In: Proceedings of the 21st international conference on web information systems engineering (WISE), pp 252–266
- Pang G, Shen C, Cao L, Hengel AVD (2021) Deep learning for anomaly detection: a review. *ACM Comput. Surv.* 54:2
- Deepak P, Savitha S (2020) Fair outlier detection. In: Proceedings of the 21st international conference on web information systems engineering (WISE), pp 447–462
- Sarki R, Ahmed K, Wang H, Zhang Y (2020) Automated detection of mild and multi-class diabetic eye diseases using deep learning. *Health Inform Sci Syst* 8(1):1–9
- Liu F, Zhou X, Cao J, Wang Z, Wang T, Wang H, Zhang Y (2020) Anomaly detection in quasi-periodic time series based on automatic data segmentation and attentional LSTM-CNN. *IEEE Trans Knowl Data Eng* 1:1
- Kamat P, Sugandhi R (2020) Anomaly detection for predictive maintenance in Industry 4.0: a survey. In: *E3S Web Conference* 170
- Bagozi A, Bianchini D, De Antonellis V, Marini A, Ragazzi D (2017) Big data summarisation and relevance evaluation for anomaly detection in cyber physical systems. In: Proceedings of 25th international conference on cooperative information systems (CoopIS 2017), pp 429–447
- Bagozi A, Bianchini D, De Antonellis V, Garda M, Marini A (2019) A relevance-based approach for big data exploration. *Future Gener Comput Syst* 51–69
- Shrivastava A, Didehban M (2019) Invited: software approaches for in-time resilience
- Klaeger T, Gottschall S, Oehm L (2021) Data science on industrial data: todays challenges in brown field applications. *Challenges* 12:1
- Ratasich D, Khalid F, Geissler F, Grosu R, Shafique M, Bartocci E (2019) A roadmap toward the resilient internet of things for cyber-physical systems. *IEEE Access* 7:13260–13283
- Musil A, Musil J, Weyns D, Bures T, Muccini H, Sharaf M (2017) Multi-disciplinary engineering for cyber-physical production systems. In: Chapter in patterns for self-adaptation in cyber-physical systems, pp 331–368
- Moura J, Hutchison D (2019) Game theory for multi-access edge computing: survey, use cases, and future trends. *IEEE Commun Surv Tutor* 21(1):260–288
- Haque MA, Shetty S, Gold K, Krishnappa B (2021) Realizing cyber-physical systems resilience frameworks and security practices. In: *Security in cyber-physical systems*, pp 1–37
- Mouelhi S, Laarouchi M-E, Cancila D, Chaouchi H (2019) Predictive formal analysis of resilience in cyber-physical systems. *IEEE Access* 7:33741–33758
- Mertoguno JS, Craven RM, Mickelson MS, Koller DP (2019) A physics-based strategy for cyber resilience of CPS
- Bi S, Wang T, Wang L, Zawodniok M (2019) Novel cyber fault prognosis and resilience control for cyber-physical systems. *IET Cyber Phys Syst Theory Appl* 4:304–312(8)
- Januario F, Cardoso A, Gil P (2018) Resilience enhancement through a multi-agent approach over cyber-physical systems. In: Proceedings of the 10th international conference on information technology and electrical engineering (ICITEE), pp 231–236
- Wu G, Li M, Li ZS (2020) Resilience-based optimal recovery strategy for cyber-physical power systems considering component multistate failures. *IEEE Trans Reliab* 1–15
- Tapia M, Thier P, Gossling-Reisemann S (2020) Building resilient cyber-physical power systems. *Theorie und Praxis* 29(1):23–29
- Barenji R, Barenji A, Hashemipour M (2014) A multi-agent RFID-enabled distributed control system for a flexible manufacturing shop. *Int J Adv Manuf Technol* 71(9):1773–1791
- Vogel-Hauser B, Diedrich C, Pantförder D, Göhner P (2014) Coupling heterogeneous production systems by a multi-agent based cyber-physical production system. In: Proceedings of the 12th IEEE international conference on industrial informatics (INDIN), pp 713–719
- Galaske N, Anderl R (2016) Disruption management for resilient processes in cyber-physical production systems. In: Proceedings of the 26th procedia CIRP 50:442–447
- Zhang Y, Beudaert X, Argandoña J, Ratchev S, Munoa J (2020) A CPPS based on GBDT for predicting failure events in milling. *Int J Adv Manuf Technol* 111(1):341–357
- Park KT, Son YH, Ko SW, Noh SD (2021) Digital twin and reinforcement learning-based resilient production control for micro smart factory. *Appl Sci* 11:7
- Murino G, Armando A, Tacchella A (2019) Resilience of cyber-physical systems: an experimental appraisal of quantitative measures. In: Proceedings of the 11th international conference on cyber conflict (CyCon), pp 1–19
- Balzereit K, Niggemann O (2021) Gradient-based reconfiguration of cyber-physical production systems. In: Proceedings of the 4th IEEE international conference on industrial cyber-physical systems (ICPS), pp 125–131
- Context-Active Resilience in Cyber Physical Systems (CAR) European Project, (2018) <http://www.msca-car.eu>
- Hankel M, Rexroth B (2015) The reference architectural model industrie 4.0 (RAMI 4.0). In: ZVEI
- Pumpuni-Lenss G, Blackburn T, Garstenauer A (2017) Resilience in complex systems: an agent-based approach. *Syst Eng* 20(2):158–172
- Blázquez-García A, Conde A, Mori U, Lozano JA (2021) A review on outlier/anomaly detection in time series data. *ACM Comput Surv* 54:3
- Aggarwal C, Han J, Wang J, Yu P (2003) A framework for clustering evolving data streams. In: Proceedings of the 29th international conference on very large data bases (VLDB), pp 81–92
- Muller T, Jazdi N, Schmidt J-P, Weyrich M (2021) Cyber-physical production systems: enhancement with a self-organized reconfiguration management. In: Proceedings of the 14th procedia CIRP conference on intelligent computation in manufacturing engineering vol 99, pp 549–554

39. Singh R, Zhang Y, Wang H, Miao Y, Ahmed K (2020) Investigation of social behaviour patterns using location-based data: a melbourne case study. *EAI Endorsed Trans Scal Inform Syst* 8:31
40. Ge Y-F, Cao J, Wang H, Zhang Y, Chen Z (2020) Distributed differential evolution for anonymity-driven vertical fragmentation in outsourced data storage. In: *Proceedings of the of 21st international conference on web information systems engineering (WISE)*, pp 213–226
41. Bradley D, Tyrrell A (2000) Hardware fault tolerance: an immunological solution. In: *Proceedings of the international conference on systems, man and cybernetics*, pp 107–112