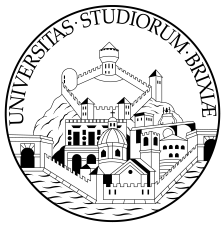


UNIVERSITÀ
DEGLI STUDI
DI BRESCIA

DOTTORATO DI RICERCA IN INGEGNERIA DELL'INFORMAZIONE

Curriculum in Ingegneria Informatica e Automatica

Settore Scientico Disciplinare ING-INF/05

XXXIII Ciclo

Methods and Techniques for Big Data Exploration

Author:

Ada BAGOZI

Supervisor:

Prof. Devis BIANCHINI

PhD Coordinator:

Prof. Costantino DE ANGELIS

Abstract

The collection, organisation and analysis of large amount of data (Big Data) in different application domains often require the involvement of experts for the identification of relevant data, without being overwhelmed by volume, velocity and variety of collected data. According to the Human-In-the-Loop Data Analysis paradigm, experts explore data to take decisions in unexpected situations, based on their long-term experience. The IDEAaS (Interactive Data Exploration As-a-Service) approach is presented, apt to enable Big Data Exploration (BDE). In the approach, novel techniques have been developed: (i) an incremental clustering algorithm, to provide summarised representation of collected data streams; (ii) a Multi-Dimensional Model to organise summarised data and enable data exploration according to different analysis dimensions; (iii) data relevance evaluation techniques, to attract the experts' attention on relevant data only during exploration. Moreover, given the ever increasing volume and velocity of data streams in many real world applications, parallel implementations of data streams clustering algorithms have been widely investigated. To this purpose, a parallel approach, called P-IDEAaS, for massive and evolving data streams clustering that adopts a multi-level strategy is presented. Firstly, the Multi-Dimensional Model based on exploration facets is used to perform a coarse-grained partition of the data stream. Furthermore, other fine-grained levels of parallelization are applied. The novel characteristic of this approach is the capability of tuning the application of parallelization levels, depending on an evaluation of the relevance of the incoming data points. Data relevance is used in order to force a stronger parallelization (and therefore higher resource usage) only when necessary, that is, in presence of relevant data. The aim is to provide an approach to adapt the selection and prioritisation of parallelization levels to the data stream complexity and data relevance, taking into account the availability of resources in the distributed processing architecture. In the first part of the thesis, after an introduction of the approach (Chapter 1) and an analysis of the related state of the art (Chapter 2), the core components of the IDEAaS approach will be described (Chapter 3). In Chapter 4, the implementation of the IDEAaS system is detailed, explaining its experimental validation. The evolution of the IDEAaS system through the introduction of the parallel implementation of the incremental clustering algorithm (P-IDEAaS) will be detailed in Chapter 5. The second part of the thesis will be devoted to the

description of how the IDEAaS approach has been successfully applied to three real case studies: (i) an anomaly detection scenario in the Industry 4.0 domain, to monitor the health status of a multi-spindle CNC machine used for the manufacturing of metal raw material, supporting the identification of unknown anomalous conditions (Chapter 6); (ii) a remote monitoring application in the Healthcare domain, to support medical doctors in controlling the health status of discharged patients during the recent pandemic emergency due to the COVID-19 virus (Chapter 7); (iii) to implement context-based resilient system through the surveillance of data collected from the entire production line in the food industry domain (Chapter 8). Finally, Chapter 9 close the thesis by highlighting some possible future work.

Abstract

La raccolta, l'organizzazione e l'analisi di grandi quantità di dati (Big Data) in differenti contesti applicativi richiede spesso il coinvolgimento degli esperti per l'identificazione di dati rilevanti, evitando allo stesso tempo che essi vengano sopraffatti dal volume, dalla velocità e dalla varietà dei dati raccolti. Secondo il paradigma denominato HILDA (Human-In-the-Loop for Data Analysis), gli utenti esperti esplorano i dati per prendere decisioni relative a comportamenti inaspettati ed anomali del sistema analizzato, basandosi sulla propria esperienza pregressa. In questa tesi verrà presentato IDEAaS (Interactive Data Exploration As-a-Service), un approccio per l'esplorazione e l'elaborazione di grandi quantità di dati, allo scopo di focalizzare l'attenzione di utenti esperti su dati rilevanti. A tal fine, IDEAaS si basa su: (i) un algoritmo di clustering incrementale, utile per fornire una rappresentazione sintetica dei dati generati sotto forma di un flusso continuo (data stream); (ii) un Modello Multi-Dimensionale, per organizzare i dati clusterizzati e per permettere l'esplorazione secondo diverse prospettive di analisi; (iii) un modello di rilevanza, per attirare l'attenzione degli esperti unicamente sui dati rilevanti, durante l'esplorazione. Considerando il costante aumento del volume e della velocità dei dati raccolti sotto forma di data stream in applicazioni reali, è stata studiata un'implementazione parallela dell'algoritmo di clustering incementale per questo tipo di dati. Tale implementazione, denominata P-IDEAaS, costituisce un approccio parallelizzato per il clustering di data stream massivi ed in continua evoluzione, e adotta una strategia multilivello. Prima di tutto, il Modello Multi-Dimensionale, organizzato nelle diverse prospettive di analisi, viene utilizzato per eseguire un partizionamento ad alto livello del flusso di dati, che poi vengono elaborati utilizzando diversi livelli di parallelizzazione. Un aspetto innovativo dell'approccio risiede nell'utilizzo della rilevanza dei dati proprio per regolare tali livelli di parallelizzazione. Infatti, il modello di rilevanza viene utilizzato per aumentare il grado di parallelizzazione (che di conseguenza porta ad un corrispettivo aumento delle risorse computazionali richieste) solo quando è necessario, cioè in presenza di dati rilevanti. Lo scopo è quello di riuscire a fornire un metodo per adattare la selezione dei livelli di parallelizzazione dando priorità ai dati che sono identificati come rilevanti, in modo da considerare la disponibilità di risorse nell'architettura di elaborazione distribuita. Nella prima parte della tesi, dopo l'introduzione (Capitolo 1) e

un'analisi dello stato dell'arte (Capitolo 2), saranno descritti i componenti principali dell'approccio IDEAaS (Capitolo 3). Nel Capitolo 4 verrà riportata l'implementazione e una validazione preliminare del sistema IDEAaS. L'implementazione parallelizzata dell'algoritmo di clustering incrementale (P-IDEAaS) verrà descritta nel Capitolo 5. La seconda parte della tesi sarà dedicata a presentare l'applicazione di IDEAaS in tre casi di studio reali: (i) in un contesto di industria 4.0, dove l'approccio è stato utilizzato per il monitoraggio dello stato di funzionamento di macchine multimandrino, al fine di identificare delle anomalie sotto forma di stati di funzionamento sconosciuto durante la lavorazione del materiale grezzo (Capitolo 6); (ii) in ambito sanitario per il monitoraggio remoto, da parte dei medici competenti, dello stato di salute di pazienti dimessi durante il contenimento della pandemia causata dal virus COVID-19 (Capitolo 7); (iii) nel settore dell'industria alimentare per la realizzazione di sistemi resilienti tenendo in considerazione il contesto di produzione, tramite il monitoraggio dei dati raccolti dell'intera linea di produzione (Capitolo 8). Infine, nel Capitolo 9 verranno discusse le conclusioni del lavoro di tesi e una serie di possibili sviluppi futuri.

Contents

Part I

The approach

Chapter 1

Introduction

1.1 Big Data Exploration

The volume, velocity and variety of real time data (Big Data), collected in different application domains and fostered by the widespread diffusion of Internet of Things (IoT) technologies, raised new research challenges on the collection, organisation and exploration of Big Data [1, 2]. In many big data applications, data are collected as a continuous stream, that must be properly processed taking into account its velocity, the volume of content that is incrementally gathered and the variety of formats and data structures in the stream. Data streams are dynamic, continuous, massive sequences of data points, with typically unpredictable input rate, very common in real world applications [3, 4]. Data streams are used in stock and traffic monitoring, network management, sensor data analysis, event detection and reaction. They are used, for example, to model data collected from embedded IoT components in Cyber-Physical Systems, in order to implement predictive maintenance, or gathered from wearable sensors and devices, for providing remote monitoring in Healthcare applications. Indeed, volume and velocity of real time data as collected in dynamic contexts of interconnected systems, as well as the endless and incremental collection of data streams, pose additional issues in order to implement Big Data Exploration (BDE) in an efficient way. First of all, high volume calls for solutions that provide users with a compact view over the large amount of collected data. Therefore, data summarisation techniques have been proposed and adapted to the data streams. Among proposed techniques, incremental clustering has been recently investigated, in order to: (i) reduce the amount of data; (ii) identify the concept drift in the data stream, i.e., a deviation from a normal/stable working condition, without being hampered by the noise present in the stream. Moreover, in a BDE context,

another critical issue is given by the complexity of analysis dimensions (in terms of their number and heterogeneity), that impacts on data variety and require a proper organisation of summarised data. To this aim, the following aspects should be considered: (i) data points in the stream can be associated to different analysis dimensions, whose number is proportional to the complexity of the observed phenomenon and of the monitored system (for example, measures collected from a Cyber Physical System in a smart factory for predictive maintenance purposes may refer to different tools used by the system or different steps of the manufacturing program that is being executed); (ii) not all data points are equally relevant (for example, concerning remote monitoring of industrial Cyber Physical Systems, observation rate should be increased only when parameters values are close to specific thresholds). Multi-dimensional modelling of data streams can be fruitfully exploited to partition the input stream according to different *exploration facets*, thus reducing the data complexity. Indeed, according to the "Human-In-the-Loop Data Analysis" vision [5], experts are required to explore data for taking decisions in unknown situations, based on their long-term experience. Data exploration [1] and exploratory data analysis [2] have been defined as multi-step processes, where data can be browsed through iterative refinements, since the user is not able to specify his/her requirements using a precise query, and for this reason should be guided through new information discovery. In addition to the organisation of data according to different exploration facets, the development of proper metrics able to quantify the *relevance* of data with respect to a specific exploration goal is of paramount importance. Relevance has been defined in literature [1] as the distance from an expected value. In applications of BDE where data streams are analysed in order to identify anomalous situations, data relevance should be quantified by measuring the difference in the shape of data stream with respect to normal working conditions.

Given the issues presented above, BDE is addressed in this thesis by highlighting three main concepts, namely *data summarisation*, *multi-dimensional modelling* and *data relevance evaluation*. On these three pillars, the IDEAaS (Interactive Data Exploration As-a-Service) approach has been developed. The approach, specifically conceived for big data streams, relies on the following novel techniques, properly combined to realise BDE under a Human-In-the-Loop vision:

- an incremental clustering algorithm, that aggregates, in the so-called *micro-clusters*, data collected as streams of numeric features; micro-clusters represent

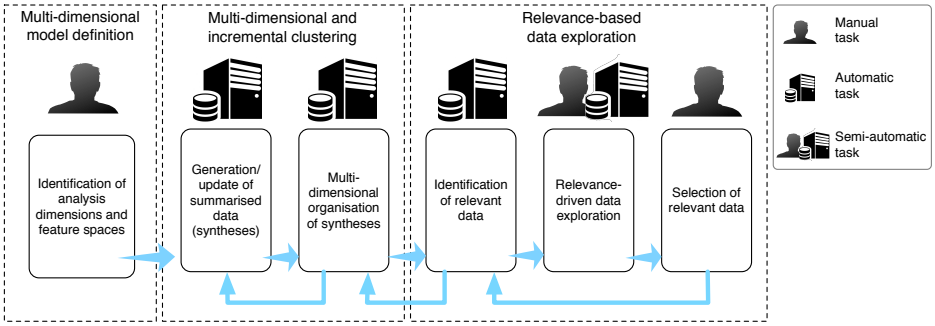


Figure 1.1: The IDEAaS overview.

a working behaviour of the monitored system, to provide summarised representation of data streams; micro-clusters are in turn organised into *snapshots* to enable exploration of different portions of the data streams;

- multi-dimensional organisation of summarised data, to allow data exploration according to different analysis dimensions;
- data relevance evaluation techniques, to support the identification of micro-clusters and snapshots that correspond to unexpected behaviours and to attract the experts' attention on them during exploration.

The IDEAaS approach has been applied in Industry 4.0 and healthcare monitoring domains, to implement new solutions for anomaly detection, remote monitoring and design of resilience systems. In the following, the main pillars of the approach, as well as the case studies in which IDEAaS has been tested, will be introduced.

1.2 Approach overview

The overview of the IDEAaS approach is reported in Figure 1.1. The approach is articulated over three main phases: (i) Multi-Dimensional Model definition, in which the experts manually identify the analysis dimensions and the features spaces to be monitored; (ii) multi-dimensional and incremental clustering, to provide a summarised representation of collected measures and organise them in the Multi-Dimensional Model; (iii) relevance-based data exploration, to support the experts during selection and exploration of relevant data only. Figure 1.1 also specifies whether tasks are manually, automatically executed or semi-automatic.

Multi-dimensional model definition. Measures collected from the monitored system are further enriched with details about analysis dimensions, that contribute to describe the specific conditions in which measures have been taken. For instance, considering remote monitoring of a Computer Numerical Control (CNC) machine that is manufacturing raw material (e.g., metal, wood), the tool that is being used when the measure has been collected, or the kind of material that is being manufactured, can provide additional useful information to distinguish among different working conditions. In this respect, the observation of different combinations of measured features is strictly related to the working conditions in which such observation occurs. Considering again the example of the CNC machine introduced above, and in particular remote monitoring of a spindle used for working raw metal, the spindle rolling friction torque increase and the tool wear are two possible problems that are frequently monitored on this kind of machines. Monitoring in these cases is performed through the collection of power absorption and kinematic features such as accelerations. If an increased power absorption is detected disregarding the tool that is used, it is possible to identify a problem in the spindle rolling friction torque increase. On the other hand, if the increase in absorbed power is related only to the usage of a particular tool, this can be recognised as a symptom of a possible tool wear. Therefore, aspects such as the tools used to shape the raw material can be considered as perspectives according to which data exploration can be performed. These can be modelled as analysis dimensions, that are used to create a Multi-Dimensional Model in which summarised data is organised for data exploration.

Multi-dimensional and incremental clustering. In order to manage and explore large quantity of data, an incremental clustering algorithm is applied on collected measures for different combinations of observed features. Multi-dimensional and incremental clustering phase, as shown in Figure 1.1, is articulated over two subtasks:

- *generation/update of summarised data (micro-clusters)* - the clustering algorithm is applied to summarise collected data and offers a two-fold advantage: (a) it gives an overall view over measures, using a reduced amount of information;

(b) it allows to represent the behaviour of the monitored system better than single measures, that might be affected by noise and false outliers while observing a given physical phenomenon of interest; in literature, different approaches to data stream clustering have been investigated; amongst them, incremental ones, such as CluStream [6], have emerged as promising solutions when treating data arriving at high rates; the peculiarity of such algorithms is to rely on the notion of lossless aggregation of data, denoted as *data micro-cluster*; a micro-cluster conceptually represents a working behaviour of the monitored system, corresponding to a set of measures that are close each other in the space of observed features;

- *multi-dimensional organisation of micro-clusters* - generated micro-clusters are organised according to different analysis perspectives through the Multi-Dimensional Model defined above; however, the iterative nature of the data exploration process requires that experts must be able to extract and explore different portions of the data stream; therefore, the incremental clustering approach has been enriched here with the generation of *snapshots* for the exploration over time of the data stream; multi-dimensional organisation of micro-clusters and snapshots are used to support experts during data exploration.

Relevance-based data exploration. Once the sets of micro-clusters and snapshots have been generated and organised within the Multi-Dimensional Model, relevance-based data exploration is articulated over three sub-tasks, as shown in Figure 1.1:

- *identification of relevant data* - in literature, data relevance has been defined as the distance from an expected status [1]; in the presented approach, data relevance evaluation is performed by computing the differences between the micro-clusters set that represents the current behaviour of the monitored system and the set that represents its normal working behaviour, using the novel metrics that will be detailed in this thesis; the notion of data relevance has been extended to snapshots as well, including the definition of *relevant snapshot*;

- *relevance-driven data exploration* - starting from relevant micro-clusters and snapshots that have been identified, the expert may perform data exploration over the analysis dimensions of the Multi-Dimensional Model; exploration is guided by the system in order to identify the working conditions which relevant micro-clusters correspond to;

- *selection of relevant data* - once the expert has been guided towards such working conditions through the relevance-based exploration in the Multi-Dimensional Model, novel techniques are applied to prune available data, thanks to the exploitation of relevant snapshots, and to show to the expert how the observed phenomena on the monitored system evolved towards the relevant status just identified.

Considering that the incremental clustering algorithm is the most time-consuming element of the approach, a parallel version of the algorithm has been proposed. In this version, the multi-dimensional modelling and data relevance evaluation are exploited for enhancing parallel clustering of massive data streams. Specifically, the main effort is focused on the online phase of clustering, where the parallelisation is worth being applied to face data volume and velocity. Novel aspects of the clustering parallelisation are summarised in the following:

- the adoption of the Multi-Dimensional Model to perform a first, coarse-grained partition of data streams, according to a *divide-and-conquer* strategy, to face their complexity;

- the combined application of other fine-grained levels of parallelisation: (i) a parallelisation based on a buffering mechanism, that splits the data stream into portions of data points on which processing is performed in parallel; (ii) a parallelisation over the set of micro-clusters that are generated and change over time; parallelisation at these levels is enforced with data relevance evaluation techniques, while maintaining the scalability and efficiency of the clustering algorithm;

- the exploitation of data relevance evaluation techniques to ensure different priority to parallelisation levels, in order to dedicate more resources for parallelisation in those cases that present higher priority (i.e., higher data relevance); this will also mitigate the overload due to the distribution of processing tasks over the network of computation nodes, which might have a negative impact on algorithm efficiency, when it is not strictly necessary;

The first part of this thesis will be devoted to the presentation of the details about the IDEAaS approach and its core modules, and about parallel implementation of the clustering algorithm. The efficiency and effectiveness of the IDEAaS approach

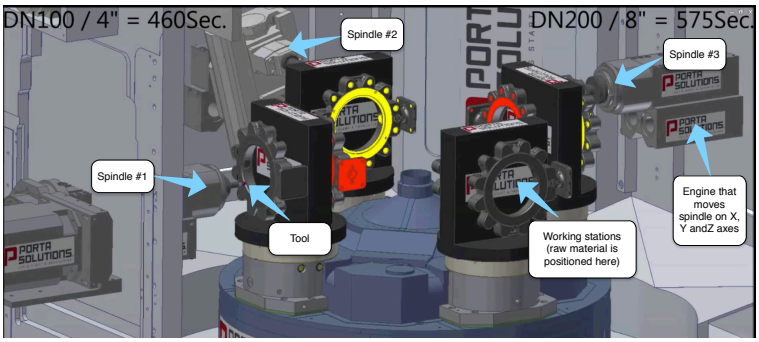


Figure 1.2: The multi-spindle machine from which real time data have been collected for exploration purposes.

have been tested through its application on different case studies, ranging from the Industry 4.0 to the healthcare domain. The second part of this thesis will be devoted to the detailed discussion about these case studies, that will be briefly introduced in the following.

1.3 Case Studies

1.3.1 Anomaly detection of Cyber Physical Production Systems

The first case study that has been considered for the application of the IDEAaS approach concerns anomaly detection in the Industry 4.0 domain, specifically conceived for Cyber Physical Production Systems (CPPS). Similarly, to more general Cyber Physical Systems, CPPS are systems where the physical side interacts with the cyber-side through a continuous collection and organisation of data (e.g., by means of IoT devices) to analyse them in order to take decisions and actuate proper (possibly automated) operations on the physical part. Specifically, CPPS are focused on the production line, where of autonomous and cooperative elements and sub-systems that are getting into connection with each other in situation dependent ways, on and across all levels of production, from processes through machines up to production and logistics networks [7]. In particular, the case study has been focused on a multi-spindle CNC machine produced by an Original Equipment Manufacturer (OEM) As shown in Figure 1.2, the considered multi-spindle machines present 3 spindles, each of them working independently on the raw material. Each spindle is mounted on a unit moved by an electrical engine to perform X, Y, Z movements. The spindle rotation is impressed by an electrical engine and its rotation speed is controlled by

the machine control. Spindles use different tools (that are selected according to the instructions specified within the Part Program [1]) in order to complete different steps in the manufacturing cycle. For each unit, the velocity of the three axes (X, Y and Z), the electrical current absorbed by each of the engines, the value of rpm for the spindle, the percentage of power absorbed by the spindle engine (charge coefficient) are measured.

The aim of the OEM is to understand if it is possible to use real time data collected directly from the machine control for monitoring the spindle axle hardening over time and the tool wear. With spindle axle hardening is referred a specific behaviour of the spindle shaft that turns hard more and more due to different possible reasons: lack of lubrication and bearing wear that may lead to possible bearing failures. Tool wear monitoring is referred to possible tool usage optimisation in order to balance the trade-off between the number of tools used and the risk of breaking the tool during operations that may lead to long down times.

This opens a set of issues, mainly related to data volumes and velocity and the considered application domain.

Firstly, the ability of providing a compact view of the huge amount of data collected from the machine is strongly required. Therefore, a data summarisation approach, where data are observed in an aggregated way, is welcome. At the same time, data aggregations should be observed on the fly, given the highly dynamic nature of the application domain, and efficient computation algorithms are required to summarise data. A multi-dimensional representation of data can be helpful too, since it allows aggregation of data according to different dimensions (e.g., time, monitored spindle, tool used for a specific manufacturing step), that might be related to the observed problems (e.g., spindle axles hardening or tool wear), thus giving proper semantics to the collected data. Moreover, Multi-Dimensional Model enables refinement of the exploration by following the hierarchical organisation of dimensions. Finally, the user who explores data (e.g., operators who are in charge of monitoring the multi-spindle machine to decide about maintenance actions) needs an underlying data-model to enable fast exploration of the available data, to be guided towards only those relevant measures that correspond to spindle hardening or tool

[1] Part Programs are sequences of instructions, which define the actions to be executed by a CNC machine.

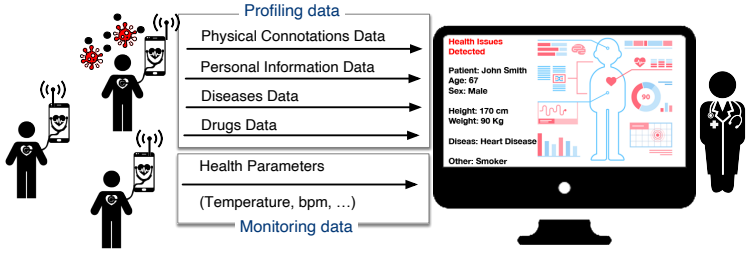


Figure 1.3: Remote monitoring of patients health parameters, with the support of smartphone applications.

wear problems. To this aim, it is required a model of relevance that enables to identify only relevant data on which the user must focus for managing critical situations, taking into account volumes and speed of data collection phase.

1.3.2 Remote monitoring services in the healthcare

Another relevant case study that will be described in this thesis to test the IDEAaS approach concerns the remote healthcare domain. Indeed, during the recent pandemic emergency due to the COVID-19 virus, several apps have been designed and implemented to ease the measure and subsequent evaluation of breath quality, analysing the dynamics of the respiratory act. In particular, the IDEAaS approach has been integrated in one of these projects, where traces of respiratory acts are recorded by the three axial accelerometers, embedded by a smartphone (positioned over the abdomen of the patient). The scenario considered in this project is depicted in Figure 1.3.Traces, plus other patient's health parameters (e.g., temperature, bpm), are exploited to evaluate the quality of breath, raising alerts in the presence of anomalies (e.g., evidences of shortness of breath). In this respect, the app could be employed as a self-evaluation instrument, albeit not conceived to substitute a medical device or the medical examination. Target users of the app are mainly discharged SARS-CoV-2 patients, for determining health improvements or emerging problems (e.g., uprising of breath difficulties). The app guides the user throughout the measurement session, with the support of predefined audio messages, to limit unwanted actions made by the user.

When a patient registers to the app, the following information is collected for profiling: (i) *Physical Connotations*, concerning sex, age, weight and height; (ii) *Personal Information*, regarding habits of the patient (e.g., whether she is a smoker), possible ongoing pregnancy and cholesterol levels; (iii) *Diseases*, which are classified by their

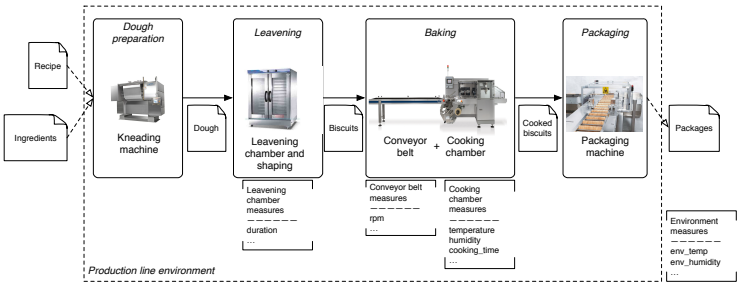


Figure 1.4: Production process for the food industry case study.

virtue (inherited, chronic, congenital) and by the part of the human body they affect; (iv) *Drugs*, assumed by patients and belonging to diverse classes, depending on their specific purpose (e.g., anti-hypertensive, immunosuppressant). Profile data enables the creation of *Patients Groups*, uniquely identified by a combination of the aforementioned data (male patients, male patients over 65 years, etc.); amongst the (potentially vast) set of possible groups, *Relevant Patients Groups* (in brief, RPG) may be recognised, that is, groups which are under the lens of doctors' consideration, as they are more exposed to (a relapse of) the infection risk. RPG may be identified from well known clinical studies (e.g., male subjects are more likely of being infected by SARS-CoV-2 with respect to female ones), but also in a dynamic way, due to the emerging critical health status in several patients within the group.

The integration of IDEAaS approach within the project brought several advantages: (i) monitoring of discharged SARS-CoV-2 patients to promptly detect worsening conditions presents many affinities with remote monitoring of a CPPS, as well as the characteristics of volume, velocity and variety of data streams collected from patients is similar; (ii) data summarisation techniques may help to reduce the amount of data that are transferred from patients and may reduce the burden of medical doctors to explore too much data about their patients; (iii) the multiple aspects to be considered (e.g., diseases, age, sex, habits) suggest the adoption of a Multi-Dimensional Model in order to partition the data streams (such a model might help to organise data streams belonging to distinct patients' groups); (iv) multi-dimensional modelling, together with data relevance evaluation within a patients' group, might help to speed up and ease the efficiency of medical doctors, who can focus on critical patients only.

1.3.3 Context-based resilience in connected Smart Factories

Another application scenario where the IDEAaS approach can be fruitfully applied concerns the design of resilient CPPS. In this case, an example in the food industry has been considered to validate the approach. The production line of the target industry is reported in Figure 1.4. The considered company produces biscuits, starting from the recipe and the ingredients 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. Indeed, 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. Moreover, also the temperature and the humidity of the cooking chamber can be regulated. Finally, other measures can be gathered at the shop floor level, such as the temperature and the humidity of the whole production line environment. In a fully connected digital factory, all these measures can be exploited to ensure the resilience, but the following specific characteristics must be taken into consideration.

C1) A multi-level hierarchy of smart components. According to the RAMI 4.0 reference architectural model [8] (IEC62264/IEC61512 standards), modern digital factories present a hierarchical organisation of the CPPS, from connected devices up to the fully connected work centers and factory. In the considered case study, the production line is composed of different machines and, among them, the oven is in turn composed of the conveyor belt and the cooking chamber. 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, that may propagate over the whole hierarchy, and making the component smarter.

C2) Recovery actions across different components. Identification of anomalous working conditions on a smart component in the hierarchy may trigger recovery actions either on the component itself or on a different one. This is enabled by the strong connection between components over the whole production line. Let's consider, for 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.

C3) The role of the humans in the loop. Re-configuration often represent a sub-optimal recovery solution. Other recovery actions, such as the redundancy of some parts in the production line, are costly and should be carefully acknowledged by workers supervising 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.

Also in this case, several benefits might derive from the integration of the IDEAaS approach. In particular, the Human-In-the-Loop perspective assumed here (C3) suggests the need of multi-dimensional modelling, data summarisation and data relevance techniques to ease the job of workers supervising the monitored machine. Moreover, among the analysis dimensions to be considered during multi-dimensional modelling, the hierarchy of CPPS, from single machines up to the whole production line, enables to properly distinguish anomalies ranging from different components in the factory, thus making the approach more effective and efficient. Finally, also in this case there is an anomaly detection aspect in common with the other considered case studies.

1.4 Thesis outline

This thesis is organised in two parts. In the first one, the core elements of the IDEAaS approach are presented. In the second part, the application of the IDEAaS approach is discussed in three case studies introduced above. In Chapter 2 the background of the approach is presented, including a comparison with the state of the art. Chapter 3 describes the core elements of IDEAaS approach, namely the Multi-Dimensional Model, the incremental clustering algorithm and the relevance-based data exploration. In Chapter 4 the IDEAaS functional architecture and experimental evaluation are discussed. In Chapter 5 a parallel version of the IDEAaS incremental clustering is presented and the associated experimental evaluation is reported.

For what concerns the three applications of the IDEAaS approach, in Chapter 6 the anomaly detection case study in the Industry 4.0 domain scenario is presented.

In Chapter 7 the approach has been applied for remote monitoring services in the healthcare domain. Chapter 8 reports the application of the approach in order to ensure context-based resilience in the Smart Factory.

Finally, Chapter 9 closes the thesis with some final remarks and a discussion about future work.

Chapter 2

Background

The IDEAaS approach and its application in the three paradigmatic case studies touch upon different challenges, corresponding to distinct research fields, that range from general-purpose Big Data Exploration approaches to anomaly detection or engineering of resilient systems. In this chapter, the state of the art in these fields is reported. Given the extent of approaches that addressed all these challenges, in the following the attention will be devoted to recent efforts specifically meant for the domains (e.g., smart factories) for which IDEAaS has been originally conceived or considering data structures (e.g., data streams) on which the proposed approach has been tailored.

2.1 General-purpose Big Data Exploration

As formerly introduced, Human-In-the-Loop Data Analysis (HILDA) is a compelling challenge to tightly couple the human role with the (Big) Data Exploration experience. In this section, focusing on a specific part of the taxonomy reported in [5] an excerpt of the literature concerning approaches on data exploration and exploratory computing has been inspected (in particular, the branch that includes the extraction of insights data for analysis purposes). In this respect, techniques that have been studied to enable BDE are: (i) data summarisation solutions; (ii) proper tools and methods (including interactive data exploration interfaces) to organise data; (iii) metrics apt to attract experts' attention on data of interest.

In [9] authors highlight relevant data within weather time series by means of coloured maps, exploitable by the user, after performing different preparatory steps (e.g., data cleansing). Likewise, in [10] and [11] raw data, gathered from different

data sources, is seamlessly collected in (No)SQL data stores, to be employed by predictive (the former) or corrective (the latter) algorithms to provide the users with insights upon past and future data. With the same philosophy, in [12] a stochastic approach is proposed to enable a probabilistic representation of data with the aim of exploiting the model for inference purposes. In [13] historical stock data is analysed to extract patterns, with the ultimate goal of increasing the prediction accuracy; these patterns also attract users' attention, to favour the exploration task. In [14] remote sensing datasets (containing images) are elaborated within a large-scale data processing system, wherein a tree-like index structure is exploited to efficiently retrieve data, ensuring real-time update capabilities. In the majority of the aforementioned works, quality metrics related to the prediction/correction outcome are presented to the user, in order to assure him/her additional instruments to take (and refine) decisions.

In [15] structured and multi-dimensional OLAP data is incrementally collected and organised in a facet-based cube structure; as a consequence, the role of the user is empowered by the application of data sampling techniques, to guess the next choices he/she will be likely to perform. In [16] incrementally collected data is compressed, fostering data decaying techniques to provide an efficient usage of the available storage capabilities. Herein, occurrence frequency thresholds are employed to highlight values, that are summarised by applying a lossless compression algorithm. Differently, in [17] multi-dimensional data is organised starting from different datasets, before the exploration process begins, adopting range queries to identify partially overlapping windows, shown to the user according to a cost-benefit criterion. In a similar way, AIDE [18] handles multi-dimensional data, which is not collected incrementally, extracts samplings from the DBMS and applies a classifier to infer data relevance to improve the exploration task.

Other approaches dealing with multi-dimensional data mainly focus on how to let the data relevance emerge, in order to suggest promising exploration actions, also depending on past interactions of the user with the data exploration interface. In [19] a recommendation table keeps track of the potential applicable transformations (regarding data, axes and chart types) that can be suggested to the user, together with a value of relevance updated whenever a specific transformation is accepted and applied. In [20] data relevance is achieved through highlights on plotted data. Differently, Hashedcubes [21] provides a surrogate multi-dimensional model for real-time

visualisation, based on optimised pivot-like data structures. Broadly speaking, these approaches are less suitable to deal with Big Data, since an effective data summarisation and organisation strategy, apt to give experts a comprehensive view of data, is missing and experts are basically "lost in the cyber-space of data".

Contribution of the IDEAaS approach. A qualitative discussion on the novelties introduced in IDEAaS with respect to the existing literature is reported in Table 2.2. Specifically, the comparison criteria (incremental data collection, summarisation techniques, multi-dimensional modelling and relevance evaluation) adopted within the IDEAaS research methodology are leveraged to thoroughly motivate the adoption of IDEAaS as a BDE solution. Only a subset of the considered approaches deal with data streams. Given the intrinsic streaming nature of data collected in dynamic contexts, including the Industry 4.0 domain upon which IDEAaS has been primarily tested, research proposals operating on pre-collected (historical) data [10, 9, 18, 13, 17, 14] are less suitable.

Moreover, proper summarisation techniques are advocated, since inspecting each single data in the stream would be very resource demanding, entailing a possible risk of loosing the overall view of the phenomenon being observed. The majority of the considered approaches uses either sampling, compression or clustering (eventually combined together, when appropriate). Regarding the latter, a variant of a renowned incremental clustering algorithm, CluStream [6], has been proposed in this thesis, extending its original implementation by introducing a novel syntheses update policy and the concept of snapshot. Clustering has been chosen due to the fact that it considers the whole set of data (if compared to sampling) and offers the opportunity of aggregating data, according to similarity metrics, forming distinct groups of related measures (if compared to compression).

In IDEAaS, techniques to provide a proper organisation of data, according to facets/dimensions, have been considered, aimed at enabling BDE under different perspectives. Most of the analysed approaches handle multi-dimensional data; nonetheless, only few of them [15, 21, 14] envisage proper data structure to organise (summarised) data and integrate such structure in the exploration process, as implemented through the multi-dimensional model included in IDEAaS.

After data has been suitably labelled with multiple analysis dimensions, the main concern is to attract experts' attention on a subset of data deemed as relevant,

specifically considering the disruptive characteristics of Big Data (volume, velocity, variety). In the investigated approaches, this is mainly achieved through visualisation effects (such as highlights [20, 22], colours [9], labels [23]) or specific methods (e.g., frequency [16], thresholds [24], pattern identification [13]). Instead, in IDEAaS a distance-based approach is applied. The distance value is computed between different clusters sets, taken at distinct time slots, in order to identify only those streams that are going to change (and therefore correspond on the physical side to changing behaviours of the monitored systems). This quantitative distance is used to attract the human observer's attention on streams that evolve only (see next chapter for details on this metrics). The multi-dimensional organisation of streams is also used to ease the observer's task, providing a partition of the data streams. This choice also enables a generalisation of the approach to different BDE scenarios, instead of relying on problem-specific solutions.

As an additional remark, a discussion on how IDEAaS addresses the 5Vs of Big Data (namely, Volume, Velocity, Variety, Veracity and Value) is provided. Specifically, IDEAaS faces *Volume* and *Velocity* by applying a variant of an incremental clustering algorithm, notably apt to operate on data streams, yielding to a lossless representation (summarisation) of data (Section 3.2). Summarisation techniques, used in combination with data relevance evaluation metrics, enable to attract the attention of experts on a subset of data only. Such techniques are also able to address *Variety*, inherently associated to the number and heterogeneity of features, feature spaces and analysis dimensions. Finally, both *Veracity* and *Value* are involved whenever it is necessary to determine which are the important situations to observe. To this aim, relevance evaluation techniques are exploited in IDEAaS to support the identification of unexpected behaviours of the monitored system and, consequently, attract experts' attention towards them (Section 3.3).

2.2 Parallel clustering of Big Data streams

As already remarked in the introduction, the most expensive element of the IDEAaS approach regards incremental clustering, also given the Big Data nature of incoming data streams. Therefore, in this thesis a parallel implementation of the clustering algorithm will be provided as well. Accordingly, the version of the IDEAaS system

	Incremental data collection	Summarisation techniques	Multi-dimensional modelling	Relevance evaluation
Cube Exploration [15]	Yes	Sampling	Yes	No
Semantic Windows [17]	No	Sampling	No	No
Costa et al. [16]	Yes	Compression, Clustering	No	Frequency-based
AIDE [18]	No	Sampling, Clustering	No	No
Orr et al. [12]	No	Sampling, Compression	No	No
Sauvanaud et al. [24]	Yes	Clustering	No	Threshold-based
Yin et al. [23]	Yes	Clustering	No	Labelling-based
Stojanovic et al. [22]	Yes	Clustering	No	Highlighting-based
Hashedcubes [21]	No	N/A	∼ (Auxiliary pivot-like data structures)	No
Saket et al. [19]	No	N/A	No	Recommendation-based
Sansen et al. [20]	No	N/A	No	Highlighting-based
Wang et al. [11]	Yes (Event-based, buffered data batches)	No	No	No
Chang [9]	∼ (Daily data collectable, historical data used)	∼ (Clustering parallel design sketched)	No	Colour-based (on maps)
Jeon et al. [13]	No (Only historical data considered)	∼ (Data aggregation over time)	No	Pattern-based
Birek et al. [10]	No	Clustering (in computational modelling methodology)	No	No
Wang et al. [14]	No (Data subscription and sharing)	N/A	∼ (Tree structure for optimal indexing)	No
IDEAaS	Yes	Clustering	Yes	Based on distance between clusters sets

Table 2.1: Overview of approaches on Big Data exploration.

that includes the new, parallel implementation of the algorithm has been named P-IDEAaS. Recently, many *parallel* implementations of clustering algorithms have been developed. A comprehensive survey on this topic can be found in [25]. Parallel clustering algorithms have been proposed on two families of Big Data platforms: (a) vertical scaling platforms, that rely on a hardware equipment enhancement using Graphics Processing Units (GPU), Field Programmable Gate Arrays (FPGA), multi-core CPU and High Performance Computing (HPC) clusters [26, 27, 28]; (b) horizontal scaling platforms, that rely on Apache Spark, MapReduce and peer-to-peer architectures [29, 30, 31]. There is, however, limited work on clustering data streams in a parallel manner. Among them, [32, 33, 34] are mentioned as parallel clustering on vertical scaling platforms and [35, 36, 37] on horizontal scaling platforms.

In [32] a GPU has been programmed with the main constructs (i.e., threads and

kernels) provided by the CUDA programming model, to ensure a proper parallel implementation of the CluStream algorithm called PaStream. PaStream contains parallel implementation of micro-clusters extraction. In particular, data stream buffering, computation of micro-cluster distance, computation of micro-cluster relevance stamp to decide which micro-cluster to remove and macro-clusters generation in the offline phase have been parallelized. With respect to the P-IDEAaS approach, PaStream does not consider any mechanism to partition the initial data stream beyond buffering, neither any technique to adapt the granularity of parallelisation, as performed through the data relevance evaluation in P-IDEAaS. Furthermore, the volume and velocity challenges of Big Data have a higher impact on the online phase, on which P-IDEAaS is mainly focused. In [33] CUDA parallel implementation of the BIRCH hierarchical clustering algorithm is described. BIRCH relies on the key concept of Clustering Feature (CF), that is a triple containing information (e.g., linear sum, square sum) of all data points in the same cluster. BIRCH builds a balanced tree called CF-Tree, where leaf nodes represent clusters and clustering procedure consists of finding the closest node descending the tree from the root to the leaf nodes. The parallel algorithm is a GPU-based version of BIRCH called GBIRCH, where a master is launched and orchestrates several slaves using CUDA Dynamic Parallelism. Each slave deals with a subset of the data points in the GPU memory, indeed, performing a parallelisation over the data that corresponds to the second parallelisation level described in this thesis (based on buffering). Each slave kernel executes the BIRCH algorithm to assign a data point to a given node of the CF-Tree: if the data point could not be absorbed by the existing nodes, it will be returned to the master and processed sequentially by it, once all the slaves are terminated. With respect to the P-IDEAaS approach and to the approach described in [32], GBIRCH does not consider any mechanism to partition the initial data stream beyond buffering and the only parallel action regards the identification of the dataset node in the CF-Tree, similar to the search of the closest micro-clusters in [32] and in P-IDEAaS. In [34] a vertical scaling solution for parallel incremental clustering is proposed. Specifically, the authors investigate the parallelisation of the EINCKM [38] algorithm, that is composed of three modular steps, namely clusters building, merging and pruning. These steps are executed in sequence, nevertheless each step is parallelized on a set of processors over the same machine and the number of processors is adapted in order to grant algorithm performance without exceeding in the use of computation resources.

In [37] authors use Spark as an in-memory open source cluster-computing framework. Spark Streaming is used to process in batches the data points coming from the stream. Batches are fixed at user specified time intervals, and parallelisation with Apache Spark is performed over the data points in the batch. For each data point, the proposed algorithm finds the nearest micro-cluster. The data points that could not be absorbed by the existing micro-clusters are processed separately and sequentially, by searching a micro-cluster to remove or two micro-clusters to merge. In addition, authors in [37] propose the usage of the Canopy algorithm, an unsupervised learning preprocessing algorithm, which can improve the preprocessing of large data in order to optimise the offline phase of CluStream. With respect to P-IDEAaS, the approach in [37] considers only the buffer parallelisation, and does not mention any parallelisation over facets nor over micro-clusters, neither for finding the micro-cluster to remove or micro-clusters to merge. Indeed these parts of the data stream clustering are the most challenging for parallelisation, since they determine the creation and removal of micro-clusters, thus requiring a sequential processing of data points. Nonetheless, introducing some parallelisation strategies also in these parts, and properly managing outliers, may improve the overall algorithm performance. Similarly, the approach in [36] is based on Apache Spark to process batches of incoming data points in parallel. In each batch, the computation of distance from a new data point and centroids of existing micro-clusters is performed in parallel as well. The issue of handling outliers is considered here, and also in this case outliers that can not be absorbed in existing micro-clusters are processed separately and sequentially. The work presented in [35] uses Apache Storm as a distributed stream processing framework, coping with policies related to memory management (the so-called shared-nothing versus shared memory issue). Authors propose two different solutions to parallelize the online phase of Clustream. The first one is run on a shared-memory architecture using a common store, and the other one is run on a non-shared-memory architecture in a decentralised way. As new data points arrive, they are assigned to a processing unit, that is in charge of generating the new set of clusters. Data points are sent to a local processing unit in a load-balanced manner, and resulting micro-clusters are sent back to a remote processing unit in a round-robin fashion. Both algorithms need a synchronisation phase in order to avoid to loose the clustering accuracy: in the shared-memory architecture, local processing units (called *allocators*) communicate directly with the global processing unit (called

aggregator) for synchronisation; data points that do not belong to any existing micro-cluster trigger the update of the global set of clusters; in the non-shared-memory solution, synchronisation is performed between local processing units according to a gossip-based algorithm in a P2P environment. Parallelisation presented in [35] is performed on the buffer level only. Moreover, also in this case, authors do not consider the opportunity of tuning different parallelisation strategies. Finally, the SAMOA library [39] is a distributed computing implementation of CluStream. It is not implemented in Spark, but rather in a Distributed Stream Processing Engine which adapts the Map-Reduce paradigm to parallel stream processing, and it does not include the offline phase.

	Platform	Dataset type	Base algorithm	Outlier inspection
PaStream [32]	GPU (vertical scaling)	synthetic, real	CluStream	(∼)
GBIRCH [33]	GPU (vertical scaling)	synthetic, real	BIRCH	No
CClustream [37]	Apache Spark (horizontal scaling)	synthetic	CluStream	No
Spark-Clustream [36]	Apache Spark (horizontal scaling)	synthetic	CluStream	No
Karunaratne et al. [35]	Apache Storm (horizontal scaling)	synthetic, real	CluStream	No
P-IDEAaS	Apache Spark (horizontal scaling)	real	CluStream	Yes

	Parallelisation focus	Parallelisation levels	Tunable parallelisation	Experimental setup
PaStream [32]	online - offline	buffer, micro-cluster relevance stamp, distance from micro-cluster centroids, offline phase	No	Intel-i7 CPU 3.40 GHz (4 cores, 8 threads), NVIDIA GeForce GTX 750 GPU
GBIRCH [33]	online	buffering only	No	Intel-i5 3.20 GHz, 8GB RAM, NVIDIA Tesla K20 GPU
CClustream [37]	online - offline	buffering only	No	Intel-i5 3.40 GHz, 8GB RAM (3 machines)
Spark-Clustream [36]	online - offline	buffering, distance from micro-cluster centroids	No	- (up to 40 processor)
Karunaratne et al. [35]	online	buffering only	No	8-node cluster, 4GB RAM
P-IDEAaS	online	buffering, micro-cluster relevance stamp, distance from micro-cluster centroids, distance between micro-clusters	Yes	Intel-i5 2.60 GHz, 1 master - 3 slaves

Table 2.2: Overview of approaches on parallel clustering of Big Data streams.

Contribution of the P-IDEAaS approach. Table 2.2 reports a comparative overview of approaches on parallel clustering of data streams. With respect to existing approaches, parallel implementation of the clustering algorithm within P-IDEAaS is structured into different levels of parallelisation (at increasing complexity), that can be tuned through the multi-dimensional model, in combination with the data relevance evaluation, in order to adapt the use of computation resources to the different characteristics of the incoming data stream. It is focused on the online phase, as it determines the stream rates that the algorithm will be able to handle. Among the approaches in literature that addressed the parallelisation of the two-phases clustering of massive and evolving data streams, the approaches proposed in [32, 36] are the only ones that go beyond the partition of the initial data stream through the buffering. Nevertheless, the P-IDEAaS approach is the only one that enables to tune parallelisation levels with respect to the buffer size, the dynamics and complexity of the stream (i.e., the number of features). Considering the commodity hardware capabilities of horizontal scaling platforms, experiments will demonstrate that the approach enables to increase performances also reducing computation costs, compared to the adoption stream. Moreover, although vertical scaling platform might ignore this aspect since they enable a performance increment with respect to the traditional approaches, as explained in [25], horizontal scaling platforms can be potentially extended without limits with the addition of new computation nodes, thus facing Big Data volume with commodity hardware. For what concerns the inspection of outliers, almost all the approaches process sequentially those data points that cannot be absorbed by existing clusters. The PaStream approach [32] assumes the same perspective as well and does not process outliers in parallel, although it proposes the use of a *Distance Matrix* to efficiently store Euclidean distance values between micro-clusters. In P-IDEAaS approach the strategy of postponing the outlier management (also experimentally evaluating the effects of this action on the clustering results) will be proposed in order to reduce as much as possible the creation, merging and/or removal of micro-clusters in the procedure.

2.3 Anomaly Detection

The IDEAaS approach is described in this thesis can be classified among approaches that have been proposed to address anomaly detection in presence of big data streams

(please refer to [40] for a comprehensive survey). These approaches differ from those based on static data, since all the observations are not available at once and measures are collected and processed incrementally. Moreover, the IDEAaS approach also differs from solutions for anomaly detection in presence of evolving graphs [41, 42], that are characterised by causal/non-causal relationships between measurements.

Among the approaches for anomaly detection on evolving data, the authors in [40] focused on unsupervised proposals, since supervised and semi-supervised scenarios are rare to happen in real-world applications, due to the lack of label information regarding the anomalies that could be detected in collected observations. Unsupervised approaches can be in turn classified into statistical-based, nearest neighbours-based and clustering-based. Statistical-based approaches usually require a priori knowledge about the underlying distribution of the measures, that is almost always unavailable when data is collected incrementally. In [43] an approach based on in-memory big data processing is described. A preparation phase is used to generate a model for the “usual state" of the system, by applying machine learning (pre-training) on stored data. An operation phase compares real-time incoming data with the “usual state" to identify anomalies. Similarly, in [44] machine learning is used to train data collected during regular execution of the manufacturing process in order to learn a probabilistic “normal model". Authors in [45] applies Hierarchical Temporal Memory (HTM) to anomaly detection, by performing two post-processing steps over the output of HTM system: (i) computing the prediction error; (ii) computing the anomaly likelihood.

Nearest neighbours-based approaches rely on the assumption that a measure can be considered as an anomaly if its distance from a significant portion of other measures is greater than a given threshold [46, 47]. In clustering-based approaches, anomalies are discovered either: (a) since they are assumed to fall into clusters with small number of data points or low density; (b) based on their distance from nearest clusters centroids. The approach in [22] operates in two steps: (i) learning of the normal behaviour of the system (based on past data), using a clustering technique (K-means algorithm); (ii) detecting at real-time an anomalous behaviour when new data does not belong to previously detected clusters. The approach in [48] builds a cluster model using Gaussian clustering, that is updated as incoming data arrives. Clustering is performed over a time window. As a new data arrives, the algorithm tries to assign it to an existing cluster. If this is not possible, the evaluation on new

data is suspended. When the time window expires, a batch clustering algorithm (e.g., DBScan) is performed, in order to check if suspended data is an anomaly or can be recognised as a new cluster.

Although our approach is cluster-based, it is focused on the evolution of summarised data over time in order to detect anomalies. Indeed, it relies on summarisation techniques as a basis on which to apply relevance evaluation. Moreover, exploration is performed over the multi-dimensional model. This distinguishes the IDEAaS approach from the approaches described in [40] and from traditional Complex Event Processing (CEP) approaches, that are mainly based on pre-defined queries and event detection rules.

2.4 Context-based resilience

Monitoring of industrial plants for anomaly detection purposes is aimed at ensuring timely recovery actions. In [49] has been integrated an anomaly detection approach with a CMMS (Computerised Maintenance Management System) to provide time-limited maintenance interventions. However, a recent ever growing interest has been devoted to resilience challenges, often related with the notion of self-adaptation, as witnessed by an increasing number of surveys on this topic [50, 51, 52]. This will be the focus of this thesis. Time-limited solutions can be considered if the identification of recovery actions fails. CPS resilience approaches are mainly focused on security issues at runtime on the communication layer or to implement resource balancing strategies between the edge and cloud computing layers. In [53] 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. This kind of approaches is out of the scope of our thesis. However, resilience/self-adaptation specifically designed for CPPS has been addressed in few works [54, 55], where ad-hoc solutions are provided, focusing on single work centers or on the production line, without considering the effects of resilience across connected components. In [56] 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.

This thesis approach contributes to the state of the art by introducing a context

model, apt to relate recovery services with work centers organised in the fully connected hierarchy of smart components (from connected devices up to the whole production line at shop floor level), according to the IEC62264/IEC61512 standards of the RAMI 4.0 reference architectural model [8]. Work centers 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 centers and, on the other hand, to personalise the visualisation of recovery actions nearby the involved work centers, supporting the operativity of workers who supervise the production line.

The adoption of context-awareness to implement resilient CPS has been investigated in the Cyber Physical Systems (CAR) Project [57]. 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 [58] 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. With respect to them, here is added the context model and has been proposed a set of context-driven phases to support the human operator in the identification of critical conditions and in the confirmation of recovery services.

Chapter 3

A Relevance-based approach for Big Data Exploration

In this chapter the core components of the IDEAaS approach will be presented with more details, namely: the multi-dimensional model to partition incoming data streams, the incremental clustering algorithm and relevance-based data exploration.

3.1 Multi-dimensional model definition

Data points in a stream can be explored and analysed according to different perspectives, that may change dynamically during data stream processing.

Definition 1 (Data Stream) *A data stream DS is a sequence of data generated over time described as* $DS = \{\overline{x}(t_0), \overline{x}(t_1), \ldots, \overline{x}(t_n)\}$, *where* $\overline{x}(t_n)$ *is a data collected at a certain time* t_n *and n can potentially be* ∞.

Within IDEAaS the considered data stream are composed of only numeric data, called data points which are defined as follows:

Definition 2 (Data points) *Data points are modelled as d-dimensional vectors* $\overline{x} = \langle x_1, x_2, \ldots, x_d \rangle$, *where each* x_i *is a measure of a feature* f_i. *Features correspond to the measurable quantities to be monitored over the observed system.*

Indeed, features are formally defined as follows.

Definition 3 (Feature) *A feature* f_i *is a measurable quantity described as* $\langle n_{f_i}, u_{f_i} \rangle$, *where* n_{f_i} *is the feature name,* u_{f_i} *represents the unit of measure. Let's denote with* $F = \{f_1, f_2, \ldots, f_d\}$ *the overall set of measurable features in the monitored system. A data point* $\overline{x}(t) = \langle x_1(t), x_2(t), \ldots x_d(t) \rangle$ *is defined as the tuple containing the measurements of features in F, collected jointly at timestamp t, expressed in terms of their units of measure* $u_{f_1}, u_{f_2}, \ldots u_{f_d}$.

A *feature space* is defined as a set of features that are measured together to observe a physical phenomenon of interest. Spindle rolling friction torque increase is an example of feature space. In fact, it can be monitored through the spindle power absorption feature jointly with the spindle rpm feature. Specifically, if the spindle rpm decreases as long as the power absorption increases, this might be due to the spindle rolling friction torque increase. Different feature spaces can be defined over the whole set F of features. Feature spaces are designed to be monitored separately according to the different perspectives that identifies portions of the data stream on which exploration can be differently focused. For example, in the smart factory anomaly detection case study, data points can be partitioned with respect to the different tools that are used during the manufacturing process on the monitored Cyber Physical System. Within the IDEAaS approach, different exploration perspectives have been modelled using a Multi-Dimensional Model (MDM).

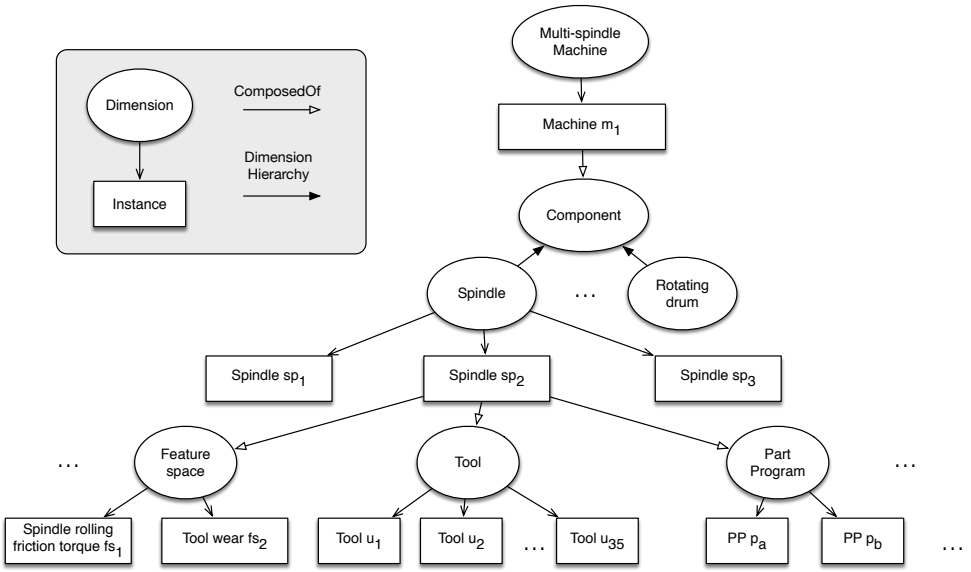


Figure 3.1: The tree structure used to represent the multi-dimensional model (an instantiation for the anomaly detection in smart factory case study.).

The MDM formalization is inspired by the literature on multi-dimensional data analysis [59] and involves dimensions and hierarchies over dimensions, according to the following definition.

Definition 4 (Multi-Dimensional Model) *A Multi-Dimensional Model over a data stream is denoted as a triple* $MDM = \langle D, H, DS \rangle$*, where:*

- $D = \{d_1, \ldots d_p\}$ *is a finite set of analysis dimensions on categorical domains* $Dom(d_k)$, $\forall k = 1 \ldots p$;

- $H = \{h_1, \dots h_m\}$ *is a finite set of hierarchies; each hierarchy is described with (i) a subset* $Dim(h_j) \subseteq D \; \forall j = 1 \dots m$ *of dimensions, such that* $Dim(h_j)$ *defines a partition of D; (ii) a total order* $\succeq_{h_j}$ *of* $Dim(h_j)$*; let* $Dom(H) = Dom(h_1) \times \dots \times Dom(h_m)$*;*
- *DS is a (potentially infinite) set of data points within the data stream.*

Figure 3.1 reports an instantiation of the MDM in the anomaly detection for smart factory case study. In the example depicted in Figure 3.1, analysis dimensions are the `Machine`, the `Spindle`, the `Tool` and the `Part Program`[1].

Combinations of instances for analysis dimensions can be used to aggregate data points according to different exploration perspectives. These combinations have been denoted as *exploration facets*, defined as follows.

Definition 5 (Exploration facet) *Given a* $MDM = \langle D, H, DS \rangle$*, a facet schema is denoted as* $\phi \in Dom(H)$*. An exploration facet* $\varphi_i \in Dom(\phi)$ *is a combination of dimension instances, one for each* $h_j \in H$*, used to partition the whole set of d-dimensional data points. Let's denote with* Φ *the set of exploration facets.*

With reference to the example reported, in Figure 3.1 the spindle sp_2, the tool u_3 and the part program pp_1 form an exploration facet used to filter the subset of data points in the highlighted data cube. Partitions created by means of exploration facets can be separately processed through the incremental clustering algorithm to generate micro-clusters.

In the following, details on how data is summarised and organised in the multi-dimensional model will be provided.

3.2 Multi-dimensional and incremental clustering

The algorithm described in this section has been published in [60], where the notion of data summarisation was introduced to aggregate in the feature space a set of measures that are closely related each other, by applying a notion of distance between measures. The micro-clusters generation procedure is organised over two steps: (i) the creation of new micro-clusters and (ii) the update of existing ones. Algorithm (1) summarises the micro-clusters generation procedure and derives from the CluStream approach [6].

[1] Part Programs are sequences of instructions, which define the actions to be executed by a numerical control machine.

A micro-cluster is represented as a hyper-sphere, representing a set of d-dimensional data points, where d is the number of measured quantities. Using a temporal extension of the concept of *cluster feature vector* (CF) [61], which also takes into account the timestamps of incoming data points, the i-th micro-cluster μc_i, containing n_i data points $\overline{X}_i = \overline{x}_{i_1}, \overline{x}_{i_2}, \ldots, \overline{x}_{i_{n_i}}$ with timestamps $\overline{T}_i = t_{i_1}, t_{i_2}, \ldots, t_{i_{n_i}}$, can be represented by a tuple defined as follows:

$$\mu c_i = \langle \overline{CF1_i^x}, \overline{CF2_i^x}, CF1_i^t, CF2_i^t, n_i \rangle \tag{3.1}$$

where: (i) each $\overline{x}_{i_k}$ is a d-dimensional data point, represented as a vector; (ii) $\overline{CF1_i^x}$ is a d-dimensional vector whose elements represent the linear sum of data points in μc_i (one for each dimension); (iii) $\overline{CF2_i^x}$ is a d-dimensional vector whose elements represent the quadratic sum of data points in μc_i (one for each dimension); (iv) $CF1_i^t$ is a scalar value representing the linear sum of timestamps in μc_i; (v) $CF2_i^t$ is a scalar value representing the quadratic sum of timestamps in μc_i; (vi) n_i is the number of data points included into μc_i. Therefore, a micro-cluster is represented through $(2d + 3)$ values instead of $(n_i * d)$, significantly reducing the memory requirements. From the elements of the tuple (3.1) above, the *centroid* $\overline{X0}_i$ and the *radius* R_i of the micro-cluster μc_i can be obtained.

3.2.1 Micro-clusters generation

The micro-clusters generation procedure reported in Algorithm (1) is iterative. At each iteration, the algorithm is logically divided in two parts: (i) firstly, each data point creates a new micro-cluster, until the maximum number of available micro-clusters is reached (lines 3-6); (ii) existing micro-clusters are update considering the new incoming data points (line 7-18).

Concerning the first two parts, instead of processing one data point at a time, data points $\overline{X}$ are collected in a buffer that spans a temporal lapse equal to $\varphi_i.\Delta t$, where $\phi_i \in \Phi$ denotes the current exploration facet in which the algorithm is being executed. At each iteration it updates the existing set of micro-clusters μC^{φ_i} for the considered exploration facet φ_i. As new records $\overline{X}$ arrive, if the maximum number of allowed micro-clusters $MAX_{\mu C^{\varphi_i}}$ has not been reached yet, a new micro-cluster is created (CREATENEWMICROCLUSTER procedure), containing only the new record (lines 4-6). The value $MAX_{\mu C^{\varphi_i}}$ is determined by the amount of main memory

Algorithm 1: Micro-clusters update function

Input: set μC^{φ_i} of micro-clusters given an exploration facet φ_i, set $\overline{X}$ of new data points
Output: updated set $\mu C^{\varphi_i}_{new}$ of micro-clusters

```
1  Function updateMicroClusters(μC^{φ_i}, X̄):
2      foreach data point x̄_i ∈ X̄ do
3          if SIZEOF(μC^{φ_i}) < MAX_{μC^{φ_i}} then
4              μc_new ← CREATENEWMICROCLUSTER(x̄_i)
5              μC^{φ_i}_new ← μC^{φ_i} ∪ {μc_new}
6          else
7              μc_near ← FINDCLOSESTMICROCLUSTER(μC^{φ_i}, x̄_i)
8              if μc_near == null then
9                  μc_remove ← GETMICROCLUSTERTOREMOVE(μC^{φ_i})
10                 if μc_remove ≠ null then
11                     μC^{φ_i}_new ← μC^{φ_i} \ {μc_remove}
12                 else
13                     (μc_a, μc_b) ← GETMICROCLUSTERSTOMERGE(μC^{φ_i})
14                     μC^{φ_i}_new ← MERGE(μc_a, μc_b)
15                 end if
16             end if
17             μc_new ← CREATENEWMICROCLUSTER(x̄_i)
18             μC^{φ_i}_new ← μC^{φ_i} ∪ {μc_new}
19         end if
20     end foreach
21     return μC^{φ_i}_new
22 End Function
```

available in order to store the micro-clusters. The parameter $MAX_{\mu C^{\varphi_i}}$ has been introduced to prevent memory overloading. Otherwise, if the maximum number of micro-clusters has been reached, the algorithm searches for an existing micro-cluster in which the new record $\overline{x}_i$ can be assigned, through the FINDCLOSESTMICROCLUSTER procedure (line 7). This procedure identifies a cluster μc_{near} that represents the cluster that is closest to the new record $\underline{x}_i$. The implementation of this procedure is compliant with the one detailed in [6] and is based on the relative distance of $\overline{x}_i$ from the centroid of μc_{near} compared to the distance from the centroids of other micro-clusters. If it is not possible to identify μc_{near} (line 8), the data point is interpreted as new emerging behaviour in the monitored system and should generate a new micro-cluster.

Since the maximum number of micro-clusters has been reached, before including the new micro-cluster into the output set $\mu C^{\varphi_i}_{new}$, the new micro-cluster must be substituted to an existing one. The GETMICROCLUSTERTOREMOVE procedure is applied to identify the micro-clusters to remove (lines 9-12). In CluStream [6],

this procedure is based on the micro-cluster age. If a micro-cluster to remove is not identified, two other micro-clusters must be merged, using GETMICROCLUSTERSTOMERGE procedure (lines 13-15).

Once a micro-cluster has been removed or two micro-clusters have been merged, a new micro cluster μc_{new} is generated from the data point $\underline{x}_i$, through the CREATENEWMICROCLUSTER procedure (line 17).

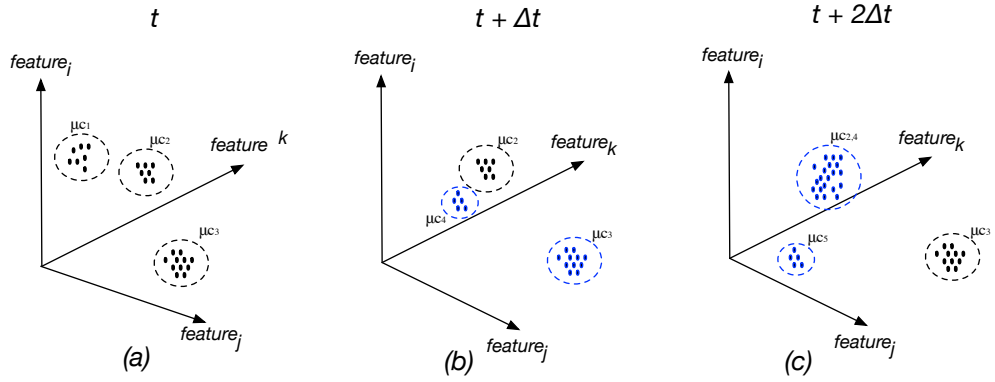


Figure 3.2: Example of an evolution of micro-clusters over time considering a three dimensions feature space. Analysis dimensions are fixed and not shown here.

Figure 3.2 illustrates three different results of the micro-clusters generation procedure for the same three-dimensional feature space; coloured micro-clusters are the micro-clusters that have changed over time. Indeed, from Figure 3.2(a) to Figure 3.2(b) the micro-cluster μc_1 has been deleted to create space for the new micro-cluster μc_4; and the micro-cluster μc_3 has been re-positioned. On the next step, from Figure 3.2(b) to Figure 3.2(c) the merging of micro-clusters μc_2 and μc_4 occurs in order to create space for the new emerging micro-cluster μc_5.

With respect to the CluStream approach [6], the micro-cluster generation procedure has been modified to generate a new set of micro-clusters starting from a set of records collected in a time interval equal to Δt. In fact, considering the volume and the velocity of the collected records, the application of the micro-cluster generation procedure on each single record may be not efficient, as demonstrated in the experimental evaluation. Hence, the clustering algorithm at a given time t produces a new set of micro-clusters $\mu C^{\varphi_i}_{new}$ starting from data points collected from time $t - \Delta t$ to t and built on the top of the previous set of micro-clusters, for a given feature space. Choosing a proper Δt interval is critical to guarantee that the time required to summarise new data is compliant with the data acquisition rate and to reduce storage

costs. Indeed, lower Δt values require more time to prepare data for summarisation and for input/output operations on the IDEAaS Data Storage. On the other hand, if Δt is set to larger values, the system is less prompt to detect changes in the observed data. Section 4.2 will illustrate experimental results obtained by analysing the impact of variations in the value of Δt.

With respect to [6], the following innovations have been additionally introduced in the IDEAaS approach: (i) the micro-clusters update criterion has been modified, considering both the age and density of generated micro-clusters; (ii) micro-clusters have been organised within the multi-dimensional model, and the notion of *snapshot* has been introduced, to enable successive exploration of different portions of incoming data stream.

3.2.2 Micro-clusters update strategy in IDEAaS

The solution proposed in [6] identifies micro-clusters to remove by relying on their *age*. Given a threshold τ, if the micro-cluster last update occurs before $t - \tau$, then the micro-cluster is considered old and it is candidate to be discarded. Further considerations on the choice of threshold τ are reported in the experimental evaluation in Section 4.2. If only the age of micro-cluster is considered, persistent behaviours of the monitored system may be neglected, although they are a valuable representation of consolidated working status. Here, two different criteria have been combined, in order of priority: (i) the age and (ii) the density of the micro-cluster, defined as follows.

Definition 6 (Density of a micro-cluster) *The density of a micro-cluster μc_j, denoted with $den(\mu c_j)$, is defined as:*

$$den(\mu c_j) = \frac{N_j}{(R_j)^n} \tag{3.2}$$

where (i) N_j is the number of records belonging to the micro-cluster, (ii) $(R_j)^n$ is the micro-cluster bounding n-dimensional cube (i.e, an envelope of the area/hyper-volume occupied by μc_j) and (iii) n is the number of features in the record, composing the considered feature space. The value $den(\mu c_j)$ is updated whenever μc_j is modified.

If a micro-cluster is classified as old (according to the threshold τ) and its density is the highest one among all old micro-clusters, then it is candidate to be removed. The rationale is that, if dense micro-clusters have not been updated for a while, then

they represent a consolidated working behaviour that is no longer present in the monitored system. These micro-clusters should be removed because their density may attract new records distorting the micro-clusters update results. Experiments in Section 4.2 confirmed this intuition. If micro-clusters candidate to be removed are not found, because all the existing micro-clusters have been updated after $t - \tau$ and cannot be classified as old, then the whole set of micro-clusters is looked up to find the two closest ones to be merged, according to the original Algorithm (1).

3.2.3 Multi-dimensional organisation of micro-clusters in IDEAaS

Distinguishing micro-clusters across different facets might be useful to foster data exploration, as suggested by the identification of tool wear in the anomaly detection case study (see Section 1.3.1). Therefore, the set of micro-clusters generated every Δt seconds is stored as a *snapshot* and labelled with analysis dimensions according to the multi-dimensional model. For example, considering the CNC machine in the case study, different snapshots can be built depending on the feature space that is being considered, the tool that is being used, the part program that is being executed. A snapshot is formally defined as follows.

Definition 7 (Snapshot) *A snapshot $SN(t)$, stored at time t, is defined as the following tuple:*

$$SN(t) = \langle \mu C(t), \rho, fs_j, \varphi_i \rangle \tag{3.3}$$

where: (i) $\mu C(t)$ is a set of micro-clusters generated at time t, (ii) $\rho : \mu C(t) \rightarrow 2^{\mu C(t-\Delta t)}$ is a mapping function that relates a micro-cluster in $\mu C(t)$ to zero, one or more micro-clusters in the set $\mu C(t - \Delta t)$ stored in the previous snapshot $SN(t - \Delta t)$, (iii) fs_j is the monitored feature space and (iv) φ_i is the exploration facet.

The co-domain of the mapping function ρ is the power set $2^{\mu C(t-\Delta t)}$, because a micro-cluster $\mu c_i \in \mu C(t)$ can be mapped to: (i) a single micro-cluster in $\mu C(t - \Delta t)$ if it is maintained across the two snapshots, (ii) a set of micro-clusters in $\mu C(t - \Delta t)$ as a result of a merge operation or (iii) the empty set if the micro-cluster μc_i has just been created in $\mu C(t)$ and therefore was not previously present in $SN(t - \Delta t)$.
As an example, all the snapshots resulting from the application of data summarisation techniques for machine m_1 (spindle sp_2), while using tool u_2 and during the execution of the part program p_a, to monitor the tool wear feature space fs_2, are

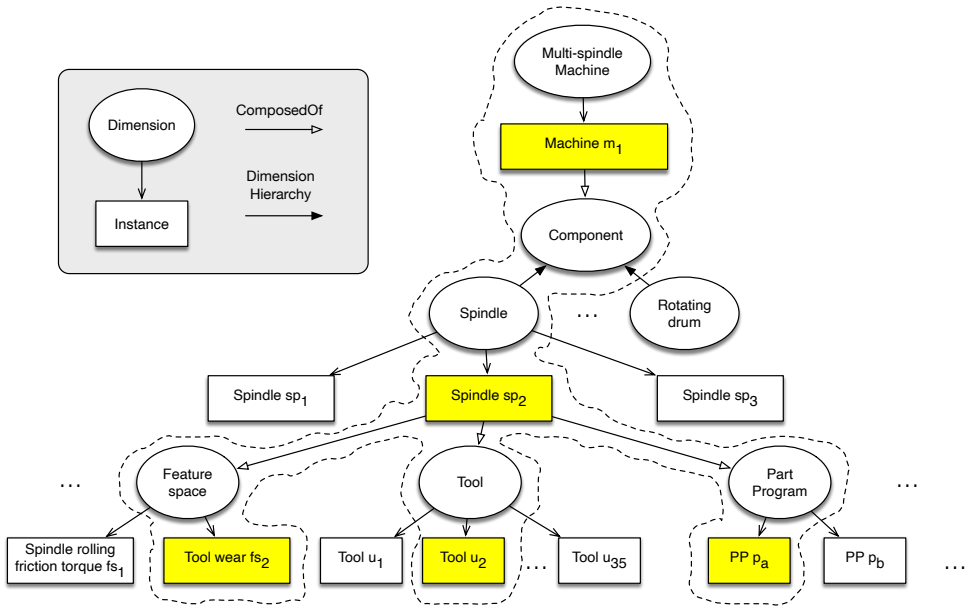


Figure 3.3: Annotation of micro-clusters snapshots using the multi-dimensional model.

annotated with the subset of the tree nodes highlighted by the dashed line in Figure 3.3. In the next section, it will be shown how exploration of the model enables to identify the snapshots of interest.

3.3 Relevance-based data exploration

In this section, it is illustrated how IDEAaS can guide the expert throughout the exploration of relevant data, following three steps: (i) the identification of relevant data through the application of novel data relevance evaluation techniques; (ii) starting from the relevant data previously identified, the exploration over the Multi-Dimensional Model; (iii) the selection of relevant data and the visualisation of its evolution over time.

3.3.1 Identification of relevant data

According to the definition given in [1], data relevance can be defined as the distance from an *expected status*. Data relevance evaluation techniques are aimed to support the identification of relevant (and potentially critical) data according to the evolution over time of identified micro-clusters. Given an exploration facet $\varphi \in \Phi$, micro-clusters evolution is detected by comparing the current set of micro-clusters

in φ (denoted with μC^{φ}_{curr}) against the set of micro-clusters generated when the monitored system is operating in normal conditions in the same facet φ (denoted with $\hat{\mu C}^{\varphi}$). Deviation from the normal working conditions means that the monitored system behaves abnormally and alerts about the identified conditions must be raised. The main concern here is to define the expected status and how to compute such a distance. In the IDEAaS approach, the expected status corresponds to the snapshot saved during the normal working behaviour of the monitored system (denoted as *reference snapshot*) and data relevance techniques are based on the computation of the distance between the set of micro-clusters in the current snapshot and the set of micro-clusters in the reference snapshot. The devised approach aims at identifying relevant snapshots through the application of relevance evaluation techniques, and then at retrieving relevant micro-clusters among the ones of the relevant snapshots. Data within such micro-clusters is proposed to the expert as relevant.

Distance between sets of micro-clusters. The proposed relevance evaluation techniques rely on the concept of *distance* between the two sets of micro-clusters $\mu C^{\varphi}_{curr} = \{\mu c_1, \mu c_2, \ldots, \mu c_n\}$ and $\hat{\mu C}^{\varphi} = \{\hat{\mu c}_1, \hat{\mu c}_2, \ldots, \hat{\mu c}_m\}$, where n and m represent the number of micro-clusters in μC^{φ}_{curr} and $\hat{\mu C}^{\varphi}$, respectively, and n and m do not necessarily coincide. The distance is computed as:

$$\Delta(\mu C^{\varphi}_{curr}, \hat{\mu C}^{\varphi}) = \frac{\sum_{\hat{\mu c}_i \in \hat{\mu C}^{\varphi}} D(\hat{\mu c}_i, \mu C^{\varphi}_{curr}) + \sum_{\mu c_j \in \mu C^{\varphi}_{curr}} D(\mu c_j, \hat{\mu C}^{\varphi})}{m+n} \tag{3.4}$$

where $D(\hat{\mu c}_i, \mu C^{\varphi}_{curr}) = min_{j=1,\ldots,n} d(\hat{\mu c}_i, \mu c_j)$ is the minimum distance between $\hat{\mu c}_i \in \hat{\mu C}^{\varphi}$ and micro-clusters in μC^{φ}_{curr}. Similarly, the distance $D(\mu c_j, \hat{\mu C}^{\varphi}) = min_{i=1,\ldots,m}$ $d(\mu c_j, \hat{\mu c}_i)$ is the minimum distance between $\mu c_j \in \mu C^{\varphi}_{curr}$ and micro-clusters in $\hat{\mu C}^{\varphi}$.

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 $d_{\overline{X0}}(\mu c_a, \mu c_b)$, 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:

$$d(\mu c_a, \mu c_b) = \alpha \cdot d_{\overline{X0}}(\mu c_a, \mu c_b) + \beta \cdot d_R(\mu c_a, \mu c_b) \tag{3.5}$$

where α, $\beta \in [0,1]$ are weights such that $\alpha + \beta = 1$, used to balance the impact of

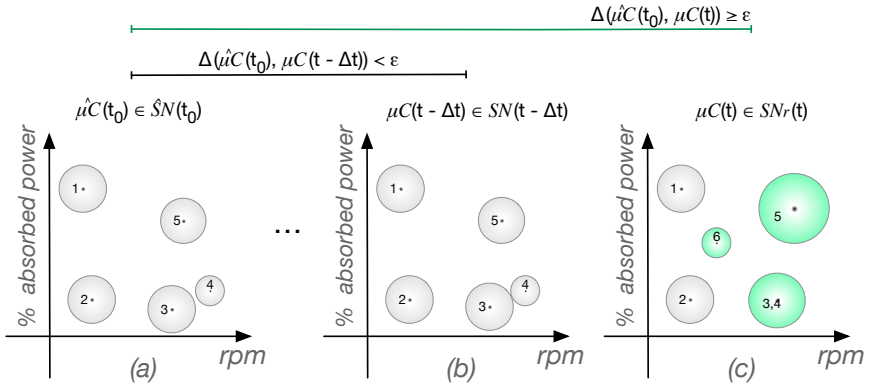


Figure 3.4: Evolution of micro-clusters over time. Feature space is composed of the spindle rpm and the percentage of absorbed power. Analysis dimensions are fixed and not shown here.

terms in Equation (3.5). In the experiments described in [60] the two terms of Equation (3.5) are equally weighted, that is, $\alpha = \beta = \frac{1}{2}$.

The distance $d_{\overline{X0}}(\mu c_a, \mu c_b)$ is computed by applying the Euclidean distance between micro-clusters centroids, according to the following formula:

$$d_{\overline{X0}}(\mu c_a, \mu c_b) = \sqrt{(\overline{X0}_a - \overline{X0}_b)^2} \tag{3.6}$$

where $\overline{X0}_a$ and $\overline{X0}_b$ are centroids of μc_a and μc_b, respectively.

The distance $d_R(\mu c_a, \mu c_b)$ is obtained by computing the difference between micro-clusters radii, that is:

$$d_R(\mu c_a, \mu c_b) = R_b - R_a \tag{3.7}$$

where R_a (resp., R_b) is the radius of μc_a (resp., μc_b).
If $d_R(\mu c_a, \mu c_b) > 0$ the micro-cluster μc_b has expanded with respect to micro-cluster μc_a, otherwise, if $d_R(\mu c_a, \mu c_b) < 0$ a micro-cluster contraction has been detected.

To better understand the rationale behind data relevance evaluation, Figure 3.4 illustrates three different micro-clusters sets related to the same bi-dimensional feature space; coloured micro-clusters are deemed as relevant due to centroid re-positioning from Figure 3.4(a) to Figure 3.4(c) (see micro-cluster μc_5), micro-cluster expansion (see micro-cluster μc_5), contraction, creation (see micro-cluster μc_6) or merging (i.e., variation of the number of micro-clusters, that has an impact on the

denominator of Equation (3.4), see micro-clusters μc_3 and μc_4.

Identification of relevant snapshots. Let $\hat{SN}(t_0) = \langle \hat{\mu C}(t_0), \rho, fs_j, \varphi_i \rangle$ be the reference snapshot, where $\hat{\mu C}(t_0) = \{\hat{\mu c}_1, \hat{\mu c}_2 \dots \hat{\mu c}_k\}$, fs_j is the observed feature space and $d_1, \dots d_p$ are the values of analysis dimensions in facet φ_i. $\hat{SN}(t_0)$ is tagged by the expert while observing the monitored system when operates normally. As explained above, data relevance at time t is based on the computation of *distance* between the current set of micro-clusters $\mu C(t) = \{\mu c_1, \mu c_2, ..., \mu c_n\}$ and $\hat{\mu C}(t_0) = \{\hat{\mu c}_1, \hat{\mu c}_2, ..., \hat{\mu c}_k\}$, where such *distance* has been denoted with $\Delta(\hat{\mu C}(t_0), \mu C(t))$. The value of this distance is used to identify relevant snapshots and, accordingly, relevant exploration facets. A relevant snapshot is formally defined as follows.

Definition 8 (Relevant snapshot) *Given a snapshot $SN(t) = \langle \mu C(t), \rho, fs_j, \varphi_i \rangle$ stored at time t and the reference snapshot $\hat{SN}(t_0) = \langle \hat{\mu C}(t_0), \rho, fs_j, \varphi_i \rangle$, $SN(t)$ is recognised as relevant if $\Delta(\hat{\mu C}(t_0), \mu C(t)) \geq \varepsilon$, where ε is a threshold that is set by the expert.*

Similarly, relevant exploration facet is defined as follows.

Definition 9 (Relevant exploration facet) *Given a set of micro-clusters μC^{φ}_{curr}, generated for an exploration facet $\varphi \in \Phi$, and given the set of micro-clusters $\hat{\mu C}^{\varphi}$ generated when the monitored system is operating in normal conditions for the same exploration facet φ, the set μC^{φ}_{curr} is marked as relevant if the condition $\Delta(\mu C^{\varphi}_{curr}, \hat{\mu C}^{\varphi}) > \epsilon$ holds, where ϵ is a threshold set by domain expert based on his/her knowledge about the monitored system. The facet φ is denoted as relevant.*

An example of the evolution of micro-clusters over time is shown in Figure 3.4. Figure 3.4(a) represents the set of micro-clusters that belong to the reference snapshot, in this case $\hat{\mu C}(t_0) = \{\hat{\mu c}_1, \hat{\mu c}_2, ..., \hat{\mu c}_5\}$. Snapshot $SN(t)$, as shown in Figure 3.4(c), is identified as relevant considering that $\Delta(\hat{\mu C}(t_0), \mu C(t)) \geq \varepsilon$. In fact, a new micro-cluster μc_6 has been created, micro-clusters μc_3 and μc_4 have been merged, micro-cluster μc_5 has been expanded and moved. In the figure, the feature space composed of the spindle rpm and the percentage of absorbed power is considered. On the other hand, the previous snapshot $SN(t - \Delta t)$, whose set of micro-clusters $\mu C(t - \Delta t)$ is represented in Figure 3.4(b), has not been labelled as relevant, given that $\Delta(\hat{\mu C}(t_0), \mu C(t - \Delta t)) < \varepsilon$ (i.e., no relevant changes in micro-clusters set have been detected).

Identification of relevant micro-clusters. Once a relevant snapshot $SN(t) = \langle \mu C(t), \rho, fs_j, \varphi_i \rangle$ at time t has been detected, the data relevance approach identifies which micro-clusters have changed with respect to the set of micro-clusters in the previous snapshot $SN(t - \Delta t) = \langle \mu C(t - \Delta t), \rho, fs_j, \varphi_i \rangle$. This information is retrieved through the application of the mapping function ρ, that will be exploited during data exploration as shown in the next section. A micro-cluster may change in different ways over time. Indeed, a micro-cluster may be:

- *created*, if $\mu c_i \in \mu C(t)$ in $SN(t)$, but $\mu c_i \notin \mu C(t - \Delta t)$ in the previous snapshot (e.g., micro-cluster μc_6 in Figure 3.4(c)); in this case $\rho(\mu c_6) = \emptyset$;
- *merged*, if $\mu c_i \in \mu C(t)$ in $SN(t)$ is the result of a merging operation between two micro-clusters $\mu c_a, \mu c_b \in \mu C(t - \Delta t)$ in the previous snapshot $SN(t - \Delta t)$ (e.g., micro-cluster $\mu c_{3,4}$ in Figure 3.4(c)); in this case $\rho(\mu c_{3,4}) = \{\mu c_3, \mu c_4\}$;
- *moved*, when a micro-cluster $\mu c_i \in \mu C(t)$ in $SN(t)$ moved from its position in the previous snapshot $SN(t - \Delta t)$ (e.g., micro-cluster μc_5 in Figure 3.4(c)); in order to verify if a micro-cluster moved over time, Equation (3.6) is exploited; in this case $\rho(\mu c_5) = \{\mu c_5\}$;
- *expanded/shrunk* when the micro-cluster $\mu c_i \in \mu C(t)$ in $SN(t)$ changed its size compared to the previous snapshot $SN(t - \Delta t)$ (e.g., micro-cluster μc_5 in Figure 3.4(c)); in order to verify if a micro-cluster expanded/shrunk over time, Equation (3.7) is exploited; in this case $\rho(\mu c_5) = \{\mu c_5\}$.

Therefore, in the example of Figure 3.4 the micro-clusters considered as relevant and proposed to the experts to start the exploration are $\overline{\mu C}(t) = \{\overline{\mu c}_{3,4}, \overline{\mu c}_5, \overline{\mu c}_6\}$.

3.3.2 Relevance-driven data exploration

Figure 3.5 sketches how the multi-dimensional model and relevance evaluation techniques help experts during relevant data exploration.

Preparing the exploration. The relevant snapshots identified in the previous step are associated to a feature space and instances of the analysis dimensions. Feature space and analysis dimensions for which relevant snapshots have been found are properly labelled within the multi-dimensional space. To start the exploration, the expert might specify a set d^r of desired values for the dimensions he/she is interested in,

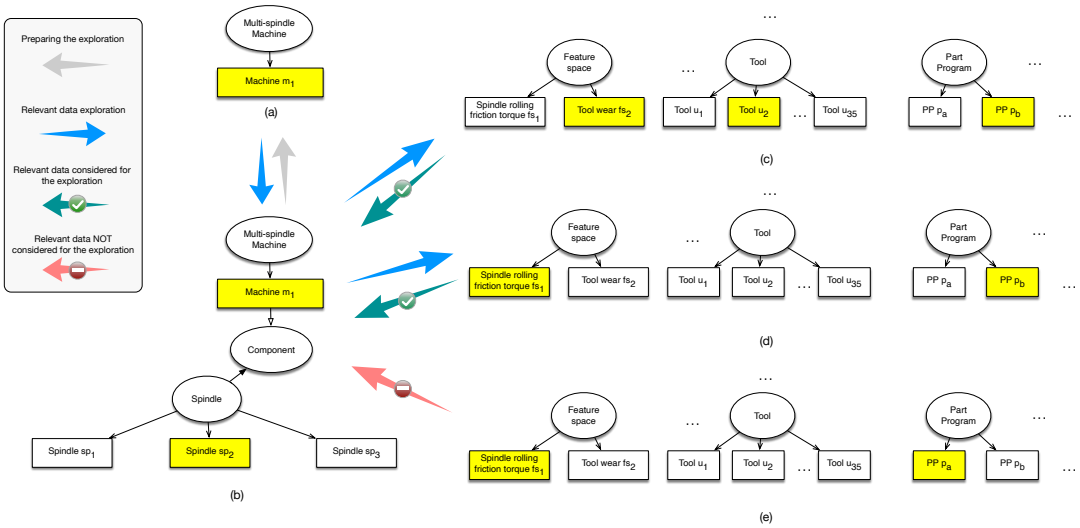


Figure 3.5: Data exploration supported by the multi-dimensional model and relevance evaluation techniques.

where $d^r = \{d_1^r, \ldots d_p^r\}$ and $d_i^r \in \mathcal{D}_i$. The expert might specify preferences on a subset of dimensions in $\mathcal{D}$ or he/she may not express any preference at all. Let *bounded* be the dimensions on which the expert expressed a preference and *unbounded* the other dimensions. IDEAaS filters out relevant snapshots that are associated with values for bounded dimensions that have not been specified in d^r. For example, referring to Figure 3.5, $d^r = \{\text{-}, \text{-}, \text{pp}_b\}$. In this case, the expert only expressed preferences on the part program, therefore IDEAaS does not consider relevant snapshots that are associated with the part program pp_a (Figure 3.5(e)), and selects relevant snapshots that have been found for the feature space "tool wear" (fs_2) while using the tool u_2 and executing the part program pp_b (Figure 3.5(c)) and those that have been found for the feature space "spindle rolling friction torque" (fs_1) while executing the part program pp_b (Figure 3.5(d)). It is assumed that the expert formulates d^r as an explicit, albeit vague, exploration request, and expects IDEAaS to suggest some promising data to explore. The expert may also leave d^r empty and IDEAaS will select only relevant snapshots at time t.

Selected relevant snapshots are used to prune the hierarchy of monitored system towards the upper levels: in Figure 3.5 relevant snapshots are associated to the spindle sp_2 and to the machine m_1, that includes the spindle among its components.

Exploring relevant data. The expert starts the exploration from the hierarchy of the monitored system, as shown in Figure 3.5(a) and Figure 3.5(b) for the motivating

Figure 3.6: Prototype data exploration GUI.

example. In the example, the machine m_1 is explored and, among the components of the machine, spindle sp_2 is highlighted to suggest to the expert that such spindle is associated with relevant snapshots, identified and selected in the previous steps. Relevant snapshots associated with dimensions values shown in Figure 3.5(c) and Figure 3.5(d) are proposed to the expert, properly ranked using relevance evaluation techniques. In particular, distance computed in Equation (3.4) is exploited as ranking criterion.

To let the expert explore selected relevant snapshots, a prototype exploratory GUI has been implemented. Figure 3.6 shows the GUI, where relevant micro-clusters have been plotted for the feature space composed of the percentage of absorbed power and the spindle rpm on the multi-spindle machine 101170 after selecting the part program 0171507160 in the anomaly detection case study. By clicking on the "Change Selected Dimensions" button the expert can change the dimensions to explore, according to the data exploration approach implemented on top of the Multi-Dimensional Model. On the GUI, physical limits of the two features in the considered feature space are plotted as well, to provide additional information to the expert

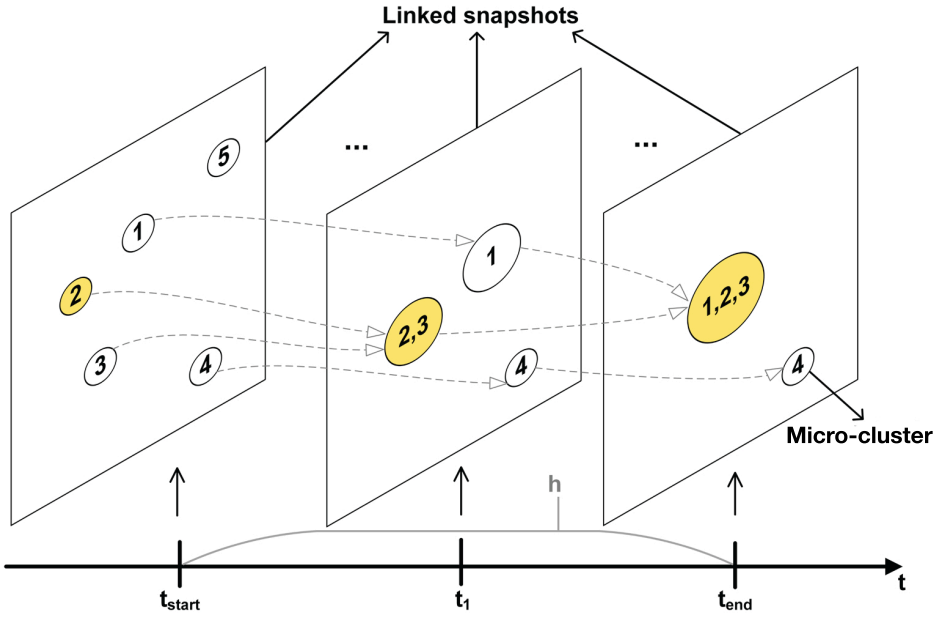


Figure 3.7: Evolution history for the relevant snapshots over the time window h with respect to the relevant micro-cluster with $id = 2$.

during data exploration. In the figure, empty micro-clusters are those that have not been identified as relevant. The coloured ones are associated with different colours and labels to distinguish among error micro-clusters (representing measures that exceeded feature allowed limits), warning micro-clusters (representing measures that are close to feature limits) and OK micro-clusters (relevant micro-clusters that aggregate measures that did not exceed any feature limit). Micro-clusters have been represented on the GUI as (hyper)spheres, as this is the most natural representation for objects endowed with a center and a radius.

3.3.3 Selection of relevant data

Once relevant snapshots have been selected through the exploration steps described in the previous section, their evolution over time up to the current time instant can be retrieved and visualised on the prototype GUI. Figure 3.7 presents the main idea behind the micro-clusters selection step. Starting from the current relevant snapshot (at time t_{end}), the sequence of previous relevant snapshots that led to the current one is retrieved. The temporal evolution of a relevant snapshot can be retrieved by exploiting backwards the linkage relationships between (relevant) snapshots, up to an initial time instant (at time t_{start}) when data relevance emerged. Links between

Algorithm 2: Relevant snapshots evolution over time

Input: current time instant t_{end}, a relevant snapshot $SN(t_{end})$
Output: the sequence of relevant snapshots Σ over the time window (t_{start}, t_{end})

```
1  Function GetRelevantSnapshotsSequence(t_end, SN(t_end)):
2      t_start ← t_end
3      Σ ← SN(t_end)
4      while t_end ≠ t_0 do
5          SN(t_prev) ← GetLinkedSnapshot(SN(t_end))
6          if IsRelevant(SN(t_prev)) then
7              Σ ← Σ ∪ SN(t_prev)
8              t_start ← t_prev
9          else
10             exit
11         end if
12     end
13     return Σ
14 End Function
```

snapshots are depicted as dashed lines in Figure 3.7, and the linkage relationship is formalised as follows.

Definition 10 (Linked snapshots) *Given a feature space fs_j and $\varphi_i = \{d_1, \ldots d_p\}$ values of analysis dimensions, snapshots $SN(t_1) = \langle \mu C(t_1), \rho_{SN(t_1)}, fs_j, \varphi_i \rangle$ and $SN(t_2) = \langle \mu C(t_2), \rho_{SN(t_2)}, fs_j, \varphi_i \rangle$, taken at time instants t_1 and t_2, respectively (with $t_2 = t_1 + \Delta t$), are linked with respect to a micro-cluster $\mu c_i \in \mu C(t_2)$ if $\rho_{SN(t_2)}(\mu c_i) \subseteq \mu C(t_1)$, that is, $\mu c_i \in SN(t_2)$ was already present in $SN(t_1)$, changed with respect to its counterpart in $SN(t_1)$ or derives from merging of other micro-clusters in $SN(t_1)$. Two snapshots $SN(t_1)$ and $SN(t_2)$ are denoted as linked if they are linked with respect to at least one micro-cluster.*

Definition 10 supports the tracing of a relevant micro-cluster across relevant linked snapshots. Algorithm (2) illustrates how to retrieve the set of such relevant snapshots, starting from the current time instant t_{end} and the snapshot $SN(t_{end})$; the outcome of the algorithm is the sequence of relevant snapshots (denoted with Σ) over the time interval h delimited by t_{start} and t_{end}.

Starting from t_{end} and the relevant snapshot $SN(t_{end})$, the procedure retrieves the previous snapshot $SN(t_{prev})$ linked to $SN(t_{end})$ by applying the GetLinkedSnapshot subroutine (line 5); if $SN(t_{prev})$ is relevant (line 6), then it is added to Σ and t_{start} is updated accordingly (lines 7-8). The loop may end due to the following conditions: (i) the current value of t_{end} reached t_0 (i.e., the time instant of the reference snapshot $\hat{SN}(t_0)$) or (ii) the previous linked snapshot is not relevant.

The rationale behind Algorithm (2) is that, once exploration at time t across the nodes of the Multi-Dimensional Model enabled experts to identify snapshots of interest for a given feature space fs_j and given values $d_1, \ldots d_p$ of analysis dimensions, the algorithm will also enable to retrieve the temporal evolution of micro-clusters within relevant snapshots back to the first time instant in which relevance of the snapshot emerged. This mechanism can be fruitfully applied in different exploration scenarios, for example to identify the changes that gradually led the monitored system in a specific working status, for diagnostic purposes, or as input for predictive maintenance strategies, that will be further investigated in future work.

To let the expert visualise all the evolution of micro-clusters across relevant snapshots, the data exploration GUI shown in Figure 3.6 provides an animation slider between time instants t_{start} and t_{end}. The expert can explore data over time by clicking on the "Play" button, or stop the stream clicking on "Pause" button.

Chapter 4

Implementation and experimental evaluation

4.1 The IDEAaS Architecture

Considering the complexity of the Big Data context in which IDEAaS operates, the architecture of the approach has been designed for being as modular as possible. It has been inspired by the architecture proposed in [62], expressly conceived for Big Data Analysis. To this purpose, the data acquisition, storage, processing and visualisation phases are clearly distinguished and decoupled in order to sustain the volume and acquisition rate of a Big Data ecosystem. Figure 4.1 shows the modular architecture.

In figure, the IDEAaS modules are distinguished in: (i) *Data Collection Modules*, that include *Data Configuration* and *Data Acquisition*, (ii) the *Core Modules*, that include the *Data Summarisation*, *Data Relevance Evaluation* and *Data Exploration API* and (iii) *IDEAaS GUI*, which is feed by the *Data Exploration API* and is designed to be used by experts who want to explore the collected data. All modules rely on and independently interact with the *IDEAaS Data Storage*, that includes DBMS technologies to store Multi-Dimensional Model metadata (i.e., analysis dimensions and their hierarchy), collected data and summarised data (i.e., micro-clusters and snapshots). In order to face the data volume and velocity during acquisition, data processing is strongly minimised to avoid bottlenecks and, in this respect, costly data elaboration steps (i.e., incremental clustering and relevance evaluation) are delegated to the *Core Modules*. This separation among modules allows an acquisition rate of $\sim$ 8240 measures per second on a mid-range computer system (Intel Core i7-6700HQ processor, CPU 2.60 GHz, 4 cores, 8 logical cores, RAM 16GB), as detailed in Section 4.2. In the

following, more details are provided on the implemented modules.

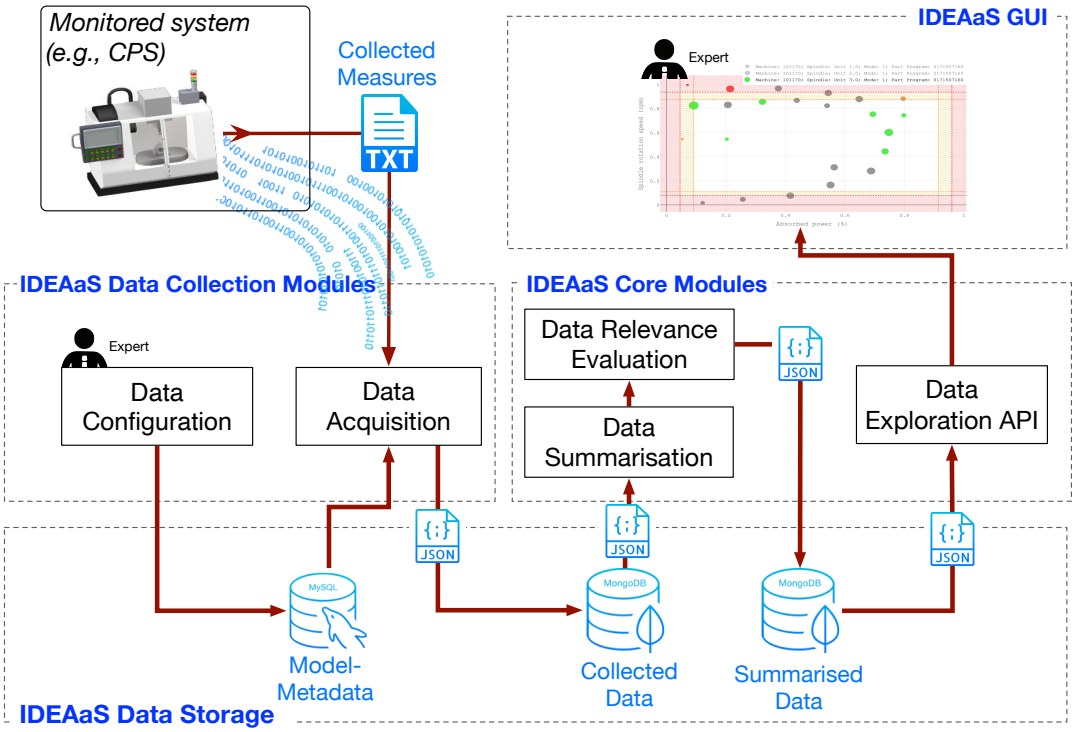


Figure 4.1: The IDEAaS architecture.

Data Collection Modules. These modules include the Data Configuration and the Data Acquisition modules. The *Data Configuration* module supports experts for the Multi-Dimensional Model definition (see Section 3.1), to save analysis dimensions and the composition of the feature spaces as sets of features. As shown through the arrows depicted in Figure 4.1, measures collected from the physical system, using sensors and IoT technologies, are pre-processed by *Data Acquisition* module. Technical details concerning this service are not given here, as they strictly depend on the application domain and, therefore, their description is out of the scope of this thesis. For example, this module may implement data cleansing and data quality control capabilities. Some future work will be discussed in this field in the last chapter. Collected measures are saved within a NoSQL database (*Collected Data*) as JSON documents using MongoDB technology and are organised into temporal-referenced collections (in particular, a new collection is created for each month). Each JSON document (automatically assembled by the *Data Acquisition* module) represents a record of collected measures within a feature space, and includes the timestamp and

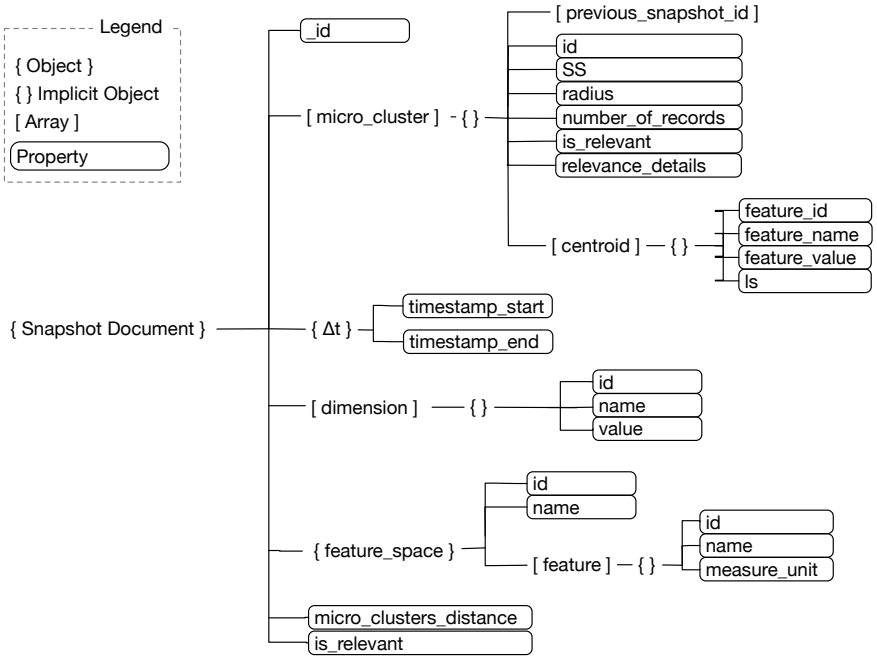


Figure 4.2: The structure of the JSON document to store relevance-enriched summarised data.

a reference to the dimensions in which the measures have been collected (such as the monitored machine, the tool used, the part program, saved in the *Model-Metadata* relational database).

Core Modules. The *Data Summarisation* module is in charge of summarising collected data, automatically updating the current set of micro-clusters and the corresponding snapshots, through the incremental clustering algorithm described in Section 3.2. It processes a batch of collected data every Δt seconds and at each iteration it generates a JSON document, containing the micro-clusters and snapshots properly organised with respect to the analysis dimensions of the Multi-Dimensional Model. The MongoDB technology is used also to store the results of incremental clustering algorithm computation (*Summarised Data*). The *Data Relevance Evaluation* module implements the relevance evaluation techniques described in Section 3.3. The output of this module is another JSON file, automatically generated every Δt seconds, where micro-clusters and snapshots organised in the multi-dimensional model are labelled with a relevance score. Figure 4.2 shows the structure of the JSON document defined to represent relevance-enriched summarised data. In particular, a *Snapshot Document* contains information about the set of micro-clusters in the snapshot, the time interval Δt over which the snapshot has been generated, the list of analysis dimensions,

information about the observed feature space and the relevance score computed for relevance evaluation as described in Section 3.3.1.

IDEAaS GUI. The *Data Exploration API* manages experts' interactions with the data exploration GUI. Every interaction, that requires to load micro-clusters and snapshots from the *Summarised Data* database, activates the *Data Exploration API*, that is in charge of communicating with the *IDEAaS Data Storage*. This module implements the last two tasks (relevance-based data exploration, selection of relevant data) shown in Figure 1.1.

Implementation technologies and system requirements. The base version of the IDEAaS system has been implemented in Java 8, on top of a Glassfish Server 4 Open Source Edition (Glassfish version 4.2). Apache Maven[1] has been integrated in the system in order to manage its dependencies, ensuring the required level of flexibility. Considering that IDEAaS has been developed in Java, it has been conceived as a multi-platform solution. Experimental evaluation (detailed in Section 4.2), has demonstrated that this configuration actually meets the characteristics of a mid-range computer system (Intel Core i7-6700HQ processor, CPU 2.60 GHz, 4 cores, 8 logical cores, RAM 16GB). For data visualisation, several graphical design libraries have been considered (e.g., D3js[2], Plotly[3] library). Plotly library has been chosen as a suitable candidate due to its versatility and ease of use. In the next chapter an evolution of the approach, where the time-consuming incremental clustering algorithm is further improved using a parallel architecture, will be described.

[1] `https://maven.apache.org/`
[2] `https://d3js.org/`
[3] `https://plot.ly/`

4.2 Experimental evaluation

4.2.1 Experimental setup

This section describes the experimental evaluation performed on the base version of IDEAaS. Experiments on the parallelised version of the incremental clustering algorithm will be described in the next chapter. The main goal of the experimental evaluation here is to demonstrate the effectiveness in promptly suggesting to experts' substantial variations in the collected data according to the definition of data relevance provided in this thesis and in presence of Big Data characteristics. The considered real time data, incrementally collected from monitored systems through sensors and IoT devices, is considered as an example of Big Data. Specifically, IDEAaS is able to handle volume and velocity of real time data, in presence of endless and incremental collection of data streams, and to cope with a high complexity of analysis dimensions.

To this aim, three experiments have been performed to evaluate: (i) the quality of data relevance evaluation techniques; (ii) the processing time, in order to verify if summarisation and relevance evaluation techniques, used to produce and store snapshots for data exploration, can face high data acquisition rates;A synthetic dataset has been built containing data inspired by the anomaly detection case study. In the second part of the thesis, concrete applications of the IDEAaS approach will be described. The dataset contains measures of 8 features for 2 feature spaces (corresponding to overlapping feature sets), collected at an acquisition rate up to $\sim$ 8000 measures per second. In total, the dataset contains $\sim$ $630,720,000$ measures collected in 6 months for the experiments. All the features present a normal distribution and have been generated considering three kinds of features in the motivating example, namely the currents (range: $-21 \div 22$ Amp, μ: $\sim$ 0 Amp, σ: $\sim$ 1.70 Amp), the speeds (range: $-23000 \div 24000$ mm/min, μ: $\sim$ -7 mm/min, σ: $\sim$ 4300 mm/min) and the rpm of the spindle (range: $0 \div 6400$ rpm, μ: $\sim$ 1850 rpm, σ: $\sim$ 2400 rpm). To simulate anomalous behaviours, in the dataset a percentage variation of values of features has been artificially introduced with respect to their value in normal working conditions. Such percentage changes randomly every 30 minutes. Moreover, two types of variations, namely gradual and sharp variations, have been introduced in order to evaluate the strength of the approach in different anomalous behaviours. Finally, measures in the collected dataset are associated with 4 analysis dimensions

organised within hierarchies with a depth level ranging from 1 to 3. Analysis dimensions have a number of instances that varies from 3 to 200; the average number of instances per dimension is equal to 58. Table 4.1 provides a summary of the characteristics of the dataset.

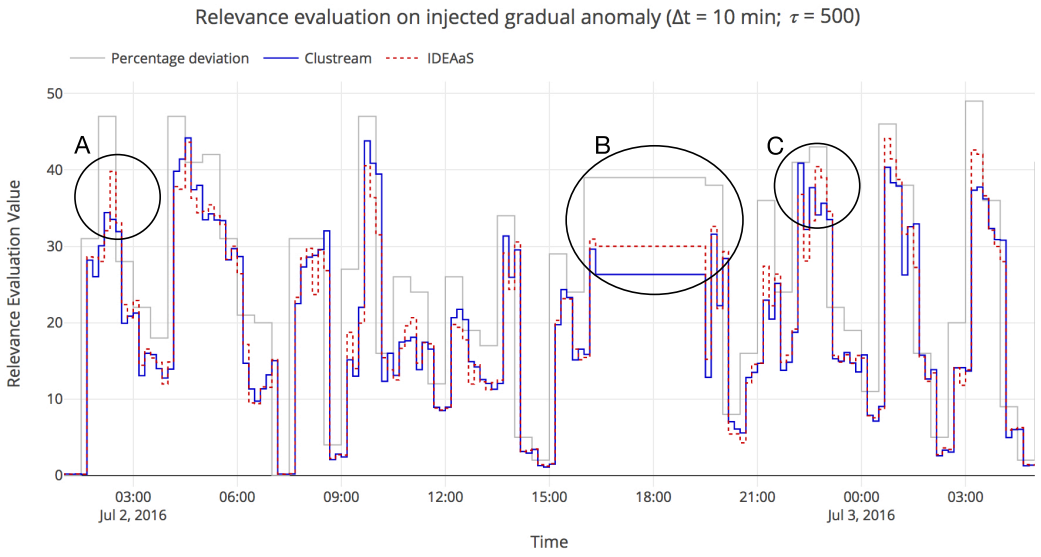
In the following, the three performed experiments will be described. Experiments have been performed on a MacBook Pro Retina with a screen resolution of 2880 x 1800 and a refresh rate of 60 Hz, having an Intel Core i7-6700HQ processor, CPU 2.60 GHz, 4 cores, 8 logical cores, RAM 16GB. The experiment on the exploration GUI, that has been implemented as a Web application, has been performed on the following browsers, considering their latest versions: Google Chrome v67, Firefox v59, Safari 11 and Microsoft Edge v42.

Table 4.1: Summary of the characteristics of the experimental dataset.

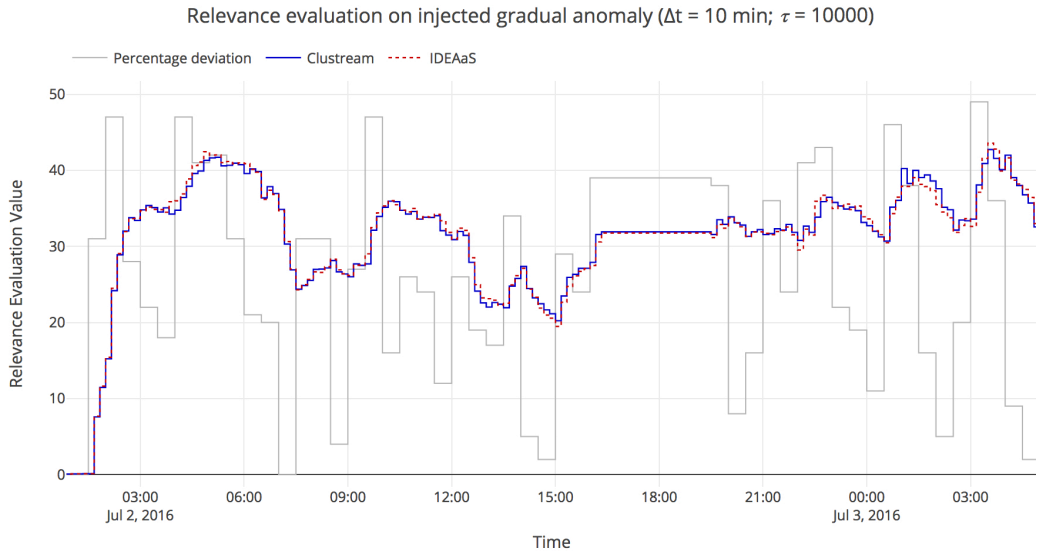
Highest data acquisition frequency	$0.1 \div 0.5$ sec
Number of features	8
Number of feature spaces	2
Max number of features per feature space	4
Average number of measures per month	105×10^6
Number of analysis dimensions	4
Hierarchy depth of analysis dimensions	$1 \div 3$
Average number of instances per dimensions	58

4.2.2 Experiment on relevance evaluation quality

The aim of this experiment has been to evaluate the impact of different micro-clusters update mechanisms on the quality of data relevance evaluation techniques. Indeed, the techniques presented in this thesis (that consider both the age and density of micro-clusters) have been compared with the CluStream algorithm [6], that only uses the age of the micro-clusters to update them. Relevance metrics have been evaluated for different values of the threshold τ, which is used to establish if a micro-cluster has to be considered old or not (see Section 3.2.2), since it has a direct impact on the capabilities of the approach to promptly detect variations. Moreover, the choice of τ is strongly related to the time interval Δt: choosing too large Δt values and too small τ (i.e. $\tau = 10$ and $\Delta t = 30$ min) values will induce the relevance evaluation approach to consider only the latest data arrived during Δt interval. Variations of incoming data, occurred before τ, may be assumed as already old and, therefore,



(a)



(b)

Figure 4.3: Correlation between the value of the percentage deviation from the values of dataset features in normal working conditions (gray line) and the value of $\Delta(\hat{\mu C}(t_0), \mu C(t))$ computed according to the snapshot relevance evaluation detailed in Section 3.3.1 for both IDEAaS (dotted red line) and CluStream (blue line) algorithms. (a) Relevance evaluation on sharp variations fixing threshold $\tau = 500$ (b) Relevance evaluation on sharp variations fixing threshold $\tau = 10000$.

not included among relevant data. On the other hand, choosing too large values of τ (i.e. $\tau = 10000$ and $\Delta t = 10$ min) does not eliminate old micro-clusters, causing resistance in the detection of new variations.

In this experiment Δt value has been set to 10 min. Further considerations on

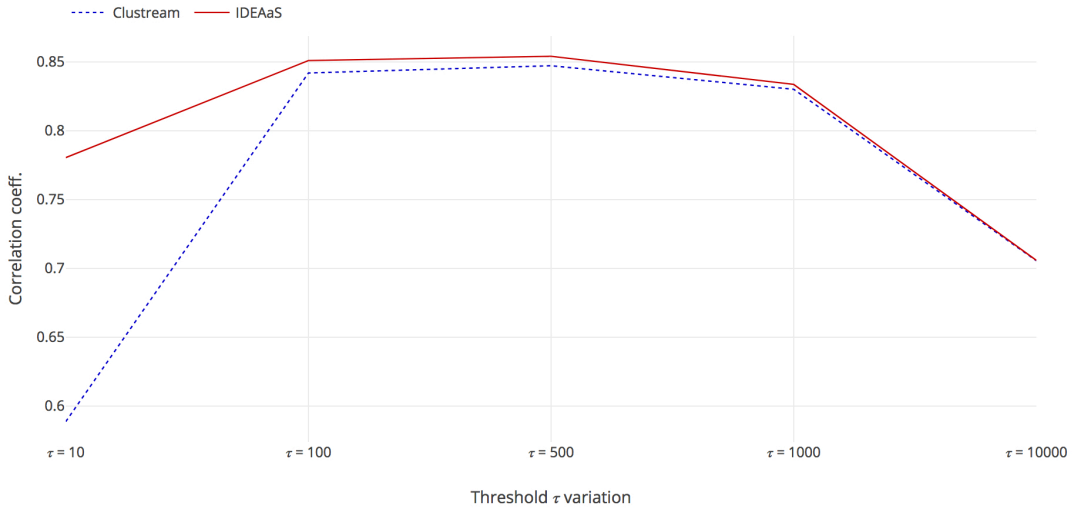
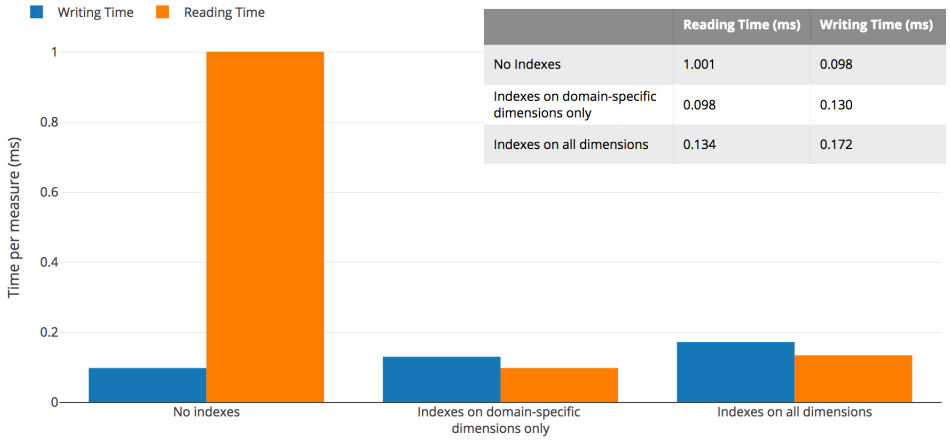


Figure 4.4: PCC between injected variations and the value computed with relevance evaluation techniques for $\Delta t = 10$ min when varying the ageing threshold τ, using micro-clusters update mechanisms of IDEAaS and CluStream [6].

the choice of Δt are reported in the next section. Several tests have been run with different values of τ threshold. Figure 4.3 reports the most interesting results. In particular, in Figure 4.3(a) $\tau = 500$ and in Figure 4.3(b) $\tau = 10000$. Figure 4.3 shows how the two compared solutions react to variations introduced in collected data. In Figure 4.3 only sharp variations have been considered, since for gradual variations both the algorithms produce effective results. Figure 4.3(b) shows how choosing too large τ values ($\tau = 10000$) inhibits algorithms from detecting introduced variations. In this case, both IDEAaS and CluStream can be considered as equivalent. The highlighted points (A) and (C) in Figure 4.3(a) show how IDEAaS relevance evaluation techniques are more effective in detecting the variations; the highlighted point (B) shows a case in which there is not new incoming data and both the algorithms remain stationary.

In order to quantify the correlation between the curves in Figure 4.3, the Pearson Correlation Coefficient (PCC) $\in [-1, +1]$ has been used. Figure 4.4 shows the PCC between injected variations and the values computed using relevance evaluation techniques for $\Delta t = 10$ min when varying the ageing threshold τ. As shown in the figure, the IDEAaS approach is more likely to follow the introduced variations, leading to a $PCC = 0.86$ in the best combination of τ and Δt ($\tau = 500$ and $\Delta t = 10$ min). When τ is too small ($\tau = 10$), the CluStream algorithm has the worst result with a $PCC = 0.5$. On the other hand, IDEAaS maintains a higher correlation



	Reading Time (ms)	Writing Time (ms)
No Indexes	1.001	0.098
Indexes on domain-specific dimensions only	0.098	0.130
Indexes on all dimensions	0.134	0.172

Figure 4.5: Average response time for writing and reading operations on the MongoDB database by applying different combinations of indexes.

with a $PCC = 0.78$. Similar results have been obtained even in the case of gradual variations.

Further experiments are in progress and aim to evaluate the impact of the centroids distance and difference in micro-clusters radii in order to evaluate data relevance (see Section 3.3.1). Negative or positive derivative of the micro-cluster radius, as well as changes of micro-clusters centroids, may be useful to distinguish among different kinds of variations (either sharp or more gradual), by properly varying weights α and β used to compute the distance between a pair of micro-clusters. Current experiments have been run with $\alpha = \beta = 1/2$, obtaining a satisfying trade-off between centroids distance and difference in micro-clusters radii.

4.2.3 Experiment on processing time

The experiment on processing time has been run in order to prove that IDEAaS can efficiently compute data summarisation and relevance evaluation, thus facing high acquisition rates.

Performances evaluation has been split in two parts: the evaluation of reading/writing operations in the NoSQL database (MongoDB), by introducing proper indexes, and the evaluation of the clustering algorithm.

Figure 4.5 shows how indexes impact on reading/writing response times for one record. When no indexes are applied, reading operations are the most expensive ones in terms of processing time, as expected. Indeed, a single read operation requires $\sim$ 1 ms, while a write operation requires $\sim$ 0.1 ms on average. On the other

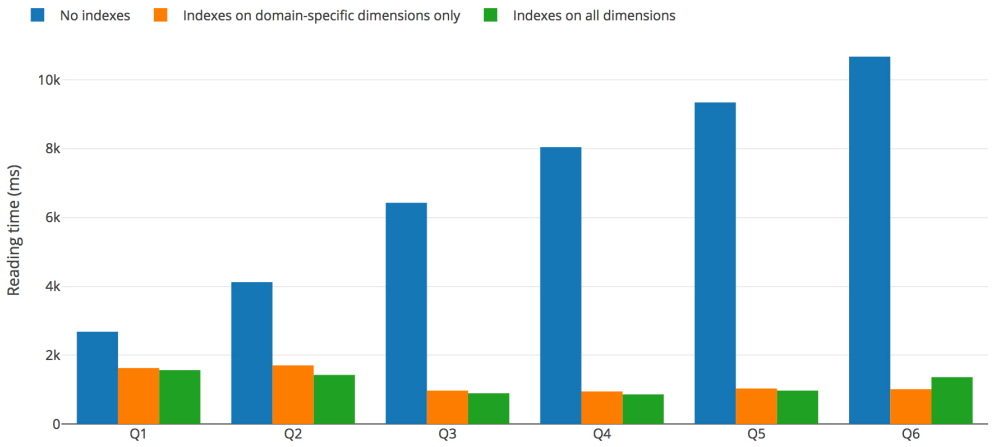
Query	Query Description
Q1	Query specifying timestamp range
Q2	Query specifying timestamp range and machine
Q3	Query specifying timestamp range, machine and spindle
Q4	Query specifying timestamp range, machine, spindle and mode
Q5	Query specifying timestamp range, machine, spindle, mode and part program
Q6	Query specifying timestamp range, machine, spindle, mode, part program and tool

Figure 4.6: Query types considered for performance evaluation.

hand, setting indexes on the analysis dimensions (including time and feature spaces) negatively impacts both on the acquisition and reading rate, with respect to the case where indexes are applied on all analysis dimensions, but not on the time and feature spaces. In the latter case, a single operation takes $\sim$ 0.13 ms for reading and $\sim$ 0.17 ms for writing.

The relevance-based data exploration strategy has been designed to isolate the exploration within a feature space and to speed up the exploration over time introducing links between snapshots. This enabled to setup indexes on analysis dimensions only, on which exploration steps are more frequent, avoiding to set indexes over feature spaces and time.

Considering this case, the capability of IDEAaS to face different data acquisition rates has been tested. The average time required to receive, process and save a new record is $\sim 0,97$ ms, leading to a maximum data processing rate equal to $\sim$ 8240 measures per second.



	Q1	Q2	Q3	Q4	Q5	Q6
No Indexes	2680.40	4123.50	6426.74	8043.19	9345.71	10677.11
Indexes on domain-specific dimensions only	1626.73	1703.19	971.50	946.01	1034.59	1011.95
Indexes on all dimensions	1567.62	1426.25	895.42	861.10	973.86	1364.80

Figure 4.7: Impact of different types of queries on the IDEAaS approach performances (data reading times in ms).

Further data reading performances have been investigated by considering different kinds of queries issued on the MongoDB database. For this purpose, different types of queries have been considered, as reported in Figure 4.6. Figure 4.7 shows how query complexity impacts on data reading performances with respect to different indexes set on the MongoDB database. Indeed, when no indexes are set, reading operation on a complex query (Q6) will take $\sim$ 10677 ms ($\sim$ 10.7 s). On the other hand, when indexes are set only on all analysis dimensions, but time and feature spaces, that represents the best scenario, the reading time is reduced to $\sim$ 1012 ms ($\sim$ 1.0 s). As shown in the figure, results confirm the above considerations made on different indexing strategies and the advantages brought by the proposed data exploration approach.

Additional time evaluation experiments have been focused on how variations of Δt values impact on the IDEAaS processing time. Processing time has been evaluated for data organised in collections where indexes on analysis dimensions have been set.

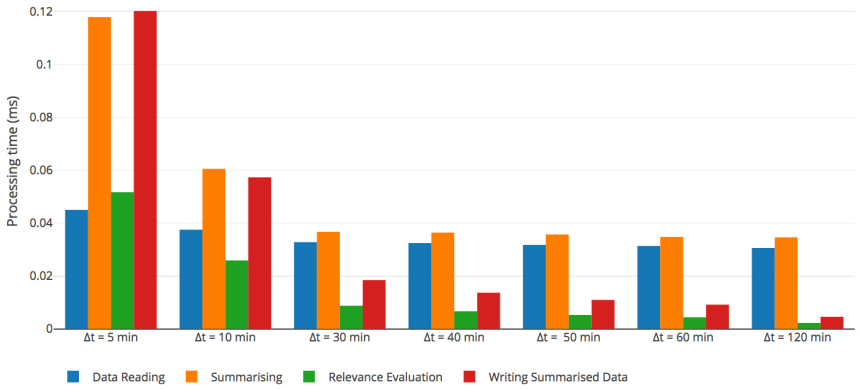


Figure 4.8: Average time of each step of the approach when varying Δt.

Figure 4.8 shows the average time required by each step of the approach to process records for different Δt values. The figure shows how lower Δt values require more time to process data. In fact, every time clustering is applied, some initialisation operations have to be performed (e.g., opening/closing connection to the database, access to the set of micro-clusters previously computed). Therefore, lower Δt values lead to more frequent initialisation operations. On the other hand, higher Δt values decrease the promptness in identifying variations on collected data.

	Number of steps to incrementally proccess ~3,440k measures	Avarage number of measures per second each step
Δt = 1 min	7299	~ 2,290
Δt = 5 min	1475	~ 23,894
Δt = 10 min	743	~ 44,158
Δt = 30 min	254	~ 82,573
Δt = 40 min	191	~ 89,587
Δt = 50 min	154	~ 95,468
Δt = 60 min	128	~ 100,235
Δt = 120 min	66	~ 111,044

Figure 4.9: Number of steps to process $\sim 3,440,000$ measures and the average number of data summarised per second at each step when varying Δt.

Finally, Figure 4.9 shows the number of steps required to incrementally summarise $\sim$3,440,000 measures and the average number of measures summarised per second for each step when varying Δt. Even in the worst case (Δt = 1 min), $\sim$ 2290 measures per second can be processed, thus demonstrating how IDEAaS is able to face satisfying data acquisition rates for the application domains like ones in which the IDEAaS approach has been applied.

Chapter 5

Parallel clustering of Big Data Streams

As underlined also in the previous chapters, the most time-consuming component in the IDEAaS approach is the incremental clustering algorithm. Efficient and effective techniques are needed to cluster data points in a stream, due to the increasing data volume and rate. For this reason, in this chapter a new version of the IDEAaS approach is proposed, named P-IDEAaS, where incremental clustering is designed as a multi-level parallel clustering approach, that starts from the Multi-Dimensional Model to partition the incoming data points. Furthermore, the approach relies on additional levels of parallelisation, that can be activated and tuned on demand to fulfill the exploration requirements, taking into account the available resources of the distributed processing architecture (e.g., by activating specific levels or by differently combining them over distinct partitions of the data stream, filtered using exploration facets and weighted according to the data relevance evaluation introduced in the previous chapters). Parallel data stream clustering approaches in the literature are either not conceived as a combination of multiple parallelisation levels or do not assume the possibility of enabling/disabling levels depending on the availability of computation resources (see Chapter 2 for an in-depth comparison). In the following sections, an explanation about the parallelisation levels conceived will be presented. Experiments to demonstrate the approach efficiency and effectiveness will be illustrated in Section 5.5.

Algorithm 3: Data stream parallel clustering procedure

Input: data stream DS, Multi-Dimensional Model MDM

1 **Procedure** `clusterEvolvingDataStream`(DS, MDM)**:**
2 $\quad\Phi \leftarrow$ GETFACETS(MDM) ▷ set of possible facets
3 $\quad$**foreach** *facet* $\varphi_i \in \Phi$ **do parallel** ▷ 1^{st} parallel level
4 $\quad\quad$**while** OBSERVEFACET$(\varphi_i) = true$ **do**
5 $\quad\quad\quad DS_i \leftarrow$ PARTITIONDATASTREAM(DS, φ_i)
6 $\quad\quad\quad \overline{X}_i \leftarrow$ GETNEWDATAPOINTS$(DS_i, \varphi_i.\Delta t)$ ▷ set $\overline{X}_i$ of new data points for facet φ_i collected in $[t - \varphi_i.\Delta t,\ t]$
7 $\quad\quad\quad \mu C^{\varphi_i}_{curr} \leftarrow$ GETCURRENTMICROCLUSTERS(φ_i)
8 $\quad\quad\quad \mu C^{\varphi_i}_{new} \leftarrow$ PARALLELUPDATEMICROCLUSTERS$(\mu C^{\varphi_i}_{curr}, \overline{X}_i)$ ▷ Algorithm (4) generates new set of micro-clusters
9 $\quad\quad\quad \hat{\mu C}^{\varphi_i} \leftarrow$ GETSTABLEMICROCLUSTERS(φ_i) ▷ $\hat{\mu C}^{\varphi_i}$ set of micro-clusters generated in normal working conditions
10 $\quad\quad\quad \varphi_{i_{relevance}} \leftarrow$ RELEVANCEEVALUATION$(\mu C^{\varphi_i}_{new}, \hat{\mu C}^{\varphi_i})$ ▷ relevance evaluation for facet φ_i
11 $\quad\quad\quad$**if** $\varphi_{i_{relevance}} > threshold$ **then** ▷ *threshold* defined by domain expert based on the criticality of the monitored system (see Definition 9 in Chapter 3)
12 $\quad\quad\quad\quad \varphi_i.\Delta t \leftarrow$ DECREASEDELTAT(φ_i)
13 $\quad\quad\quad$**else**
14 $\quad\quad\quad\quad \varphi_i.\Delta t \leftarrow$ INCREASEDELTAT(φ_i)
15 $\quad\quad\quad$**end if**
16 $\quad\quad\quad \varphi_i.$SETCURRENTMICROCLUSTERS$(\mu C^{\varphi_i}_{new})$
17 $\quad\quad\quad$WAIT$(\varphi_i.\Delta t)$
18 $\quad\quad$**end**
19 $\quad$**end foreach**
20 **End Procedure**

5.1 Parallelisation based on exploration facets

The first level of parallelisation involves the exploration facets as shown in Algorithm (3). The Multi-Dimensional Model has been specifically conceived to partition data points in the stream according to the exploration facets. It is worth remarking here that some analysis dimensions that compose exploration facets may dynamically change during the data stream collection: for example, the tool used by the CNC machine in the anomaly detection case study, or the physical activity performed by the monitored patient in eHealthcare case study.

The GETFACET function in Algorithm (3) on line 2 is invoked to return all the facets defined in Φ (see Definition 5 in Chapter 3). The facets on which exploration must be primarily focused are those identified as relevant according to the data relevance evaluation within those facets (see Definition 9 in Chapter 3). Therefore, the OBSERVEFACET function (line 4) is used to focus on relevant facets only. Relevant

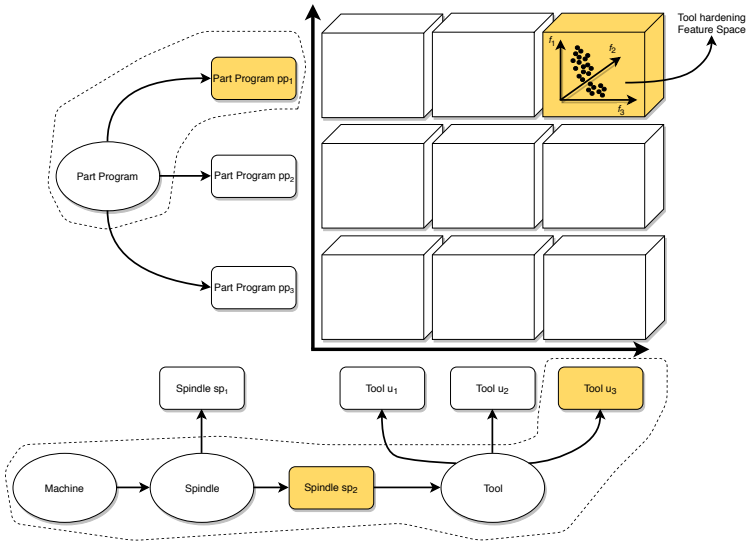


Figure 5.1: Multi-Dimensional Model for data stream exploration in the smart factory domain.

facets are used to partition the incoming data stream (PARTITIONDATASTREAM function on line 5) and to run the clustering algorithm in parallel upon these partitions (lines 3-19), according to the Multi-Dimensional Model (see the highlighted small cube in the upper right corner of Figure 5.1). In parallel for each partitioned stream DS_i, incoming data points collected every $\varphi_i.\Delta t$ seconds (GETNEWDATAPOINTS function on line 6) and the current set of clusters (GETCURRENTMICROCLUSTERS function on line 7) are processed to generate the new set of micro-clusters (PARALLELUPDATEMICROCLUSTERS function on line 8). The latter function is and extension of Algorithm (1) in Chapter 3 by introducing parallelisation, as reported in Algorithm (4). Parallelisation will be detailed in the next subsections.

Once the new set of micro-clusters has been obtained, the relevance evaluation detailed in Section 3.4 is performed in order to detect changes in the set of new micro-clusters $\mu C_{new}^{\varphi_i}$ with respect to the set of stable micro-clusters $\hat{\mu C}^{\varphi_i}$, corresponding to the normal working conditions for facet φ_i (lines 9-10). The $\varphi_i.\Delta t$ value represents the buffer over which micro-clusters generation and relevance evaluation are performed, that may vary depending on the facet φ_i. Choosing a proper buffer size influences the ability of the system to promptly detect micro-clusters changes

Algorithm 4: Parallel micro-clusters update function

Input: set μC^{φ_i} of micro-clusters given an exploration facet φ_i, set $\overline{X}$ of new data points
Output: updated set $\mu C^{\varphi_i}_{new}$ of micro-clusters

1 **Function** `parallelUpdateMicroClusters(`μC^{φ_i}, $\overline{X}$`):`
2 $\quad \overline{X}_{out} \leftarrow \{\}$ ▷ $\overline{X}_{out}$ data points that could not be assigned to μC^{φ_i}
3 $\quad$ **foreach** *data point* $\overline{x}_i \in \overline{X}$ **do parallel** ▷ 2^{nd} level (buffering)
4 $\quad\quad \mu c_{near} \leftarrow$ FINDCLOSESTMICROCLUSTER$(\mu C^{\varphi_i}, \overline{x}_i)$ ▷ 3^{rd} level parallel
5 $\quad\quad$ **if** $\mu c_{near} == null$ **then**
6 $\quad\quad\quad \overline{X}_{out} \leftarrow \overline{X}_{out} \cup \{\overline{x}_i\}$
7 $\quad\quad$ **end if**
8 $\quad$ **end foreach**
9 $\quad$ **while** SIZEOF$(\overline{X}_{out}) > 0$ **do**
10 $\quad\quad \overline{x}_{out} \leftarrow$ PICKANDREMOVEFROM$(\overline{X}_{out})$
11 $\quad\quad$ **if** $|\mu C^{\varphi_i}| == MAX$ **then**
12 $\quad\quad\quad \mu c_{remove} \leftarrow$ GETMICROCLUSTERTOREMOVE(μC^{φ_i}) ▷ 3^{rd} level parallel
13 $\quad\quad\quad$ **if** $\mu c_{remove} \neq null$ **then**
14 $\quad\quad\quad\quad \mu C^{\varphi_i}_{new} \leftarrow \mu C^{\varphi_i} \setminus \{\mu c_{remove}\}$
15 $\quad\quad\quad$ **else**
16 $\quad\quad\quad\quad (\mu c_a, \mu c_b) \leftarrow$ GETMICROCLUSTERSTOMERGE(μC^{φ_i}) ▷ $\mu c_a \neq \mu c_b$; 3^{rd} level parallel
17 $\quad\quad\quad\quad \mu C^{\varphi_i}_{new} \leftarrow$ MERGE$(\mu c_a, \mu c_b)$
18 $\quad\quad\quad$ **end if**
19 $\quad\quad$ **end if**
20 $\quad\quad \mu c_{new} \leftarrow$ CREATENEWMICROCLUSTER$(\overline{x}_{out})$
21 $\quad\quad \overline{X}_{out} \leftarrow$ ASSIGNDATAPOINTS$(\overline{X}_{out}, \mu c_{new})$
22 $\quad\quad \mu C^{\varphi_i}_{new} \leftarrow \mu C^{\varphi_i} \cup \{\mu c_{new}\}$
23 $\quad$ **end**
24 $\quad$ **return** $\mu C^{\varphi_i}_{new}$
25 **End Function**

corresponding to abnormal behaviours. The shorter $\varphi_i.\Delta t$ value, the faster the reaction of the system to an occurring warning or error, but at the same time the more frequent the micro-clusters updating procedure, thus requiring more parallelisation resources as explained in the next subsections and Algorithm (4). The relevance evaluation value is used to set the buffer size (lines 11-15). Higher data relevance, exceeding the threshold (see Definition 9 in Chapter 3), implies the need to reduce $\varphi_i.\Delta t$ (DECREASEDELTAT function on line 12). On the other hand, if no critical data is detected, $\varphi_i.\Delta t$ can be increased (INCREASEDELTAT function on line 14). The adaptive change of $\phi_i.\Delta t$ can be usefully exploited in anomaly detection applications, as explained in Chapter 6. Additional levels of parallelisation can be applied during

the generation and update of micro-clusters, as summarised in Algorithm (4) and described in the following.

5.2 Parallelisation based on data buffering

Algorithm (4) is logically divided in two parts: (i) firstly, data points that can be assigned to one of the existing micro-clusters are distinguished from those that require the generation of new micro-clusters (lines 2-8), denoting the latter data points as $\overline{X}_{out}$; (ii) therefore, further updating of the micro-clusters starting from $\overline{X}_{out}$ is performed (lines 9-23).

Concerning the first part, instead of processing one data point at a time, data points are collected in a buffer that spans a temporal lapse equal to $\varphi_i.\Delta t$. Specifically, the parallelisation based on data buffering focuses on the FINDCLOSESTMICROCLUSTER function (line 4). As a result, all the data points within the buffer will be assigned to the micro-clusters in parallel and not in a serialised way, as illustrated in Figure 5.2, except for data points that cannot be assigned to any existing micro-cluster ($\overline{X}_{out}$). Note that the data relevance evaluation, used to tune the value of $\varphi_i.\Delta t$ as previously explained, has an impact on this parallelisation level.

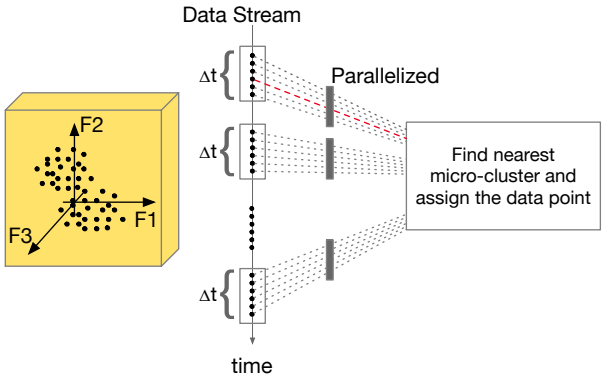


Figure 5.2: Parallelisation based on data buffering.

Data points in $\overline{X}_{out}$ are interpreted as new emerging behaviours in the monitored system. With respect to the other data points, they are not processed in parallel, since their introduction requires the removal (GETMICROCLUSTERTOREMOVE function on line 12) or merging (GETMICROCLUSTERSTOMERGE function on line 16) of existing micro-clusters, if the maximum number of micro-clusters has been already reached (line 11), and in any case the introduction of new micro-clusters (CREATENEWMICROCLUSTER function on line 20), thus affecting the subsequent

processing steps and the computation cost. Therefore, processing of data points in $\overline{X}_{out}$ is postponed and data points that do not require to update the current set of micro-clusters are processed first. Nevertheless, even if their processing is not executed in parallel, a third level of parallelisation can be applied, as shown in the algorithm and described in the next subsections for functions on line 12 (micro-clusters removal) and on line 16 (micro-clusters merging).

Moreover, when a new micro-cluster has been created, the algorithm checks if any other data point in $\overline{X}_{out}$ can be assigned to it (ASSIGNDATAPOINTS function on line 21). This procedure will assign all the possible data points in $\overline{X}_{out}$ to the new micro-cluster (without checking the whole set of existing micro-clusters, to which data points in $\overline{X}_{out}$ can not be assigned by definition, see line 2) and returns only the remaining ones. This may improve the efficiency of $\overline{X}_{out}$ processing, as explained in Section 5.4. However, postponing the assignment of data points in $\overline{X}_{out}$ requires that the quality of clustering does not depend on the order in which data points are processed. To assess whether the processing order of data points has a tangible effect on the quality of clustering, relevance evaluation techniques has been exploited, and experiments reported in Section 5.5 has been performed.

5.3 Parallelisation based on the set of micro-clusters

The third level of parallelisation concerns the three operations of finding the closest micro-cluster (see Algorithm (4) on line 4), finding the micro-cluster to remove (on line 12), finding micro-clusters to merge (on line 16). In the following paragraphs, the parallel execution of these operations is explained.

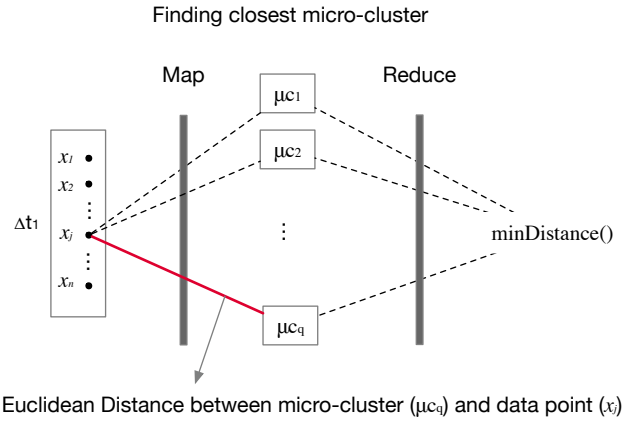


Figure 5.3: Parallel calculation of distance between data points and micro-clusters.

Finding the closest micro-cluster Figure 5.3 reports the parallel implementation of the FINDCLOSESTMICROCLUSTER function according to the Map-Reduce paradigm. To find the closest micro-cluster for an incoming data point, the Euclidean distance between the data point and the centroid of each micro-cluster is computed, and the closest micro-cluster is selected. As can be seen in the figure, all the Euclidean distance calculations between an incoming data point x_j and the centroids of existing micro-clusters $\mu c_1 \dots \mu c_q$ are performed in parallel, in the Map step. After all the distance values have been calculated, the minimum one is identified in the Reduce step.

Finding the micro-cluster to remove Parallel implementation of the GETMICRO-CLUSTERTOREMOVE function is illustrated in Figure 5.4 according to the Map-Reduce paradigm. It performs the parallel computation of the *getRelevanceStamp* function, which checks if the micro-clusters are candidate to be removed according to the logic explain in Section 3.2.2 in Chapter 3, for each of the existing micro-clusters $\mu c_1 \dots \mu c_q$ (Map step). Among the micro-clusters candidate to be removed, if any, the oldest one is chosen (Reduce step).

This function does not require any calculation of Euclidean distance between data points, but involves the linear and quadratic sum of timestamps $CF1_i^t$, $CF2_i^t$ of each micro-cluster μc_i.

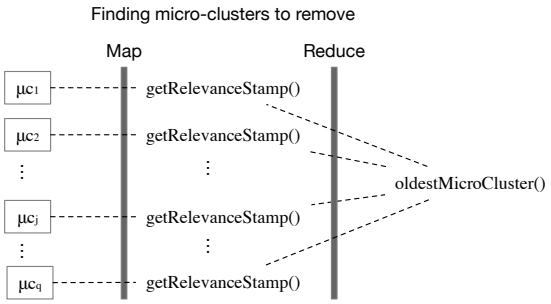


Figure 5.4: Parallel calculation of relevance stamps of micro-clusters in order to identify micro-clusters to remove.

Finding micro-clusters to merge If no removable micro-clusters have been found, then the two closest micro-clusters must be merged together. The mutual proximity between two micro-clusters μc_a and μc_b corresponds to the evaluation of the mutual Euclidean distance of their centroids $\overline{X0}_a = \langle x_{1\mu c_a}, x_{2\mu c_a}, \dots, x_{d\mu c_a} \rangle$ and $\overline{X0}_b = \langle x_{1\mu c_b}, x_{2\mu c_b}, \dots, x_{d\mu c_b} \rangle$, where d denotes the number of features. The computation

of this distance can be executed in parallel for each pair μc_a and μc_b according to the Map-Reduce paradigm. The distance $D(\mu c_a, \mu c_b)$ between two micro-clusters is calculated as:

$$D(\mu c_a, \mu c_b) = \sqrt{\sum_{i=1}^{d} (x_{i\mu c_a} - x_{i\mu c_b})^2} \tag{5.1}$$

Parallel implementation of the GETMICROCLUSTERSTOMERGE function is illustrated in Figure 5.5. The proximity calculation between clusters is executed in parallel in the Map step, while the identification of the minimum distance for merging the pair of closest micro-clusters μc_a and μc_b is performed in the Reduce step (ensuring that $\mu c_a \neq \mu c_b$).

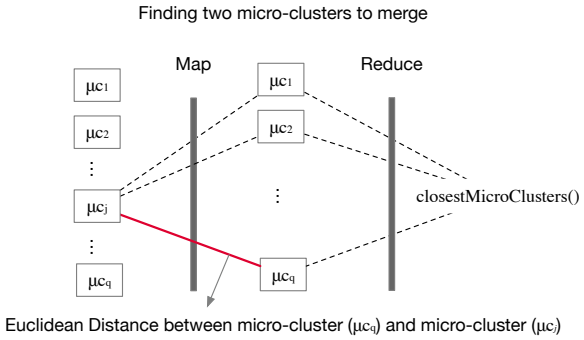


Figure 5.5: Pairwise Euclidean distance calculation between micro-clusters to identify closest micro-clusters to merge.

5.4 Complexity analysis

Let's denote with $|\overline{X}|$ the number of data points collected in $\varphi_i.\Delta t$, with $|\overline{X}_{out}|$ the number of data points that cannot be assigned to any micro-cluster, with p the parallelisation degree (i.e., the number of computation nodes used to enable parallel computation of the distance from micro-cluster centroids), with d the number of dimensions, with q the number of micro-clusters, with N_q the average number of points in micro-clusters (to be considered for the computation of $CF1_i^t$ and $CF2_i^t$). The computation complexity of the operation to find the closest-micro-cluster is $O(\frac{|\overline{X}|-|\overline{X}_{out}|}{p} \cdot d \cdot q)$. The complexity of the operation to find the micro-cluster to remove in the worst case is $O(\frac{q}{p} \cdot N_q \cdot |\overline{X}_{out}|)$, since this operation must be performed for each data point in $\overline{X}_{out}$. Finally, the complexity of the third operation to find the micro-clusters to merge in the worst case is $O(\frac{q^2}{p} \cdot d \cdot N_q \cdot \overline{X}_{out})$, since every distance must be computed for every possible pair of q micro-clusters and for each data point

in $\overline{X}_{out}$. Computation complexity of the three operations shows as the less expensive operation is the fist one, as expected. The choice of postponing the processing of data points in $\overline{X}_{out}$ aims at increasing cases in which the only operation required is the first one, thus reducing the overall complexity of the clustering.

The three levels of parallelisation that have been introduced in this chapter can be combined together. Combining differently these levels, in order to adapt the parallel implementation of the clustering algorithm to the availability of parallelisation resources, is one of the contributions of P-IDEAaS version of the approach compared to the literature. Positive and weak points of combining these levels will be discussed after presenting the experimental results.

5.5 Experimental Evaluation

The experiments described in this chapter have been performed using the dataset introduced in the experimental setup in Section 4.2.1 in Chapter 3, in order to compare performances of different configurations and combinations of the proposed parallelisation levels. One of the most practical advantages brought by a multi-level parallelisation approach relies on the fact that levels can be differently enabled depending on the investigated scenario, evaluating the real necessities of applying all parallelisation levels or only a subset of them, with a trade-off between scalability and parallelisation costs. The point here is to identify some general guidelines to select and tune parallelisation levels, depending on the characteristics of the data stream that is being processed in terms of complexity in the variety of exploration facets and in terms of data relevance. To this aim, the MDM and the data relevance evaluation techniques come to the rescue. In the following, experiments to figure out these aspects will be described, namely the impact of parallelisation levels on the overall performances of the parallel implementation of the algorithm and the feasibility of parallelisation (due to the data points in $\overline{X}_{out}$, see Algorithm (4) above).

Experiments have been performed on Apache Spark (version 2.3.1 for Hadoop 2.7+), running one Spark master and three Spark slaves, configured with Docker (engine: 18.06.1-ce). Each node has been configured with Intel-i5 2.60 GHz dual-core processor and a 4GB memory RAM.

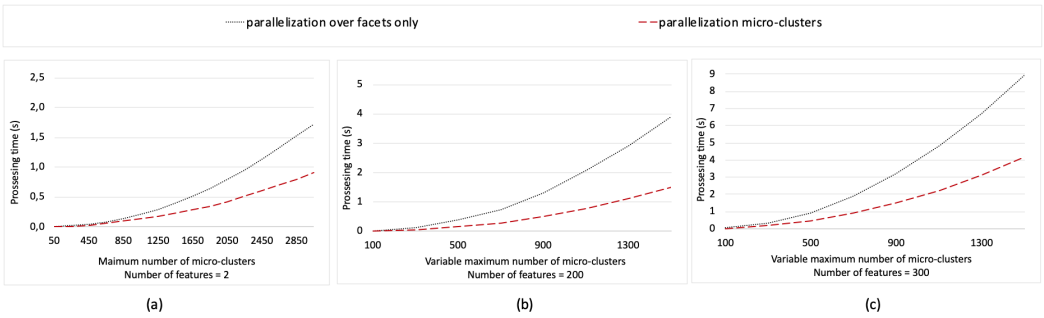


Figure 5.6: Processing time of parallel data stream clustering by varying the maximum number of allowed micro-clusters and by setting the number of features to 2 (a), to 200 (b) and to 300 (c). Parallelisation levels based on exploration facets and on micro-clusters are applied.

5.5.1 Scalability of parallelisation levels

To test the impact of combining different parallelisation levels on the scalability of the approach, several experiments have been performed. Considering the undisputed advantages of the parallelisation based on the exploration facets, experiments have been mainly focused on the combination of this level with the one based on data buffering and with the third parallelisation level (based on micro-clusters). Combinations of parallelisation levels have been tested by varying the maximum number of allowed micro-clusters (q) and the number of data points in a buffer (n).

Figure 5.6 shows results of scalability experiments by combining the first parallelisation level (based on facets) and the third one (based on micro-clusters), by varying the maximum number of allowed micro-clusters and the number of features. The application of the third level of parallelisation is always convenient and reduces the processing time with respect to the parallelisation based on facets only as expected. These parallelisation levels have been therefore combined with the second parallelisation level (based on buffering) and tested by varying the number of data points in the buffer and different number of micro-clusters (Figure 5.7). Figure 5.7 shows how the second parallelisation level decreases its scalability as the number of data points in the buffer increases for the same number of micro-clusters. If the maximum number of allowed micro-clusters is not comparable with the cardinality (i.e., number of data points) of the buffer, potentially a greater number of data points can be assigned to $\overline{X}_{out}$. As explained in Section 5.4, assignment of data points in $\overline{X}_{out}$ moves the main computation effort towards the GETMICROCLUSTERTOREMOVE and GETMICROCLUSTERSTOMERGE functions in Algorithm (3) and makes the performance

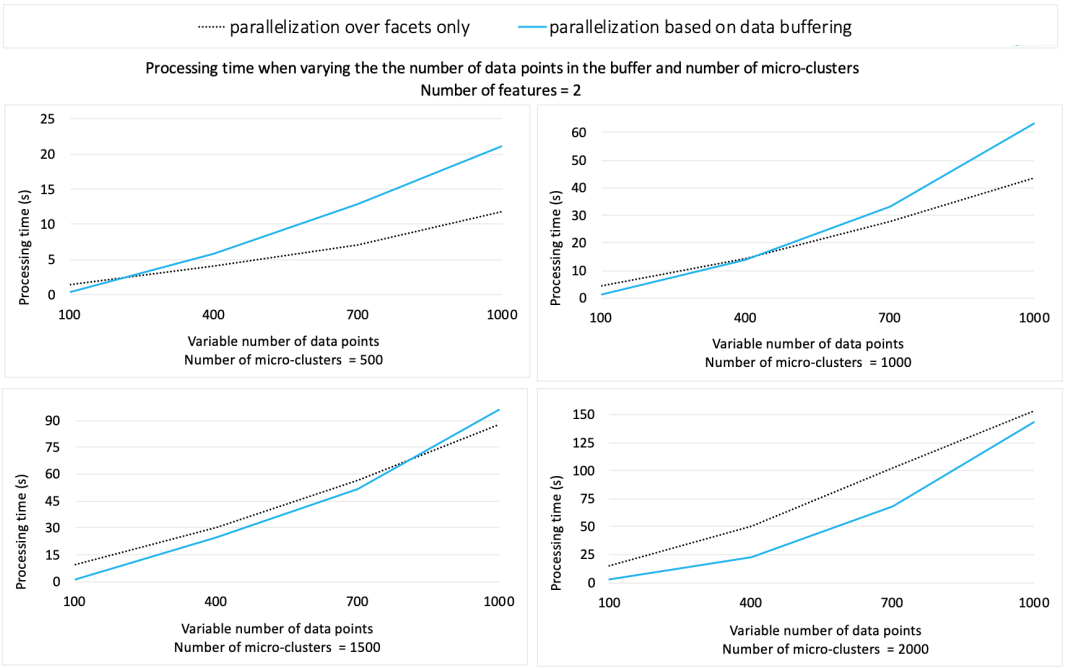


Figure 5.7: Impact of the second level of parallelisation (based on data buffering) on the processing time when varying the maximum number of allowed micro-clusters and the number of data points in the buffer (number of features set to 2).

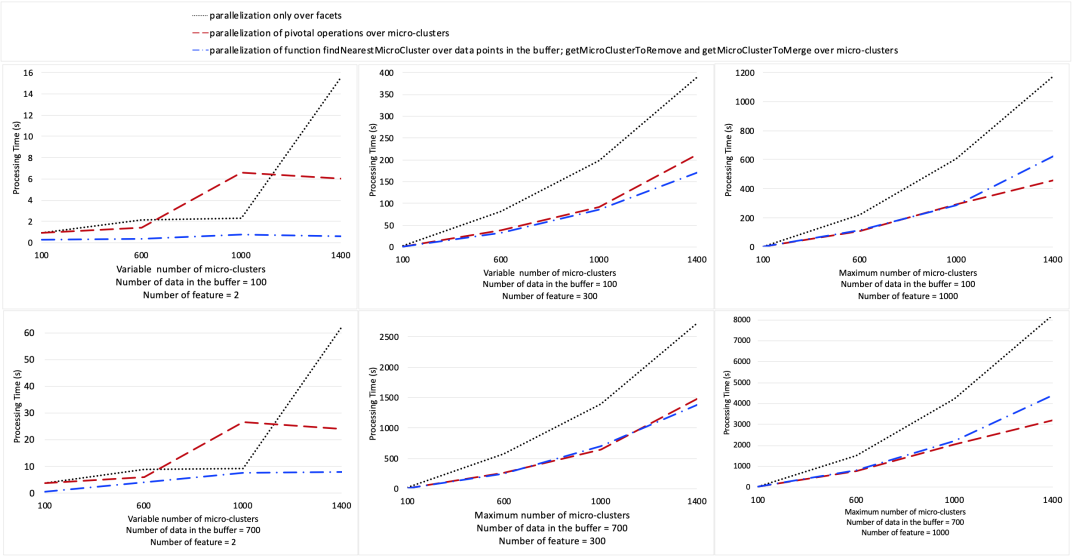


Figure 5.8: Impact of different combinations of parallelisation levels on processing time when varying maximum number of micro-clusters and the number of data points in the buffer (vertically), while the number of features is set to 2 (a), to 300 (b), to 1000 (c).

improvement due to the parallelisation based on data buffering less effective.

Finally, Figure 5.8 reports different combinations of all parallelisation levels for different numbers of data points in the buffer and different numbers of features,

when varying the maximum number of allowed micro-clusters. Figure 5.8 shows how parallelisation levels can be differently combined depending on the characteristics of the data stream that is being processed. When the size of the buffer exceeds the maximum number of allowed micro-clusters, scalability of the approach decreases if the parallelisation level based on buffering is activated. Only in case of low complexity (i.e., low number of features measured on the monitored system) the application of all the three levels of parallelisation brings the maximum advantages.

5.5.2 Parallelisation feasibility

Data relevance evaluation is exploited to determine if the parallelisation based on data buffering is worth being performed or not. In fact, the detection of anomalies based on relevance evaluation is caused by changes in the set of generated micro-clusters, that can be due to the variation of micro-clusters radius, to changes in micro-clusters position and to the appearance of new micro-clusters. Specifically, changes are detected if the value of $\Delta(\mu C^{\varphi}_{curr}, \hat{\mu C}^{\varphi})$ for a given exploration facet φ, according to Equation (3.4), exceeds a given threshold. This is due to data points that cannot be assigned to any existing micro-cluster, but require the generation of new micro-clusters that substitute existing ones, that are removed or merged according to Algorithm (4). In the algorithm the set of such data points is denoted as $\overline{X}_{out}$. Data points in $\overline{X}_{out}$ are processed after all the other data points in the buffer have been assigned to existing micro-clusters. As explained in Section 5.4 and in the experiments on the scalability of the approach, this decreases the performance, depending on the size of the buffer, the maximum number of allowed micro-clusters and the percentage of data points in $\overline{X}_{out}$ compared to the buffer size. When the cardinality of $\overline{X}_{out}$ set increases, the parallelisation based on buffering is expected to have a lower effect on the scalability of the approach and can be skipped, to avoid useless consumption of computation resources. Therefore, an experiment has been performed to evaluate the feasibility of parallelisation based on data buffering and what are the effects of $\overline{X}_{out}$ on the data relevance evaluation results. In the experiment, both the cardinality of $\overline{X}_{out}$ and the position of data points of $\overline{X}_{out}$ into the buffer have been considered. In fact, since data relevance evaluation techniques are used to attract the attention on the relevant portions of the data stream only, if the value of $\Delta(\mu C^{\varphi}_{curr}, \hat{\mu C}^{\varphi})$ is invariant with respect to the characteristics of $\overline{X}_{out}$, then parallelisation based on data buffering can be considered as feasible and effective and is worth being applied. This

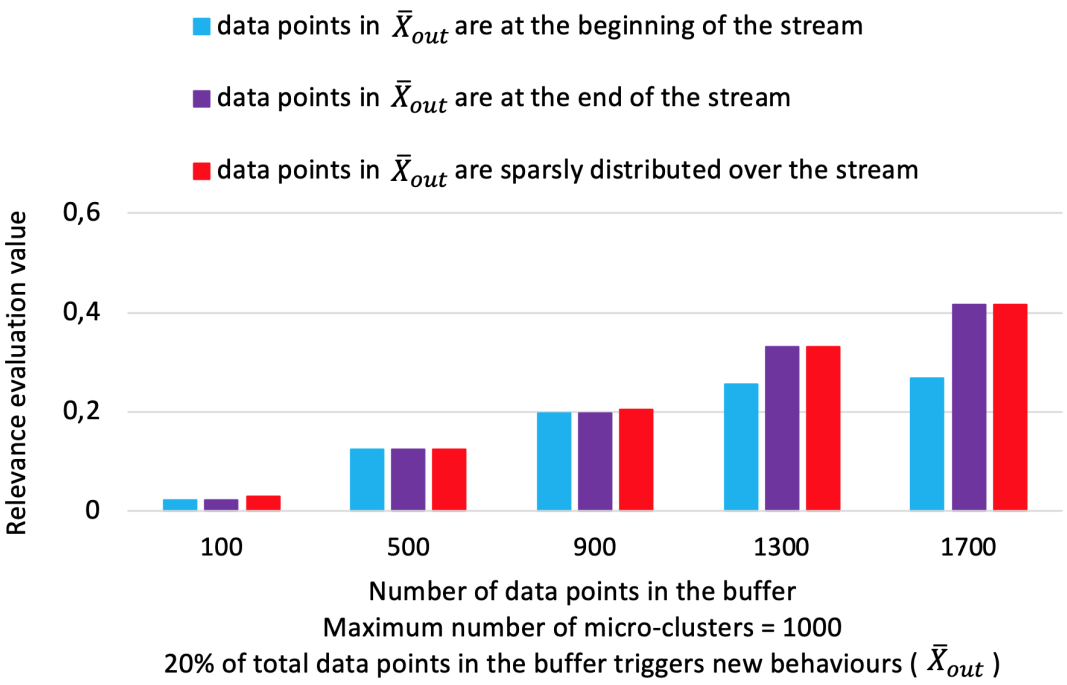


Figure 5.9: Relevance evaluation $\Delta(\mu C^{\varphi}_{curr}, \hat{\mu C}^{\varphi})$, with respect to $\hat{\mu C}^{\varphi}$, in a given exploration facet φ, when varying the number of data points in the buffer and setting maximum number of micro-clusters equal to 1000.

check can be performed in the first portion of the data stream that is being analysed and can be reinforced periodically.

Figure 5.9 reports the relevance evaluation $\Delta(\mu C^{\varphi}_{curr}, \hat{\mu C}^{\varphi})$ for different permutations of $\overline{X}_{out}$ and for different sizes of the buffering. Specifically, three different cases have been tested, considering 1000 micro-clusters (q=1000) and varying incoming data points, with 20% of them belonging to the set $\overline{X}_{out}$. Data points in $\overline{X}_{out}$ have been positioned at the beginning, at the end and equally distributed into the analysed portion of the buffer. As shown in Figure 5.9, the position of data points of $\overline{X}_{out}$ within the buffer has less effect in the relevance evaluation as the buffer size decreases. In fact, reduced values of $\varphi.\Delta t$ means a reduced parallelisation based on data buffering, that is, data points are processed sequentially, mitigating the problem introduced by new behaviours in parallelisation as explained in Section 5.2. This is also true when the maximum number of allowed micro-clusters is low. Intuitively, this is caused by the fact that a limited number of micro-clusters entails a scarce possibility for a novel data point to be assigned to one of them and, therefore, that is more likely to be labelled as a new behaviour and assigned to $\overline{X}_{out}$. For this reason, the parallelisation based on buffering with few micro-clusters and a lot of data

points (i.e., high buffer size) is not appropriate as the number of data points assigned to $\overline{X}_{out}$ increases. On the other hand, this parallelisation level is feasible and appropriate when the maximum number of micro-clusters is high enough to represent well the number of data points in the buffer.

5.5.3 Final considerations

The experimental results on the scalability of parallelisation levels suggest the following considerations:

- as the maximum number of allowed micro-clusters decreases, when the number of data points in the buffer increases, parallelisation based on data buffering does not ensure a good scalability (see Figures 5.7 and 5.8);
- as the number of features increases (that is, the complexity of data stream increases as well), more data points belonging to $\overline{X}_{out}$ can be potentially found since a higher number of measures is considered; again, such data points have a negative impact on scalability due to the parallelisation based on data buffering (Figure 5.8).

Therefore, the length $\varphi.\Delta t$ of the buffer for a given exploration facet φ can be varied until the data relevance evaluation $\Delta(\mu C^{\varphi}_{curr}, \hat{\mu C}^{\varphi})$ is invariant with respect to the characteristics of the $\overline{X}_{out}$ set. Following these considerations, that derive from the experimental evaluation described in this chapter, the application of the Multi-Dimensional Model is able to reduce the computational effort required for data stream clustering by partitioning the stream and enabling a first level of parallelisation as expected. This level can be combined with the other two levels, but this does not automatically ensure an improvement in terms of scalability, since it depends on the buffer size, the dynamicity of data in the stream (i.e., the cardinality of the $\overline{X}_{out}$ set) and the complexity of the stream (i.e., the number of features).

The application of the other two levels has to be properly set and this is performed by relying on the data relevance evaluation techniques, that may prevent from the application of costly and useless parallelisation levels. Therefore, these techniques can be used to decide about the parallelisation feasibility. A performance comparison with other horizontal scaling platforms will be performed in the future using common datasets and the same experimental setup. The aim of this work has

been mainly on demonstrating the effectiveness of an adaptive tuning of parallelisation levels with respect to the characteristics of the stream, that revealed better performance compared to the application of all possible parallelisation strategies. Similar performance increments with respect to a full-fledged application of parallelisation levels could be potentially observed also in other approaches.

Part II

Applications

Chapter 6

Big Data exploration for anomaly detection

In this chapter the adoption of the Multi-Dimensional Model, data summarisation and relevance evaluation techniques to implement anomaly detection based on data streams will be discussed. This application scenario is based on the anomaly detection for smart factory case study introduced in 1.3.1 on the multi-spindle machine reported here in Figure 6.1.

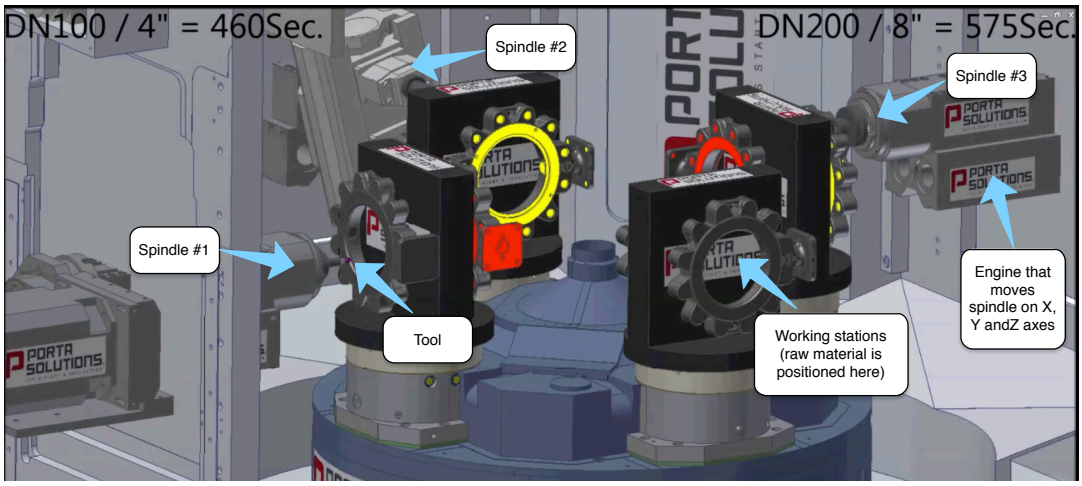


Figure 6.1: The multi-spindle machine from which real time data have been collected for exploration purposes.

In this case study, a set of anomaly detection services in the smart factory scenario have been designed and tested, by relying on the IDEAaS system. The work described in this chapter has been performed in the Smart4CPPS project[1] and has been published in the following papers:

[1]This project has been funded by Lombardy Region (2018-2021): http://www.smart4cpps.it/

[63] Ada Bagozi, Devis Bianchini, Valeria De Antonellis, and Alessandro Marini. A relevance-based data exploration approach to assist operators in anomaly detection. In Proc. of 26th Int. Conference on Cooperative Information Systems (CoopIS2018), pages 354–371, Valletta, Malta, 2018;

[49] Ada Bagozi, Devis Bianchini, Valeria De Antonellis, Alessandro Marini, and Davide Ragazzi. Big Data Summarisation and Relevance Evaluation for Anomaly Detection in Cyber Physical Systems. In Proc. of 25th Int. Conference on Cooperative Information Systems (CoopIS 2017), pages 429–447, 2017.

6.1 Anomaly detection services in a nutshell

Figure 6.2 presents the IDEAaS modular architecture extended through the introduction of *Anomaly Detection Services*. The main purpose is to implement data-intensive functionalities in order to enable the anomaly detection in a smart factory scenario.

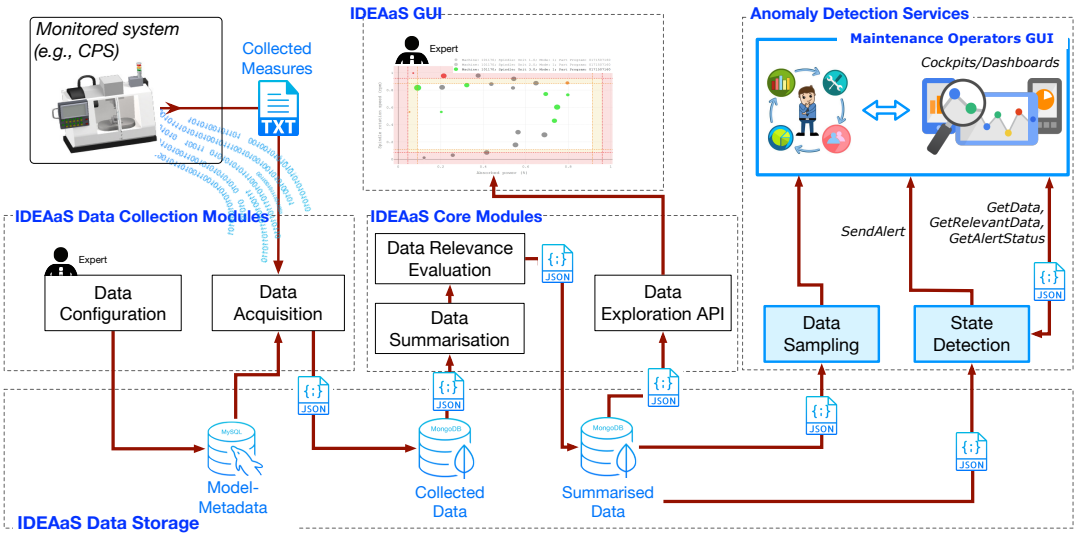


Figure 6.2: Anomaly detection services built on top of the IDEAaS modular architecture.

Among *Anomaly Detection Services*, *Data Sampling* based on data summarisation and relevance evaluation techniques has been introduced in order to reduce the total amount of data to be visualised on the operator's cockpit. The way data summarisation, relevance evaluation and sampling techniques are used to assist operators in anomaly detection is enabled by the *State Detection* service, which is detailed in the next sections. Finally, a remote visualisation cockpit, called *Maintenance Operators*

GUI, has been designed on top of the *Data Sampling* and *State Detection* services, to guide the operators during anomaly detection and relevant data discovery.

State Detection Service. The State Detection Service is in charge of detecting possible anomalies through the observation of data collected from the monitored system and managing the interaction with visualisation tools, such as cockpits and dashboards, on which (maintenance) operators can explore data.

In order to detect anomalies, four different values for the *status* of the monitored system have been considered, (a) `ok`, when the system works normally; (b) `changed`, when the system behaviour changed with respect to the normal one, but no anomalies have been detected yet; (c) `warning`, when the system works in anomalous conditions that may lead to breakdown or damage; (d) `error`, when the system works in unacceptable conditions or does not operate. The `changed` and `warning` status are used to perform an early detection of a potential deviation towards an error status. The `warning` or `error` status occurs when one or more features exceed a given bound. Besides defining *features* bounds, the notion of *contextual bounds* have been introduced. A contextual bound represents the limit of a feature within specific conditions (e.g., determined by the tool used and/or the part program that is being executed) in which the feature is measured. The rationale is that, in specific conditions, a feature should assume values within a specific range, that might be different from the overall physical limits for the same feature disregarding the working conditions. In this context, conditions can be identified through analysis dimensions of the Multi-Dimensional Model. If the measure overtakes warning bounds, but not the error ones, then the feature status is set as `warning`, otherwise the feature is in the `error` status. Features (contextual) bounds are fixed by domain experts, for instance through to the FMEA/FMECA analysis. The operators can monitor state changes in order to revise features and contextual bounds for specific working conditions.

The *State Detection Service* includes data relevance evaluation techniques to attract the operator's attention on state changes. In fact, the State Detection Service provides the following methods, as remarked in Figure 6.2:

- `SendAlert` sends asynchronous notifications about detected changes of the working status in the monitored system, based on Summarised Data; to this aim, this method relies on the Data Relevance Evaluation module of the IDEAaS

architecture and adapts the anomaly detection frequency according to the data relevance, as explained in the next sections;

- `GetAlertStatus` sends a summary report on the current status of the monitored system; this service is required to synchronise visualisation tools to the current status of the physical system, when external cockpits and dashboards get connected with the State Detection Service.

Data visualisation must take into account the high volume of information to be visualised and facilitate the interaction of operators with the Graphical User Interface (GUI) of the visualisation tool. To this purpose, the following additional methods are exposed by the State Detection Service:

- `ExploreRelevantData` sends relevant data, by relying on the Data Relevance Evaluation module; data is transferred as micro-clusters about measures (obtained through the application of IDEAaS incremental clustering algorithm) and visualised according to the Multi-Dimensional Model; this method has been designed to support (maintenance) operators to focus on relevant data only, without specifying any data search and filtering criteria, since operators might not have any a-priori knowledge about which data can be considered as relevant;

- `GetData` sends data within a given time interval and/or for specific search and filtering criteria expressed on dimensions of the Multi-Dimensional Model; this functionality can be used, for example, once relevant summarised data has been identified; since sent data may reach a massive size, sampling techniques are applied; hence, sampling takes into account the relevance of data that is being transmitted, by adapting the sampling ratio to the data relevance.

6.2 Relevance-based data exploration for anomaly detection

For anomaly detection purposes, for each micro-cluster $\overline{\mu c}_c \in \overline{\mu C}(t)$, where $\overline{\mu C}(t)$ is the set of micro-cluster identified as relevant (see Section 3.3.1 in Chapter 3), the distance of micro-cluster centroid from the warning and error bounds is computed. In the following, absolute features bounds will be considered, but the same considerations hold for the contextual ones. The record vector of distances (one value for each feature in the feature space) between the centroid of the micro-cluster $\overline{\mu c}_c$ and the

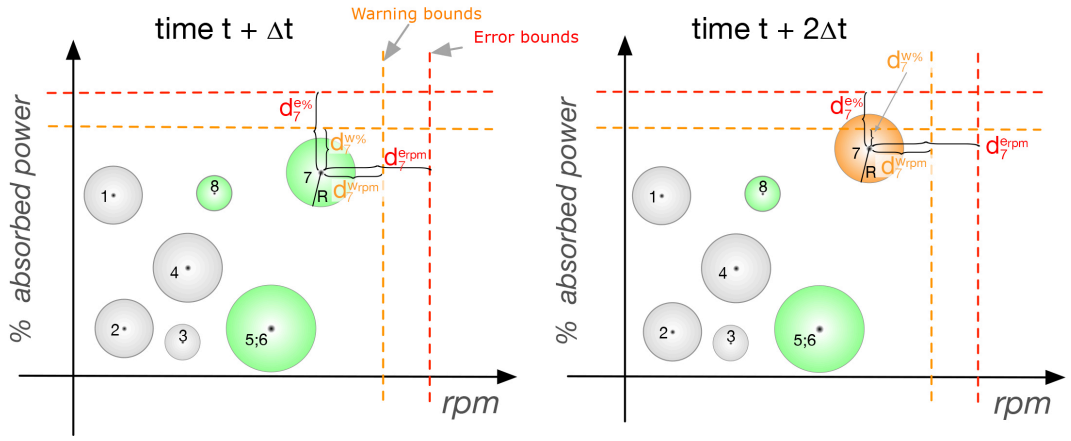


Figure 6.3: Anomaly detection through data exploration based on relevance evaluation in the multi-spindle case study: data relevance techniques detect changes in micro-clusters set due to spindle rolling friction torque increase, that may be identified when the rpm value decreases and, at the same time, the power absorption increases.

warning bounds is denoted with $\mathbf{d}_c^w$ and the record vector of distances between the centroid of $\overline{\mu c}_c$ and the error bounds is denoted with $\mathbf{d}_c^e$. The State Detection Service uses $\mathbf{d}_c^w$ and $\mathbf{d}_c^e$ to perform anomaly detection, by distinguishing among `ok`, `warning` and `error` status. Figure 6.3 reports an example of anomaly detection performed on micro-clusters computed on the mukti-spindle machine considered in the first case study. As mentioned in the case study, spindle rolling friction torque increase and the tool wear are two possible problems that are frequently monitored on this kind of machines. Monitoring in these cases is performed through the collection of power absorption (% absorbed power) and kinematic features such as accelerations (rpm). If an increased power absorption is detected disregarding the tool that is used, it is possible to identify a problem in the spindle rolling friction torque increase. On the other hand, if the increase in absorbed power is related only to the usage of a particular tool, this can be recognised as a symptom of a possible tool wear. In the spindle rolling friction torque increase example, $d_7^{e_{\%}}$ represents the distance of the centroid of the micro-cluster $\overline{\mu c}_{c7}$ from the error bound of the percentage of absorbed power (see Figure 6.3). Each relevant micro-cluster in $\overline{\mu c}_c$ is therefore enriched with distances from warning and error bounds in order to enable anomaly detection as:

$$\overline{\mu c}_c = \langle \overline{CF1_c^x}, \overline{CF2_c^x}, CF1_c^t, CF2_c^t, n_c, \mathbf{d}_c^w, \mathbf{d}_c^e \rangle \tag{6.1}$$

Every Δt seconds, when the set of all micro-clusters $\mu C(t)$ is updated, data is analysed to check for anomalies, updating the set of critical micro-clusters $\overline{\mu C}(t)$.

For example, in Figure 6.3 micro-cluster $\overline{\mu c}_{c7}$ moved over time getting closer to the boundaries. Note that distance also helps to detect *potential* state changes. In fact, at time $t + \Delta t$ micro-cluster $\overline{\mu c}_{c7}$ still remains inside the wealth zone (`ok` status), but its movement is detected through relevance-based techniques. Therefore, micro-cluster $\overline{\mu c}_{c7}$ is recognised as relevant and monitored to promptly detect potential `warning` or `error` status occurrences. After Δt seconds, micro-cluster $\overline{\mu c}_{c7}$ moved again and crosses the warning bound of the percentage of absorbed power feature, raising a warning alert. The `warning` status is assigned to the feature and is propagated to the feature space and over the hierarchy of monitored system according to the following rules: (i) the status of a feature space corresponds to the worst one among its features; (ii) similarly, the status of a physical component (e.g., the spindle) corresponds to the worst one among monitored feature spaces on that component and the status of composite systems (e.g., the multi-spindle machine) corresponds to the worst one among its components. Figure 6.3 also shows that it is possible to identify the feature with respect to the warning or error bound that has been exceeded (e.g., among rpm and percentage of absorbed power). If a `warning` or `error` status is detected, the State Detection Service notifies an alert message to the cockpit with the new status, using the `SendAlert` method. This check is performed every Δt seconds. When a micro-cluster moved closer to bounds, the IDEAaS system reacts by reducing the interval time Δt to check data for anomalies as described in the following.

6.3 Adaptive relevance evaluation

Setup of Δt parameter influences the performances of the anomaly detection approach. Small Δt values increase the promptness in identifying relevant micro-clusters, in order to attract the attention of the operators on them. On the other hand, response times of data acquisition and clustering may not be able to face small Δt values (see experimental evaluation in Section 4.2). The rationale behind the anomaly detection approach is to change Δt as micro-clusters get closer to `warning` and `error` bounds, since they correspond to potentially critical situations that must be monitored at finer granularity.

To this aim, Δt value is changed according to the distance of relevant micro-cluster $\overline{\mu c}_c \in \overline{\mu C}(t)$ that is closer to `warning` and `error` bounds. We denote with $d_c^{w\_min}$ (resp., $d_c^{e\_min}$) the component of $\mathbf{d}_c^w$ (resp., $\mathbf{d}_c^e$) that presents the minimum distance from the `warning` bounds (resp., the `error` bounds). The interval time Δt is updated as follows:

- if $\frac{d_c^{w\_min}}{R} > 1$, the feature status is set to `ok` (see for example micro-cluster $\overline{\mu c}_{c7}$ in Figure 6.3 at time $t + \Delta t$), Δt is set to a default value defined by the domain expert according to his/her knowledge about the monitored system;
- if $\frac{d_c^{w\_min}}{R} <= 1$ and $\frac{d_c^{e\_min}}{R} > 1$ the micro-cluster centroid is between warning bounds and error bounds (see for example micro-cluster $\overline{\mu c}_{c7}$ in Figure 6.3 at time $t + 2\Delta t$), the feature status is set to `warning`, Δt is reduced as $\Delta t = \Delta t(\frac{d_c^{e\_min}}{R} - 1)$ until $\Delta t =$ minimum value supported by the approach (see experimental evaluation in Section 4.2);
- if $\frac{d_c^{e\_min}}{R} <= 1$ the micro-cluster centroid is beyond error bounds, the feature status is set to `error`, Δt is set to the minimum supported value (that is, checks are made as more frequently as possible).

Adaptive sampling for data visualisation. An effective visualisation of an unexpected working status and related data on operator's cockpit must consider the impact of data volume and velocity, to avoid operators be overwhelmed by the huge amount of data. To this purpose, data sampling techniques are usually applied, where sampling is performed taking into account the size and capacity of the cockpit interface, independently of the specific conditions which visualised data refers to. In the anomaly detection approach described in this chapter, that relies on the IDEAaS system, clustering and relevance evaluation techniques are used to implement adaptive sampling for data visualisation. To this purpose, `ExploreRelevantData` and `GetData` methods of the State Detection Service have been implemented.

Request for relevant data. When the operator at time t requests for relevant data, the method `ExploreRelevantData` is invoked. This method relies on relevance evaluation techniques to recognise the most recent relevant micro-clusters set $\overline{\mu C}(t_i)$, processed at time t_i ($t_i <= t$). Each micro-cluster $\overline{\mu c}_c \in \overline{\mu C}(t_i)$ is marked with

the corresponding status and with additional information about whether the micro-cluster moved, changed (expansion or contraction) or has been removed. All micro-clusters in $\overline{\mu C}(t_i)$ recognised as anomalous are properly highlighted with different colours as shown in Figure 6.3.

Exploration of relevant micro-clusters. Once relevant micro-clusters have been identified, the operator may request to explore in detail records that have been clustered within relevant micro-clusters. These records are returned by invoking the `GetData` method. Records may correspond to a time-window h, and for specific values of analysis dimensions, the amount of extracted data may be really large and difficult to visualise. In order to enable data visualisation, a classical adaptive sampling technique has been designed. Nevertheless, in this approach sampling frequency varies according to data relevance evaluation. Considering max_n as the maximum number of data supported by the visualisation tool and n as the number of data points extracted from the database, when $n >> max_n$ a sampling technique is applied selecting only max_n data points among the n data points that are ready for visualisation. Sampling rate is adaptively modified by a factor that depends on the detected status (`warning` or `error`) within the time-window. When data is not recognised as critical, the sampling rate is set to the minimum value. In the case all data in the interval is not relevant, or is equally relevant, the sampling frequency is set to $\frac{max_n}{t-h}$. This strategy facilitates the cooperation between operators who act remotely on powerful visualisation interfaces and on-site operators, who may need data visualisation on less powerful HMI embedded in or close to the monitored machine, by setting different values of max_n.

Figure 6.4 shows the design of remote visualisation cockpit. The cockpit guides data exploration through analysis dimensions in the considered domain, therefore it first considers the monitored system, along with the relevant feature spaces. Figure 6.4 shows an overview of the data of the multi-spindle machine with ID 101143 and its status. In the overview, the operator can visualise the status of the three spindles of multi-spindle machine, denoted with `"Unit 1.0"`, `"Unit 2.0"` and `"Unit 3.0"`. Indeed spindle `"Unit 1.0"` is working correctly with respect to all the observed feature spaces, while spindle `"Unit 2.0"` is in `warning` status. In particular, micro-clusters calculated for features `"f4"` and `"f5"` are detected as relevant and associated to the

Figure 6.4: Visualisation of relevant data on operator's cockpit in the anomaly detection application scenario (`GetData` method).

`warning` status. Therefore, the `warning` status is propagated to the `"tool wear"` feature space as well. Finally, spindle `"Unit 3.0"` is in error status. In fact, even if the `"tool wear"` warning status has been detected, a more critical status is identified for feature space `"spindle rolling friction torque increase"`. Starting from relevant data, the operator may request to visualise data in details through the `GetData` method, as shown in Figure 6.4. Moreover, the operator may further explore data by setting the time interval of data to be plotted and the other dimensions of the Multi-Dimensional Model (such as the tool or the part program) to filter data in the exploration process. In this example max_n is fixed to 3600 records. The value of max_n can be chosen considering the device on which the operator is navigating. On the left part of Figure 6.4, the operator requests to visualise data corresponding to the spindle rolling friction torque increase of `"Unit 3.0"` spindle. In this case the amount

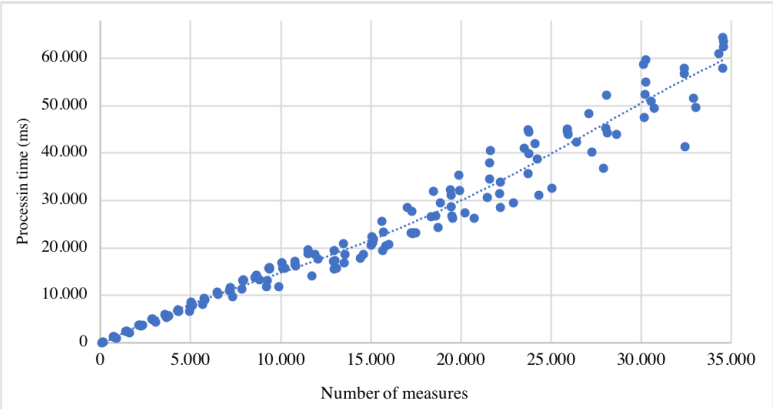


Figure 6.5: Response times of the State Detection Service with respect to the number of processed measures.

of data to be visualised is under the max_n value, therefore the sampling techniques are not applied. In the right part of Figure 6.4 the operator selected a wider time interval for the same feature space and dimensions, that, in this scenario, corresponds to 7200 records, exceeding the max_n value. In the figure is shown how all the data, without sampling, is plotted on the cockpit: due to the high number of measures, it is evident that this visualisation would be not valuable for the operator.

6.4 Experimental evaluation

Experiments on the State Detection Service have been performed in order to test its performance in terms of processing time and its effectiveness in promptly detecting anomalies. In this case study, a real dataset has been considered, based on measures collected from three multi-spindle machines, each of them mounting three spindles. For each spindle the values of 8 features have been collected every 500ms. Globally an acquisition rate of 144 measures per second has been addressed. After six months of monitoring on the three machines 630,720,000 measures have been collected. Experiments have been run on a MacBook Pro mounting MacOS High Sierra, 2,8 GHz Intel Core i7, RAM 16GB.

Figure 6.5 plots response times of the State Detection Service with respect to the number of analysed measures. As evident in the figure, response times proportionally (but not exponentially) increase with the number of processed measures. As shown in Figure 6.5 the designed State Detection Service can process 35000 measures

in 60 seconds on average, corresponding to $\sim$ 583 measures per second. Therefore, it can successfully cope with the acquisition rate.

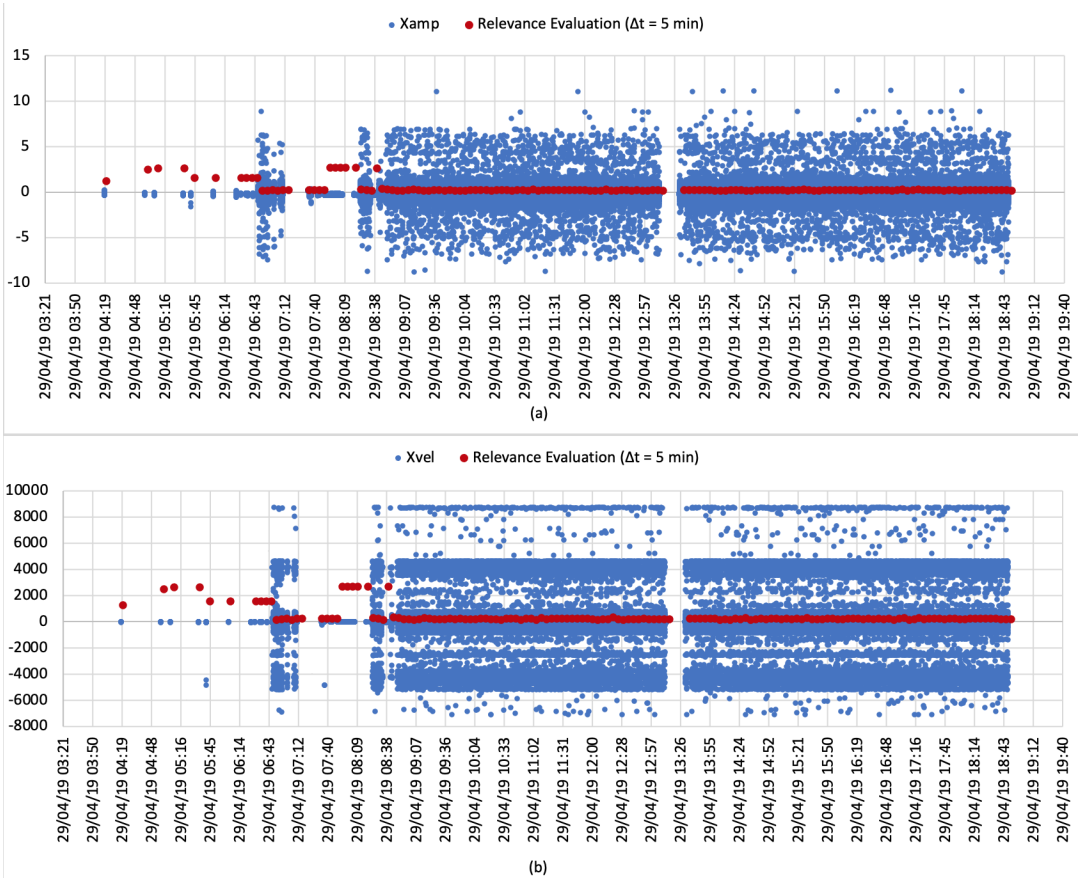


Figure 6.6: Relevance evaluation for anomaly detection applied every 5 minutes (Δt = 5 min) on feature space composed by the electrical current absorbed (a) and the velocity (b) of X axes.

The effectiveness of the service to detect anomalies has been tested using the measures collected on the movements of each spindle over X, Y and Z axes. Specifically, the velocity of the spindle movement over each axis (e.g., Xvel) has been measures, together with the electrical current (e.g., Xamp) absorbed by the engines that are responsible of moving the spindle over the axes (one engine for each axis).

In Figure 6.6 anomaly detection service has been applied every 5 minutes (Δt = 5 min) considering the X axis. In the figure, blu points represent single data points in the stream, corresponding to measured values for the Xvel and Xamp features. Detecting anomalies by directly observing these data points appear as really difficult for an operator who is exploring the collected data. 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 Section 3.3. Figure shows how the relevance evaluation techniques are able of detecting anomalies on the real data stream. Indeed, operators who performed a maintenance intervention on the machine that has been monitored in this case confirmed that on April 29 there have been anomalies on the monitored machine.

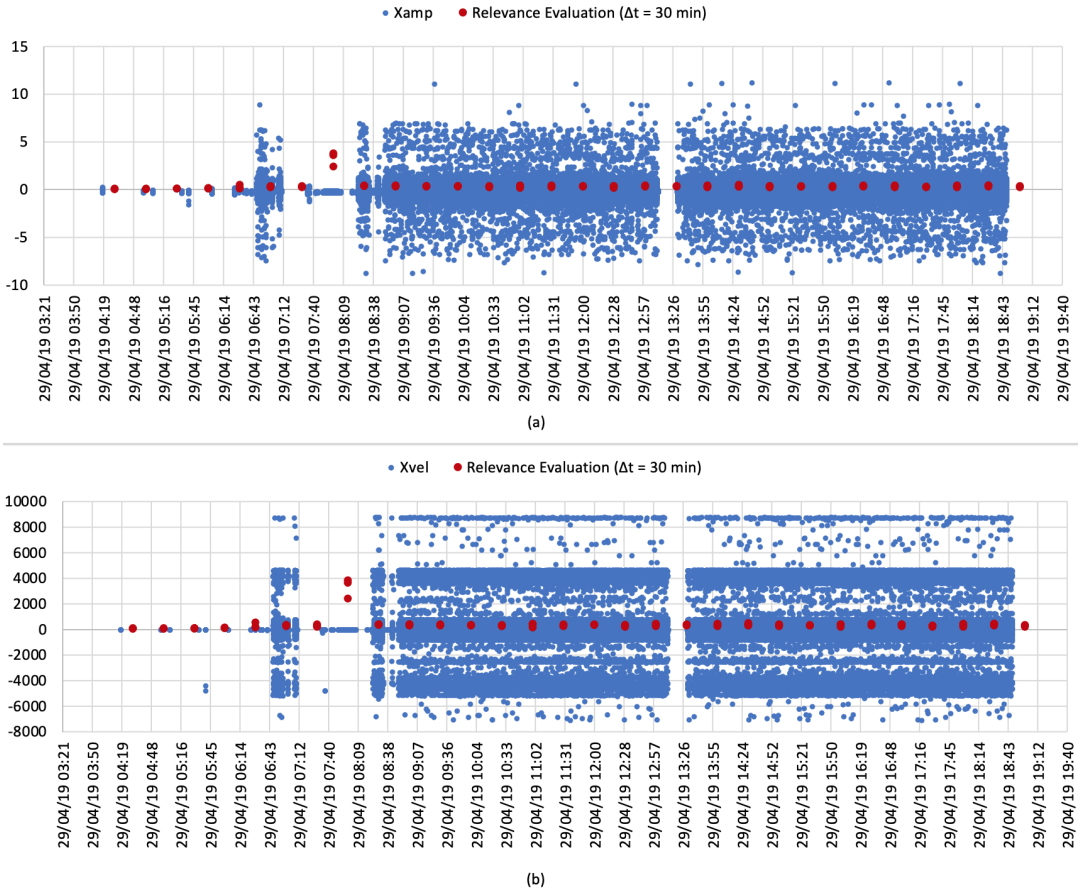


Figure 6.7: Relevance evaluation for anomaly detection applied every 30 minutes (Δt = 30 min) on feature space composed by the electrical current absorbed (a) and the velocity (b) of X axes.

In Figure 6.7 anomaly detection service has been applied every 30 minutes (Δt = 30 min). Figure shows how by enlarging the Δt value, as expected, the promptness in identifying anomalous conditions decreases and the visualisation of the cluster distance from stable working conditions is less evident with respect to Figure 6.7.

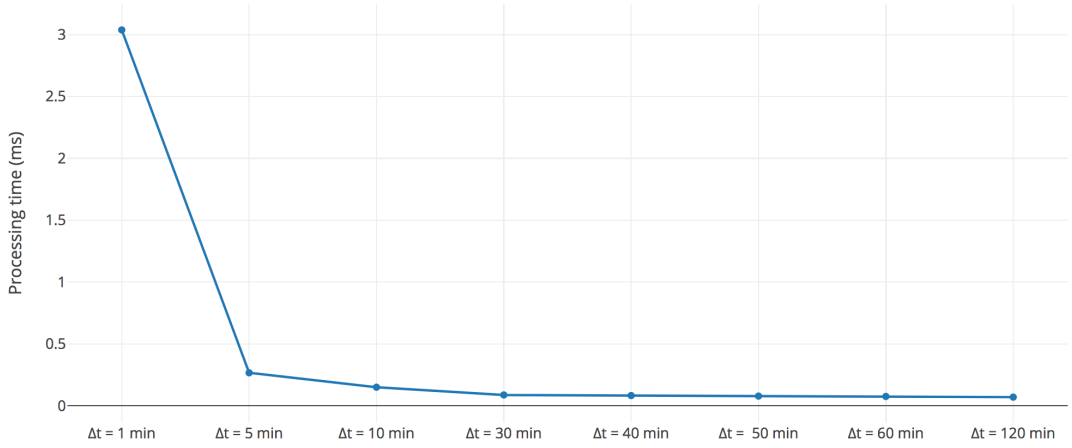


Figure 6.8: Response time for IDEAaS processing time, given by data summarisation and relevance evaluation, with respect to Δt value.

Figure 6.8 shows the average time required by the IDEAaS data summarisation and relevance evaluation to process a single record for different Δt values. In figure it is shown how lower Δt values require more time to process data. In fact, every time clustering is applied, some initialisation steps have to be performed (e.g., opening/closing connection to database, access to the set of micro-clusters previously computed).

Therefore, lower Δt values lead to more frequent execution of initialisation steps. On the other hand, higher Δt values decrease the promptness in identifying anomalous situations, as shown in Figure 6.7 (Δt = 30 min) with respect to Figure 6.6 (Δt = 5 min). Indeed, when deal with anomaly detection applications where timing is crucial to avoid losses, 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.

As a final remark, for what concerns the efficacy of the cockpit to support (maintenance) operators during data exploration, sampling techniques offer doubtless advantages to ease exploration of data through the proposed implementation of the visualisation cockpit. It is worth remarking here that visualising all the data, without adaptive sampling techniques, is not valuable for the operators and will prevent them from easy inspecting and identifying incoming anomalies.

Chapter 7

Remote monitoring services in the healthcare domain

In this chapter the adoption of the Multi-Dimensional Model, data summarisation and relevance evaluation techniques to implement remote monitoring services in the healthcare domain will be discussed. This application scenario is based on the healthcare case study introduced in 1.3.2. The scenario of the patients monitoring is reported in Figure 7.1.

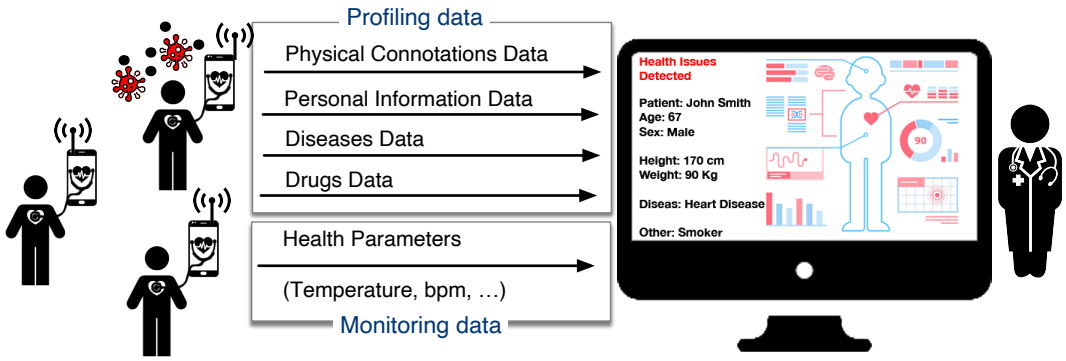


Figure 7.1: Remote monitoring of patients health parameters, with the support of smartphone applications.

In this case study, the IDEAaS system has been used within a project, where some body parameters (in particular, concerning respiratory acts) have been collected from discharged SARS-CoV-2 patients in order to remotely monitor their conditions. Traces of respiratory acts have been recorded by the three axial accelerometers, embedded by a smartphone (positioned over the abdomen of the patient). The work described in this chapter has been published in the following paper:

[64] Ada Bagozi, Devis Bianchini, Valeria De Antonellis, and Massimiliano Garda.

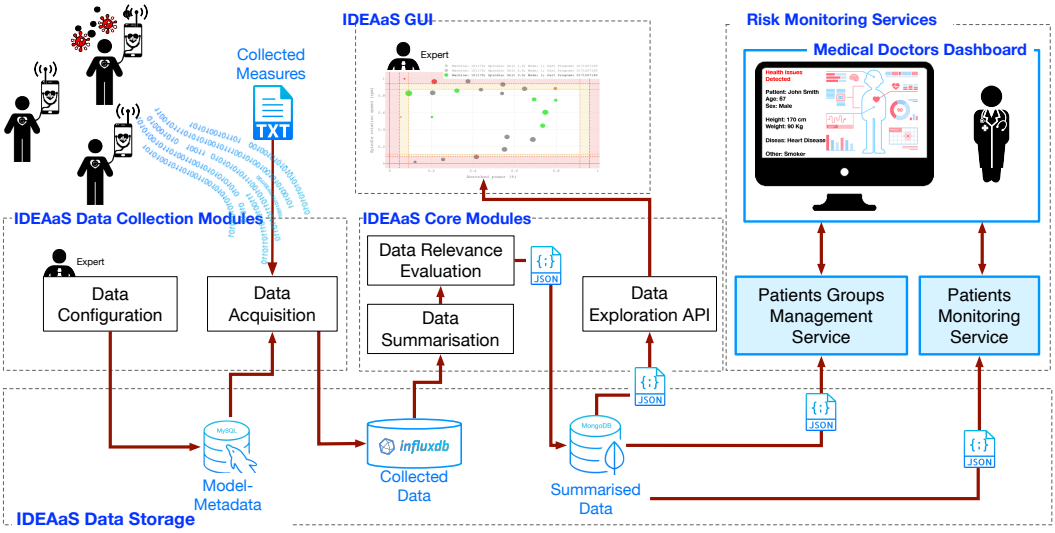


Figure 7.2: Risk Monitoring Services built on top of the IDEAaS modular architecture.

Risk Monitoring Services of Discharged SARS-COV-2 Patients. In Web Information Systems Engineering – WISE 2020, pages 578–590, 2020.

7.1 Risk monitoring services in the healthcare domain

Figure 7.2 presents the IDEAaS modular architecture extended with the introduction of *Risk Monitoring Services* in the healthcare domain. Data is collected within an InfluxDB time series database, where measures are organised according to measurement sessions and labelled with the dimensions of the Multi-Dimensional Model. With respect to the IDEAaS modular architecture explosed in Chapter 3, the storage system of single data points in this case study (i.e., InfluxDB) substituted the original technology chosen for IDEAaS (i.e., MongoDB). This further demonstrate the feasibility of modular architecture adopted for the approach. Summarised Data, obtained through incremental clustering algorithm, is stored within MongoDB database of the IDEAaS system. Measures of interest and considered analysis dimensions will be detailed in the next section. Among *Risk Monitoring Services*, *Patient Groups Management Service* based on data summarisation and relevance evaluation techniques has been introduced in order to help the doctor in identifying only relevant groups of patients, that may be more vulnerable to COVID-19. On the other hand, *Patients Monitoring Services* has been designed to monitor each patient and identify when critical health state emerges. Finally, a remote visualisation dashboard, called *Doctors GUI*, has been designed on top of the *Patient Groups Management* and *Patients*

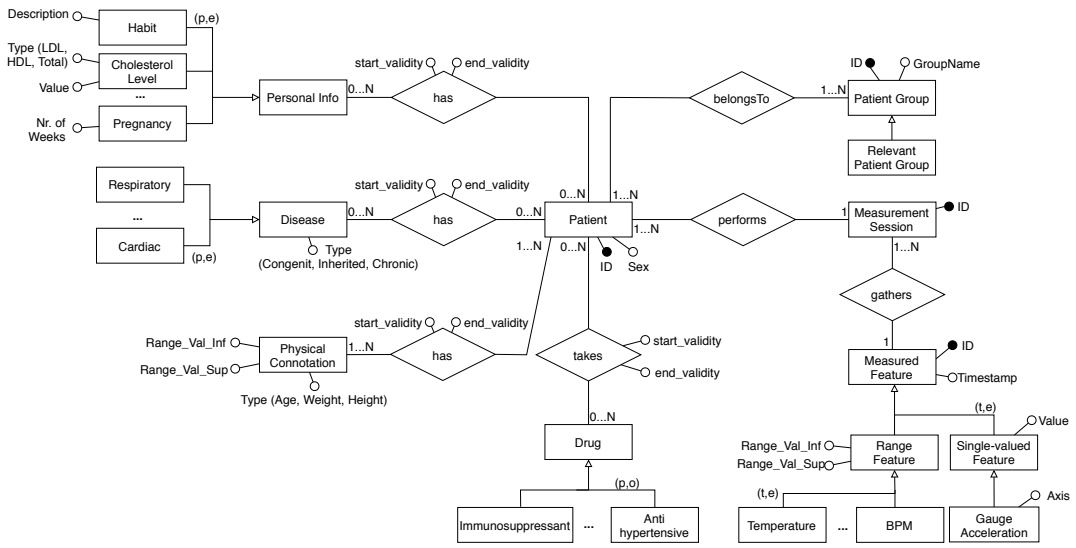


Figure 7.3: ER conceptual data model for patients' profiling and monitoring.

Monitoring services, to guide the medical doctor during critical health state discovery.

7.2 Patients' profiles and monitoring data

In the following, the measures collected by the app are detailed, modelled through the Entity-Relationship (ER) diagram (Figure 7.3). The app collects data of both patients' profiles and monitoring data (collected during measurement sessions).

Profiling data. When a patient registers to the app, the following information is collected: (i) *Physical Connotations*, concerning sex, age, weight and height; (ii) *Personal Information*, regarding habits of the patient (e.g., whether she is a smoker), possible ongoing pregnancy and cholesterol levels; (iii) *Diseases*, which are classified by their virtue (inherited, chronic, congenital) and by the part of the human body they affect; (iv) *Drugs*, assumed by patients and belonging to diverse classes, depending on their specific purpose (e.g., anti-hypertensive, immunosuppressant). Profile data enables the creation of *Patients Groups*, uniquely identified by a combination of the aforementioned data (male patients, male patients over 65 years, etc.); amongst the (potentially vast) set of possible groups, *Relevant Patients Groups* (in brief, RPG) may be recognised, that is, groups which are under the lens of medical doctors' consideration, as they are more exposed to (a relapse of) the infection risk. RPG may be

identified from well known clinical studies (e.g., male subjects are more likely of being infected by SARS-CoV-2 with respect to female ones), but also in a dynamic way, due to the emerging critical health status in several patients within the group through the application of IDEAaS relevance evaluation techniques.

Monitoring data. Through the app, the patient can perform a *Measurement Session*, which consists in a two-phase exercise aimed at assessing the quality of her breath. After the registration, the user logs in to the app to start the measurement session, articulated over two different phases: (a) an initial series of ten regulated respiratory acts (*controlled breath* phase); (b) free breathing followed by deep inspiration and forced expiration (*deep and short breath* phase). Two types of features are measured: (a) *Range Features* such as the temperature and the bpm, sampled only once before the measurement session begins, and for which the patient has to select amongst predefined ranges (defined by clinicians and domain experts) the value falls in (e.g., if the bpm value is 77, it is included in the range $[75, 80]$); (b) *Single-valued Features*, regularly sampled multiple times within a measurement session (e.g., the gravity acceleration measured along the X axis within a session consists of $\approx 43k$ samples).

Patients groups management services. Data collected from the app is explored according to different perspectives, to identify subsets of data upon which medical doctors' analysis must be focused, thus coping with the complexity and the variety of the domain (patients with different physical connotations, habits, diseases, etc.). Relevant patients groups might not be a-priori known, but they could be progressively detected through the ongoing risk monitoring procedure (performed by the *Risk Monitoring Services* described in the next section). Starting from these premises, Risk Monitoring Service are based on a Multi-Dimensional Model grounded on four pivotal elements (dimensions, facets, features and measurements), descending from the conceptualisation provided in Section 3.1 and exploited by the Patients Groups Management Service to organise patients data, enabling an intuitive, structured and effective exploration of available information. Patients groups can be conceived, in this case study, as exploration facets introduced in Chapter 3.
Given the set $\mathcal{D}$ of available dimensions, the extent of the space of patients groups (i.e., the cardinality of the set Φ, denoted as $|\Phi|$) can be very large, as it spans all the possible combinations of dimension instances (by definition, $|\Phi| \leq 2^N - 1$, where

$N = \sum_{i=1...p} |Dom(d_i)|$, excluding the empty set combination and, generally, non combinable dimension instances). For this reason, doctors' focus should be on the *relevant patients groups*, meant to emphasise only specific groups of patients, whose composing dimension instances configure the clinical picture of a patient as harmful.

In this context data summarisation have been applied in order to represent collected data characterising the health status of patients belonging to the same group, using a reduced amount of information. Additionally, relevance evaluation techniques are applied on the summarised data instead of considering single measures, that can be affected by errors and false outliers due to measurement execution performed by non-expert patients. As mentioned in Chapter 3 the clustering algorithm at a given time t produces a set of micro-clusters $\mu C(t)$, starting from measures collected from timestamp $t - \Delta t$ to timestamp t and built on top of the previous set of micro-clusters $\mu C(t - \Delta t)$, for a given patients group. Roughly speaking, micro-clusters conceptually represent a specific state in a patient's health status. A set of micro-clusters is contained within a *snapshot*. For this case scenario, snapshots have been further evolved by introducing the breath phase of the patient during data collection, and are defined as follows.

Definition 11 (Enriched Snapshot) *An enriched snapshot $SN_i^e(t)$, stored at time t, is defined as the following tuple:*

$$SN_i^e(t) = \langle \mu C(t), \rho, fs_i, \varphi_i, \pi_i \rangle \tag{7.1}$$

where: (i) $\mu C_i(t)$ is a set of micro-clusters generated at time t, (ii)$\rho : \mu C(t) \rightarrow 2^{\mu C(t-\Delta t)}$ is a mapping function that relates a micro-cluster in $\mu C(t)$ to zero, one or more micro-clusters in the set $\mu C(t - \Delta t)$ stored in the previous snapshot $SN^e(t - \Delta t)$, (iii) fs_i is the monitored feature space and (iv) φ_i is the exploration facet; (v) π_i is the breath phase (i.e., controlled breath, deep and short breath).

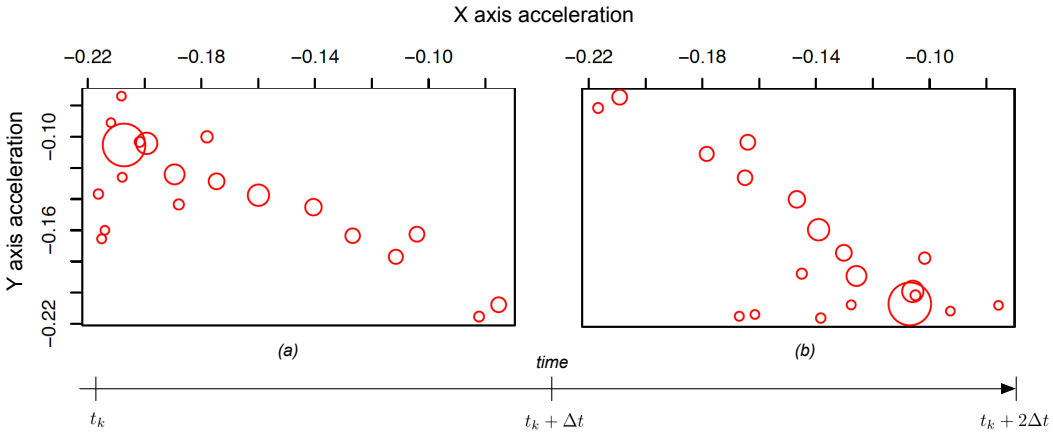


Figure 7.4: Results of incremental clustering of a stream of records reporting X and Y axes acceleration values over time for a patient in the group "Males with age between 60 and 75 years" during controlled breath phase. Micro-clusters set changes from (a) to (b) denote weariness of the patient during expiration and inspiration activities.

Figure 7.4 shows two snapshots taken at time $t_k + \Delta t$ and $t_k + 2\Delta t$, where measured features are accelerations over X and Y axes during controlled breath phase. Specifically, they are referred to a patient, belonging to a patients group gathering males, with age in the $[60, 75]$ range. Red circles represent identified micro-clusters within snapshots. A sequence of snapshots identifies a pattern, that is a behaviour related to the evolution of patient's health status. For example, in Figure 7.4(b) a change in the micro-clusters set with respect to Figure 7.4(a) is evident. Health anomaly detection techniques described in the following are used to identify such changes.

7.3 Relevance-based healthcare data exploration

Patients whose current health status is approaching an *anomalous status* have been considered as relevant. The anomalous status is expressed through a set of thresholds for each observed feature within a specific patients group and a breath phase (certified by relying on doctors' long-term expertise). Indeed, due to the fact that each measured feature is a physical quantity, it may present limits (bounds) that should not be violated. In particular, *warning* and *error* bounds have been distinguished: (i) a warning signals that values of a feature are getting closer to irreversible changes; (ii) an error identifies unacceptable values for a feature, determining health conditions in which a patient cannot withstand. Warning and error bounds in this

case are very similar to the bounds defined for anomaly detection in the smart factory (see Chapter 6).

However, also the transition towards a warning or error condition is worth being detected. To this aim, the anomaly detection mechanism described in Chapter 6 has been adopted. Let's denote with reference snapshot $SN_i^e(t_0)$ the enriched snapshot of a patient in healthy conditions after being discharged. The reference snapshot represents a baseline for all patients in normal health conditions given fs_i, φ_i and π_i. Data relevance at time t is based on the computation of *distance* between the set of micro-clusters $\mu C_i(t)$, contained in the snapshot $SN_i^e(t)$ and $\mu C_i(t_0)$, in the reference snapshot $SN_i^e(t_0)$. As mentioned before, relevance techniques allow to identify what are the micro-clusters that changed over time (namely, appeared, merged or removed) for a specific combination of fs_i, φ_i and π_i. By detecting these changes, it is possible to focus in advance the attention of medical doctors on relevant micro-clusters that are approaching anomalous conditions, also enabling the prompt identification of unusual conditions on monitored patients, where warning or error bounds have not been defined yet, solely based on the notion of *data relevance*. Indeed, groups containing at least one relevant patient have been defined as *Relevant Patients Groups*.

Patient monitoring services Exploration of relevant patients groups is performed on top of the Multi-Dimensional Model, in order for medical doctors to restrict the search space while monitoring patients' health status. Exploration is performed in two main steps: (i) firstly, health anomaly detection is used to identify relevant groups to start the exploration from; (ii) therefore, the data organisation imposed by the Multi-Dimensional Model is exploited to further guide the exploration.

How the exploration starts. In case the medical doctor is willing to focus her analysis on a specific patients group, she can directly choose a facet, drawn from the set of facets Φ. Conversely, in the case the doctor has explicit, albeit not completely defined, exploration demands, instead of indicating a single exploration facet as before, she may specify a set d^r of desired dimension instances for the dimensions she is interested in, where $d^r = \{v_{d_1^r}, \ldots, v_{d_p^r}\}$ and $v_{d_i^r} \in Dom(d_i^r)$. Let's denote with $\Phi^r \subseteq \Phi$ the set of corresponding patients groups. In both cases, the doctor can be supported in the selection by proposing her the patients groups in Φ (resp., Φ^r) identified as

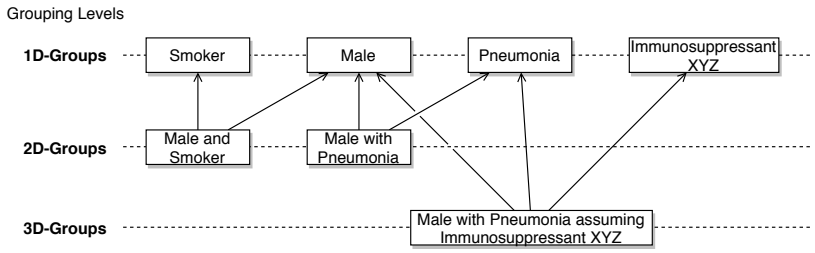


Figure 7.5: Examples of facets (patients groups) with increasing grouping levels.

relevant. Moreover, relevant patients groups are ranked considering the percentage of patients presenting anomalies inside each group; in this respect, medical doctor's attention will be attracted towards those groups in Φ (resp., Φ^r), ranked with higher percentage values.

How the exploration goes on. Let $\Phi_{rel} \subseteq \Phi$ (resp., $\Phi_{rel} \subseteq \Phi^r$) be the set of relevant patients groups. The doctor is guided by the MDM in order to explore the groups in Φ_{rel}. According to Figure 7.5, relevant patients groups at highest grouping levels (e.g., "*males* with *pneumonia* assuming *immunosuppressant*") are proposed first. Starting from them, the doctor may split facets into composing dimensions, moving towards lower grouping levels. For example, starting from the patients group mentioned above, the doctor may inspect the percentage of relevant patients among *males*, among those affected by *pneumonia* and among those assuming *immunosuppressant*. This may help identifying facets and dimensions that are correlated the most with SARS-CoV-2 episodes, thus further increasing the knowledge on this pandemic phenomenon.

Once a relevant patients group has been selected in the set Φ_{rel}, the medical doctor may continue the exploration by adopting different strategies. On the one hand, she may decide to focus her attention on a specific patient, trying to diagnose the event that led to the anomalous situation (for instance, warning detected on features related to the acceleration over the three axes may be a symptom of shortness of breath). On the other hand, the doctor may carry out a comparative analysis over different patients of the same group, devoted to discover why anomalies detected in a single patient are somehow recurring in other patients of the group (for instance, due to a genetic defect shared by patients).

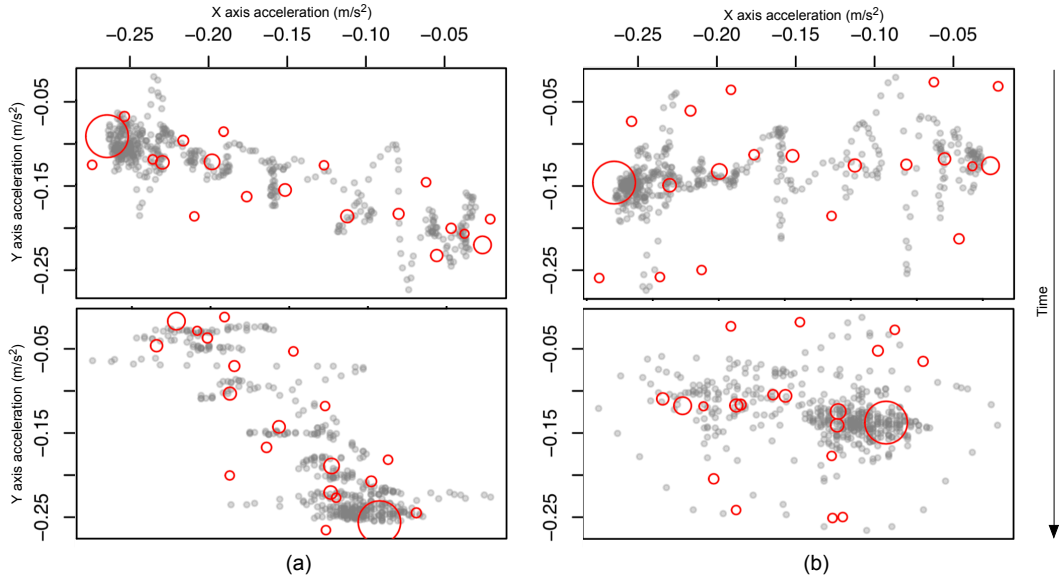


Figure 7.6: Results of incremental clustering on discharged SARS-CoV-2 patients for controlled breath (a) and deep and short breath (b) phases.

7.4 Experimental Evaluation

Experiments are being conducted in order to assess: (i) the quality of data relevance evaluation techniques in the healthcare domain; (ii) processing time to verify whether summarisation and relevance evaluation techniques could face variable data acquisition rates and (iii) effectiveness of the data exploration techniques to properly attract the attention of medical doctors on relevant patients groups. Preliminary results demonstrated that relevance-based anomaly detection techniques were capable of detecting variations (either gradual or sharp) in incoming data. Moreover, data summarisation and relevance-based anomaly detection could be efficiently carried out due to a thorough storage environment, ensuring processing time meeting high acquisition rates. Remarkably, relevance-based anomaly detection was particularly apt to ease data exploration, by identifying relevant snapshots, labelled with the facets of the Multi-Dimensional Model. Figure 7.6 shows the results of incremental clustering on a male patient data, with age in the $[60, 75]$ range, on which worsening respiratory conditions have been detected. Visually, computed micro-clusters (red circles) are able to better detect a movement in acceleration values on the X axis, with respect to raw data (grey points), that are affected by variations and noise. As mentioned in the previous use case application, Δt plays an important role in the promptness of the approach in detecting anomalies. However, it is

possible to set Δt to an higher value, in order to reduce the required computational resources, when dealing with an healthcare scenario, similar to the one considered in this application, namely where: (i) the frequency of collected data is not high for each patient; (ii) the number of patients may be large; (iii) patients health state changes can be considered slow over time. In this specific application scenario, for example, Δt may be set to one hour or at time intervals that are comparable with hourly granularity. The validation of impact of the Multi-Dimensional Model on exploration will continue through an intense patients recruitment and testing. Recruitment will involve an initial sample of 50-100 discharged patients aged 18-75 years, 50% males with a SARS-CoV-2 diagnosis. App questionnaires for collecting profiling data from enrolled patients and patients inclusion/exclusion criteria are being defined. Exclusion criteria concern conditions affecting the capacity of the patient to provide informed consent and to use the app (e.g., physical and intellectual disability, dementia, current delusions or hallucinations). Selected patients will suffer of different kinds of respiratory and cardiac co-morbidities and assume different drugs such as immunosuppressant and anti-hypertensive therapies. For each group, a percentage of patients will be selected among smokers, as risk factor for SARS-CoV-2 episodes. GDPR procedures will be specified, in particular for data and contact tracing in remote medicine. Finally, usability experiments on the monitoring dashboard will be performed with the collaboration of different categories of medical doctors (e.g., general practitioners, medical researchers).

Chapter 8

Context-based resilience in the Smart Factory

In this chapter the adoption of the IDEAaS system is discussed to implement resilience based on data streams, collected in a Smart Factory. This application scenario is based on the context based resilience in connected Smart Factories case study introduced in 1.3.3 on the production process for the food industry reported here in Figure 8.1.

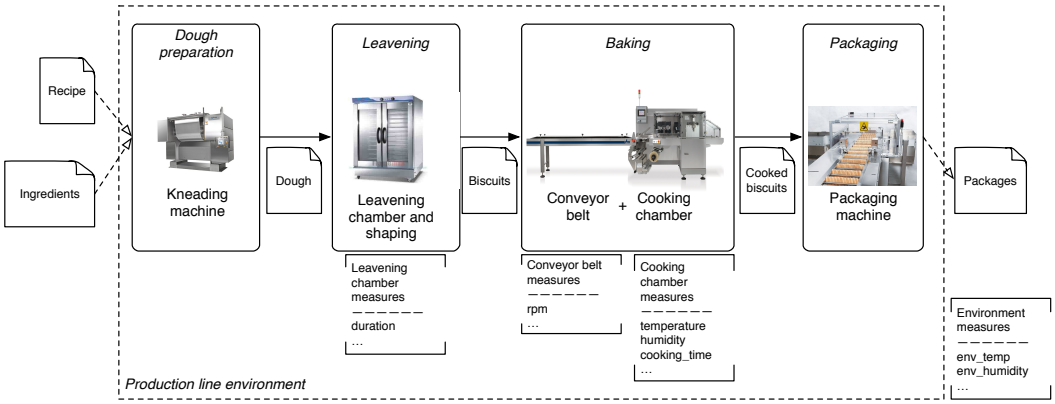


Figure 8.1: Production process for the food industry case study.

The work described in this chapter has been published in the following paper:

[65] Ada Bagozi, Devis Bianchini, and Valeria De Antonellis. Designing Context-Based Services for Resilient Cyber Physical Production Systems. In Web Information Systems Engineering – WISE 2020, pages 474–488, 2020.

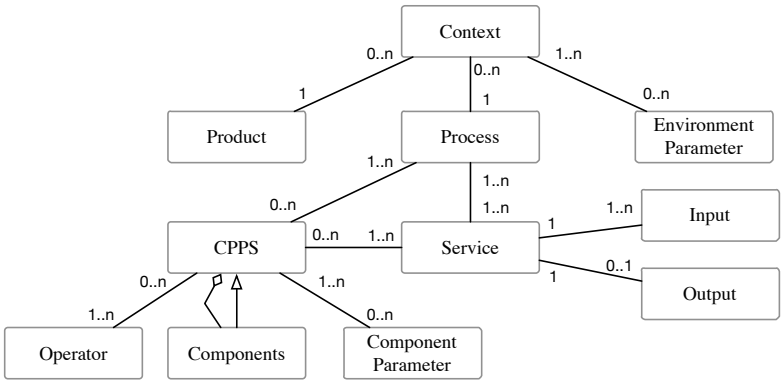


Figure 8.2: Resilient CPPS context model.

8.1 Context Model

Figure 8.2 reports the context model adopted to implement the context-based approach to resilience in the smart factory as described in this chapter. In the model, the *Context* is described by the *Product* that is being produced (e.g., a certain type of biscuits), the *Process* to produce a certain product (e.g., biscuits baking process) and the *Environment Parameters* that may influence the production (e.g., the environment temperature and humidity). A *Process* is executed by one or more *components*, that are used to successfully complete the production. For example, the biscuits baking process includes the kneading machine to prepare the dough, the leavening chamber to prepare biscuits, the oven to bake the biscuits, and so on. Components can be organised hierarchically, according to the RAMI 4.0 reference architectural model [8] (IEC62264/IEC61512 standards) for the smart factory. For example, the oven is composed of the conveyor belt and the cooking chamber.A component is supervised by at least one *Operator* and can be monitored and controlled through a set of *Component Parameters* (e.g., the oven temperature). Furthermore, a production process is associated to one or more Services, that represent recovery actions that can be executed on a component or on the entire production line to ensure resilience. Below some examples of recovery services are given.

Both *Environment Parameters* and *Component Parameters* are used to monitor the behaviour of a *CPPS* or of the entire production line in a given *Context*. Indeed, parameters are used to observe physical phenomena of the monitored system and are formally defined as follows. They corresponds to the measured features in the formula definition of the IDEAaS system and are treated in the same way, being also

associated with error and warning bounds as in the anomaly detection case studies.

During parameter monitoring, if an anomaly is detected, recovery actions are required and performed by invoking *Services*. Indeed, according to the model, for a *CPPS* it is possible to define one or more, possibly alternative, *Services*. A recovery service is formally defined as follows.

Definition 12 (Recovery service) *A recovery service S_j associated to a CPPS (or one of its components) is described as a tuple*

$$S_j = \langle n_{S_j}, IN_{S_j}, out_{S_j}, type_{S_j}, CPPS_{S_j} \rangle \tag{8.1}$$

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 an optional service output; (iv) $type_{S_j}$ is the service type; (v) $CPPS_{S_j}$ is the component or the whole production line to which the service is associated. Service I/O can be either Component or Environment Parameters. Let's denote with $\mathbf{S} = \{S_1, S_2 \dots S_n\}$ the overall set of recovery services.

Flexibility of service-oriented architectures enables to include and dynamically add different types of services. For instance, a recovery service may implement the function that relates one or more input parameters with the output one. This type of service will be referred as "re-configuration". For example, the following service

`setOvenTemperature(ConveyorBelt.rpm)` $\rightarrow$ `CookingChamber.temperature`

is a re-configuration service to set the cooking chamber temperature when the conveyor belt rpm changes, to avoid cookies overheating. When a re-configuration is not an applicable solution (e.g., if the service returns a cooking chamber temperature out of an acceptable range of values), other recovery actions must be applied, such as to replace or repair the conveyor belt. An example of "component substitution" service would be the following:

`replaceConveyorBeltRotatingEngine(ConveyorBelt.rpm)` $\rightarrow$ `void`

that has no output parameter to modify. This service is associated to the conveyor belt. The function implemented within the re-configuration services, as well as other service information (e.g., execution cost, time) that can be used to choose among different kinds of services, are based on the knowledge of the domain. The examples of recovery service types considered here is not exhaustive and may be extended [66].

8.2 Context-based resilience

In this section, each phase of the proposed context-driven approach to support the on-field operators in the identification of critical conditions and in the runtime selection of services that implement proper recovery actions will be presented. While presenting each phase, the possible involvement of the IDEAaS system will be discussed.

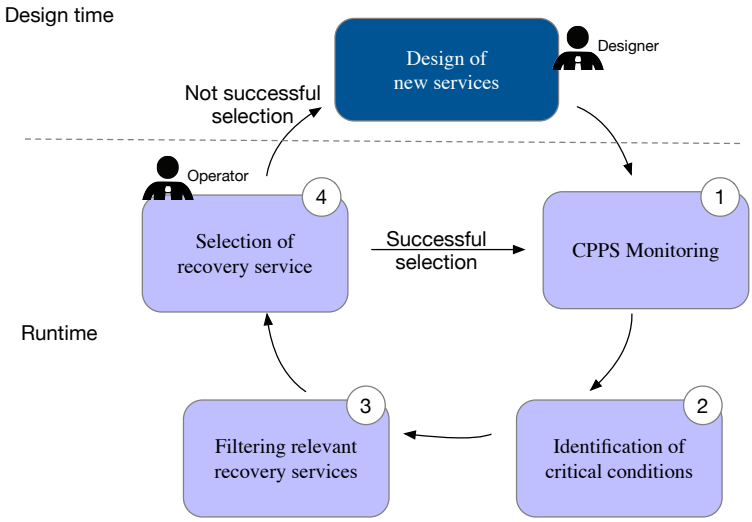


Figure 8.3: Five context-driven monitoring and recovery phases to design resilient Cyber Physical Production Systems.

Figure 8.3 reports the five phases of the proposed approach. As a pre-condition, the designer, who has the domain knowledge about the production plant, is in charge of preparing the context model described in the previous section, as well as an initial set of recovery services. At runtime, the CPPS is monitored by relying on IDEAaS (Phase 1). Critical working conditions are detected by inspecting data streams collected from monitored CPPS. Anomalies are propagated over the hierarchy of connected CPPS (Phase 2). When critical conditions are detected, the relevant recovery services are identified (Phase 3). Service filtering takes into account the involved component, on which anomalies have been detected, and parameters whose values exceed warning or error bounds. Selected recovery services are suggested to the on-field operators working on the involved component (Phase 4). If no recovery services have been found, a feedback is stored for planning the design of additional recovery services in the future. The filtering and selection phases of relevant recovery services will be detailed in Section 8.2.

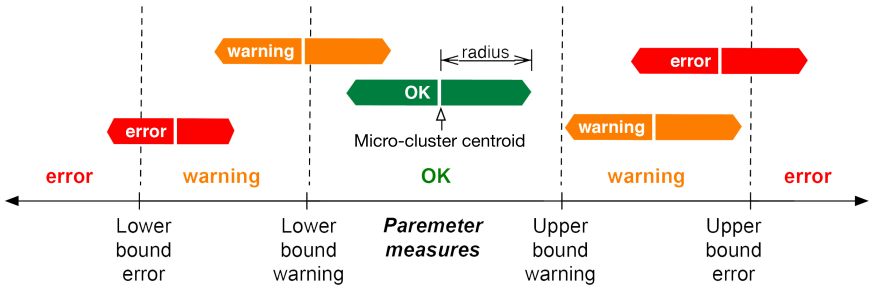


Figure 8.4: Warning and error bounds for parameters measurements in order to detect the system status.

Identification of critical conditions. The goal of the anomaly detection is to identify critical conditions and send alerts concerning the *system status*. Three different values for the status are considered: *(a)* `ok`, when the system works normally as established by domain experts according to their expertise; *(b)* `warning`, when the system works in anomalous conditions that may lead to breakdown or damage (proactive anomaly detection); *(c)* `error`, when the system works in unacceptable conditions or does not operate (reactive anomaly detection). Therefore, the warning status is used to perform an early detection of a potential deviation towards an error status. The identification of the system status is based on the comparison between parameters measurements and `warning`/`error` bounds.

As mentioned before anomaly detection is performed over a summarised representation of collected measures, incrementally built. Every Δt seconds, IDEAaS incremental clustering is applied and, micro-clusters are compared against bounds as shown in Figure 8.4, in order to classify the status of a CPPS, starting from the status of its monitored parameters. The following options are considered for the status of each CPPS:

- `ok`, if the status for all its parameters is `ok`;
- `warning`, if the status of at least one parameter is `warning`;
- `error`, if the status of at least one parameter is `error`.

Once the status of a single component has been established, it is propagated over the hierarchy of connected components as follows:

- `ok`, if the status for all its components is `ok`;

- `warning`, if the status of at least one component is `warning`;
- `error`, if the status of at least one component is `error`.

When the status of a component, after applying the propagation rules, changes towards `warning` or `error`, an alert is raised by the system.

Filtering and selection of relevant recovery services. Once an anomalous event (corresponding to a critical condition) is detected on one of the components, the event is used to identify relevant services that implement recovery actions on the involved component or connected ones. Relevant recovery services are identified by inspecting their inputs. In particular, a recovery service is candidate to be identified as relevant if one of its input parameters have been classified in the `error` (reactive resilience) or `warning` status (proactive resilience). On the other hand, before including the service among relevant ones, the candidate is checked to verify the following conditions: *(a)* if the service type is "re-configuration", the value of its output parameters must not exceed any parameter bound; *(b)* if the service type is "component substitution", the component associated to the service should have an alternative machinery or component ready to be used in substitution of the one affected by the anomaly. For all the other options (e.g., repair), ad-hoc procedure must be engaged. Formally, a relevant recovery service is defined as follows.

Definition 13 (Relevant recovery service) *A recovery service $S_k \in \mathbf{S}$ is defined as relevant if:*

- *the status of at least one input parameter in IN_{S_k} has been set to* `warning` *or* `error`*;*
- *if $type_{S_k}$=*`re-configuration`*, then the value of the output parameter out_{S_k} resulting from service execution must fall inside the admissible parameter bounds;*
- *if $type_{S_k}$=*`component_substitution`*, then there must exist a component ready to be substituted to the $CPPS_{S_k}$ associated to S_k.*

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 compared against parameter bounds of the cooking chamber temperature.

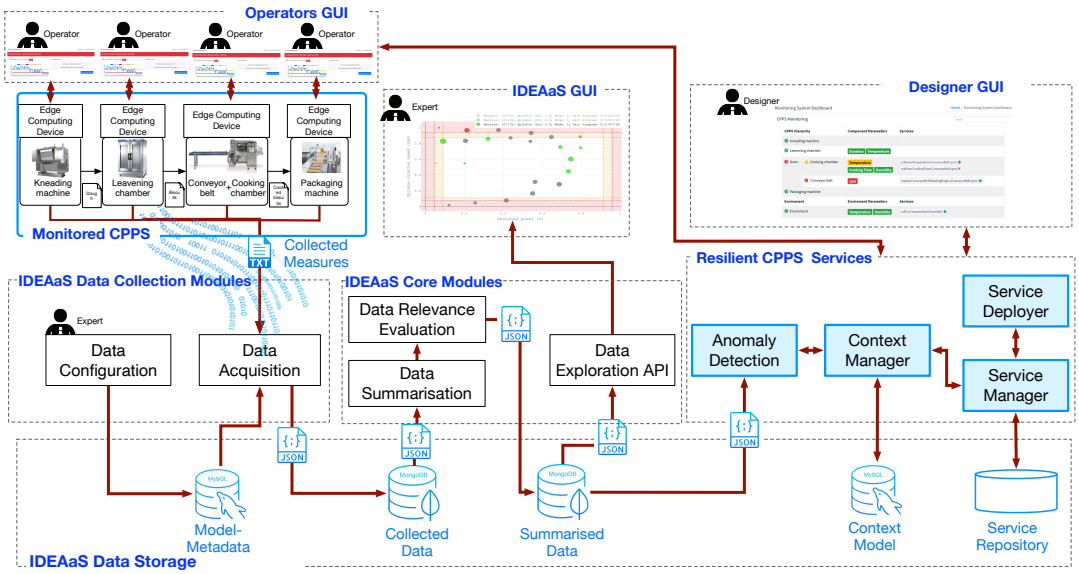


Figure 8.5: Context-driven resilience built on top of the IDEAaS modular architecture.

Once relevant recovery services have been identified, they are suggested to the operator assigned to the $CPPS_{S_k}$ in order to be confirmed and executed on the component or the whole production line (for re-configuration services) or to proceed with a substitution of the affected part (for component 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. Another feature of this implementation of the case study 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 CPPS component and visualised on the edge computing device. 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.

8.3 Implementation and validation

Case study architecture. The approach described in this chapter has been integrated with the anomaly detection module and the resulting architecture is presented in Figure 8.5. 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, either *Environment or Component parameters*, to be analysed; *(iii)* the observed CPPS; *(iv)* the product that is being produced; *(v)* the running process. Such information is extracted from the *Context Model* database.

Collected measurements of the parameters in the context are properly summarised as micro-clusters by applying IDEAaS data summarisation techniques and micro-clusters are labelled applying relevance evaluation techniques.

Labelled summarised data are visualised: *(a)* on the `Designer GUI` to let the designer monitor the overall evolution of the system; *(b)* on the `Edge Computing Device` of the involved component, to let the on-field operator to better understand the behaviour of the component. Moreover, when anomalous conditions are detected, the `Context Manager` is notified with the identifier of the context and the list of critical parameters on which the anomaly occurred, together with their measurements. The `Context Manager` will search for relevant recovery services, associated to the component in the context. Once relevant recovery services have been identified, the `Context Manager` launches the execution of the services by interacting with the `Service Manager`, which is responsible for services registration in the *Service Repository* and for their execution. The result of the services execution is sent to the operator assigned to the component, in order to let the operator choose the most suitable service.

On the other hand, when the identification of relevant recovery services fails, the `Context Manager` reports the unsuccessful service selection to the designer, thought 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 between the `Service Deployer` and the `Service Manager`. The `Context Manager` is notified as well, in order to update the *Context model*.

Validation of the approach. In this section proof-of-concept validation is performed in the case study to demonstrate its applicability. In particular, the focus wil be on: *(i)* processing time required to promptly detect anomalies and activate recovery actions services (being this aspect a potential bottleneck for the whole approach); *(ii)* a proposal of dashboards to be used in real case studies in order to show the

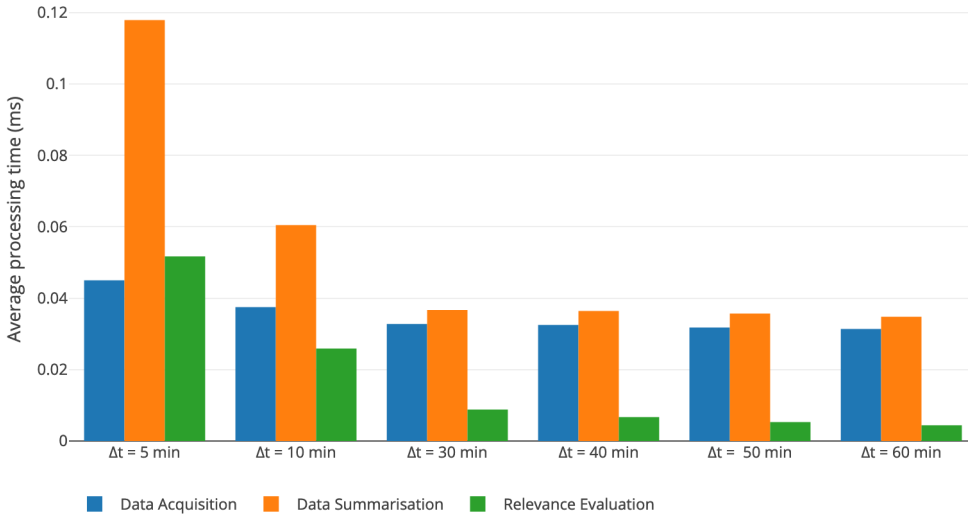
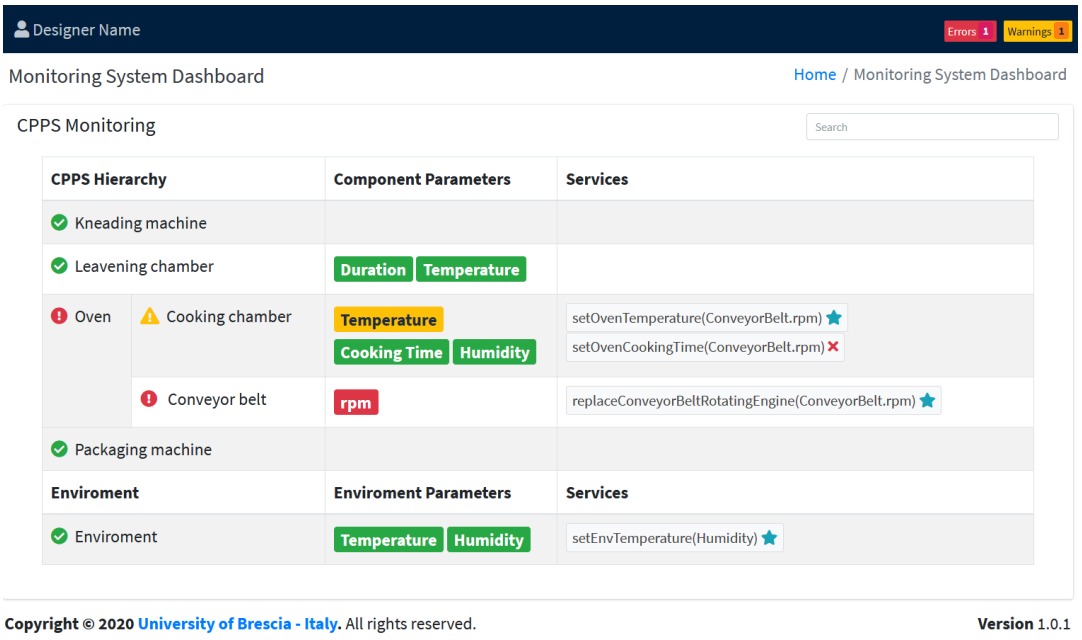


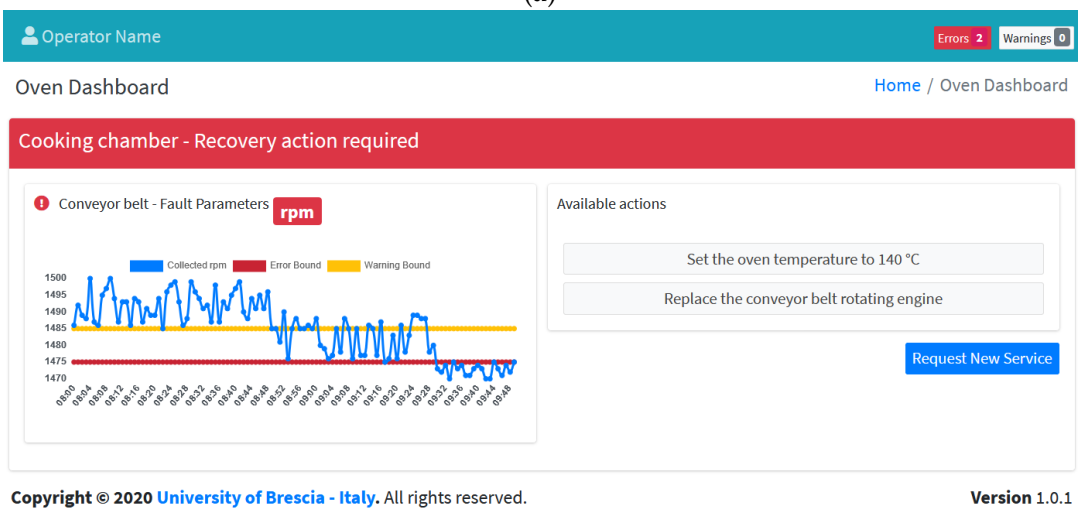
Figure 8.6: Average response times per measure for anomaly detection that activates recovery activation services.

effectiveness of the context model for structuring and filtering the right information displayed on the right location within the digital factory.

Figure 8.6 reports the average time required for data acquisition, summarisation and relevance evaluation. Experiments have been run on a MacBook Pro Retina, with an Intel Core i7-6700HQ processor, at 2.60 GHz, 4 cores, RAM 16GB. Figure 8.6 reports the average response time for each collected measure, with respect to the Δt interval in which data summarisation and relevance evaluation are performed. In a counter-intuitive way, as mentioned before, 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 of the algorithm iteration (i.e., every Δt seconds), a certain amount of time is required for some recurring operations such as opening/closing connection to database and retrieval of the set of micro-clusters previously computed. Therefore, enlarging 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, as mentioned in the anomaly detection for smart factory case study, Δt should be as lower as possible, in order to detect near real-time anomalies and to promptly implement recovery actions in order to reduce losses. According to these conditions, the capability to detect anomalies on the collected measures has been quantified using the Pearson Correlation Coefficient (PCC) $\in [-1, +1]$, that estimates the correlation between the real variations and the



(a)



(b)

Figure 8.7: Dashboards for context-driven monitoring and recovery phases: (a) designer view; (b) operator view.

detected ones. In the experiment, the best PCC value is higher than 0.85 in the case $\Delta t = 10$ minutes, that represents a strong correlation. Further evaluation is being performed.

To demonstrate the effectiveness of the context model for structuring and filtering information in case of faults, proof-of-concepts dashboards have been designed, one for the designer (Monitored System Dashboard, MSD) and one for the operator who supervises the production line or one of its components. Figure 8.7 reports

the MSD *(a)* and the Oven Dashboard *(b)*. Let's consider now the following scenario. Rpm values that exceed the error bounds have been detected on the rotating engine of the conveyor belt. To recover from the anomaly, three services have been selected as relevant ones, namely `setOvenTemperature`, `setOvenCookingTime` and `replaceConveyorBeltRotatingEngine`. However, `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. On the MSD it is visualised the hierarchy of the CPPS and, for each component, the parameters and the available services, according to the model presented in this chapter. The parameters measurements that exceed their error bounds are highlighted in red, the ones that exceed their 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 Figure 8.7(b), 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 refuse 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.

Chapter 9

Concluding Remarks

This thesis has been focused on methods and techniques to deal with Big Data Exploration (BDE). To this aim, the IDEAaS (Interactive Data Exploration As-a-Service) approach has been designed and developed. The approach, specifically conceived for big data streams, relies on the following novel techniques, properly combined to realise BDE under a Human-In-the-Loop vision:

- an incremental clustering algorithm, that aggregates, in the so-called *micro-clusters*, data collected as streams of numeric features; micro-clusters represent a working behaviour of the monitored system, to provide summarised representation of data streams; micro-clusters have been in turn organised into *snapshots* to enable exploration of different portions of the data streams;
- multi-dimensional organisation of summarised data, to allow data exploration according to different analysis dimensions;
- data relevance evaluation techniques, to support the identification of micro-clusters and snapshots that correspond to unexpected behaviours (relevant data) and to attract the experts' attention on them during exploration.

Furthermore, considering that the incremental clustering algorithm is the most time-consuming element of the approach, a parallel version of the algorithm has been designed and implemented. Indeed, the parallel implementation of the algorithm has been named P-IDEAaS. In this version, the Multi-Dimensional Model and data relevance evaluation have been exploited for enhancing parallel clustering of massive data streams. Novel aspects of the clustering parallelisation are summarised in the following:

- the adoption of the multi-dimensional model to perform a first, coarse-grained partition of data streams, according to a *divide-and-conquer* strategy, to face their complexity;
- the combined application of other fine-grained levels of parallelisation: (i) a parallelisation based on a buffering mechanism, that splits the data stream into portions of data points on which processing is performed in parallel; (ii) a parallelisation over the set of micro-clusters that are generated and change over time;
- the exploitation of data relevance evaluation techniques to ensure different priority to parallelisation levels, in order to dedicate more resources for parallelisation in those cases that present higher priority (i.e., higher data relevance); this also mitigated the overload due to the distribution of processing tasks over the network of computation nodes, which might have a negative impact on algorithm efficiency, when it is not strictly necessary.

Finally, the efficiency and effectiveness of the IDEAaS approach have been tested through its application on different case studies, ranging from the Industry 4.0 to the healthcare domain. Specifically, the following real scenarios have been considered: (i) anomaly detection case study on multi-spindle machines working on metal raw material in the Industry 4.0 domain; (ii) remote monitoring services in the healthcare domain on SARS-CoV-2 discharged patients; (iii) context-based resilience over the entire production line in the connected factory (food industry domain), in the Smart Factory domain.

9.1 Future Work

Future development efforts will be devoted to investigate the possibility to balance the distribution of IDEAaS functionalities between the cloud and the edge computing: Human-In-the-Loop Data Analytics results enabled by current IDEAaS core components will be used to extract anomaly detection rules, to be applied at real time on the data streams and executed on the edge side. Therefore, security issues will be considered and addressed as well. Moreover, further tests will be also performed to consolidate the data exploration GUI, extending its functionality with the involvement of a larger group of experts in different application domains to assess its

effectiveness in supporting data exploration. Currently, the proposed GUI is mainly focused on easing data exploration through the summarisation of collected measures and their organisation in the Multi-Dimensional Model, as well as their pruning according to the relevance-based evaluation. Future effort will be addressed to improve the GUI to proper intercept experts' feedback, further refining the relevance evaluation techniques.

List of Figures

List of Tables

Bibliography

[1] M. Buoncristiano, G. Mecca, E. Quintarelli, Roveri, D. Santoro, and L. Tanca. Database Challenges for Exploratory Computing. *SIGMOD Record*, 44(2):17–22, 2015.

[2] J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publisher, 2006.

[3] M.M. Gaber, A. Zaslavsky, and S. Krishnaswamy. Data stream mining. In O. Maimon and L. Rokach, editors, *Data mining and knowledge discovery handbook*, pages 759–787. Springer, Berlin, 2009.

[4] B.R. Prasad and S. Agarwal. Data stream mining: platforms, algorithms, performance evaluators and research trends. *International Journal of Database Theory Applications*, 9(9):201–218, 2016.

[5] AnHai Doan. Human-in-the-loop data analysis: A personal perspective. In *Proceedings of the Workshop on Human-In-the-Loop Data Analytics*, HILDA'18, New York, NY, USA, 2018. Association for Computing Machinery.

[6] C. Aggarwal, J. Han, J. Wang, and P. Yu. A framework for clustering evolving data streams. In *Proc. of 29th International Conference on Very Large Data Bases (VLDB 2003)*, pages 81–92, 2003.

[7] L. Monostori. Cyber-physical production systems: Roots, expectations and R&D challenges. In *Proc. of the 47th CIRP Conference on Manufacturing Systems*, pages 9–13, 2014.

[8] M. Hankel and B. Rexroth. The Reference Architectural Model Industrie 4.0 (RAMI 4.0). In *ZVEI*, 2015.

[9] Victor Chang. Towards data analysis for weather cloud computing. *Knowledge-Based Systems*, 127:29 – 45, 2017.

[10] A novel big data analytics and intelligent technique to predict driver's intent. *Computers in Industry*, 99:226 – 240, 2018.

[11] Chang Wang, Yongxin Zhu, Weiwei Shi, Victor Chang, P. Vijayakumar, Bin Liu, Yishu Mao, Jiabao Wang, and Yiping Fan. A dependable time series analytic framework for cyber-physical systems of iot-based smart grid. *ACM Trans. Cyber-Phys. Syst.*, 3(1), August 2018.

[12] Laurel Orr, Dan Suciu, and Magdalena Balazinska. Probabilistic database summarization for interactive data exploration. *PVLDB*, 10(10):1154–1165, 2017.

[13] Seungwoo Jeon, Bonghee Hong, and Victor Chang. Pattern graph tracking-based stock price prediction using big data. *Future Generation Computer Systems*, 80:171 – 187, 2018.

[14] Lizhe Wang, Yan Ma, Jining Yan, Victor Chang, and Albert Y. Zomaya. pipscloud: High performance cloud computing for remote sensing big data management and processing. *Future Generation Computer Systems*, 78:353 – 368, 2018.

[15] Niranjan Kamat and Arnab Nandi. A session-based approach to fast-but-approximate interactive data cube exploration. *ACM Trans. Knowl. Discov. Data*, 12(1):9:1–9:26, February 2018.

[16] C. Costa, A. Charalampous, A. Konstantinidis, D. Zeinalipour-Yazti, and M. F. Mokbel. Decaying telco big data with data postdiction. In *Proc. 19th IEEE Int. Conf. Mobile Data Management (MDM)*, pages 106–115, June 2018.

[17] A. Kalinin, U. Cetintemel, and S. Zdonik. Interactive data exploration using semantic windows. In *Proc. of the ACM SIGMOD International Conference on Management of Data*, pages 505–516, 2014.

[18] K. Dimitriadou, O. Papaemmanouil, and Y. Diao. Aide: An active learning-based approach for interactive data exploration. *IEEE Transactions on Knowledge and Data Engineering*, 28(11):2842–2856, Nov 2016.

[19] B. Saket, H. Kim, E. T. Brown, and A. Endert. Visualization by demonstration: An interaction paradigm for visual data exploration. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):331–340, Jan 2017.

[20] Joris Sansen, Gaëlle Richer, Timothée Jourde, Frédéric Lalanne, David Auber, and Romain Bourqui. Visual exploration of large multidimensional data using parallel coordinates on big data infrastructure. In *Informatics*, volume 4, page 21. Multidisciplinary Digital Publishing Institute, 2017.

[21] C. A. L. Pahins, S. A. Stephens, C. Scheidegger, and J. L. D. Comba. Hashedcubes: Simple, low memory, real-time visual exploration of big data. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):671–680, January 2017.

[22] L. Stojanovic, M. Dinic, N. Stojanovic, and A. Stojadinovic. Big data-driven anomaly detection in industry (4.0): An approach and a case study. In *2016 IEEE International Conference on Big Data (Big Data)*, pages 1647–1652, Dec 2016.

[23] Chunyong Yin, Sun Zhang, Zhichao Yin, and Jin Wang. Anomaly detection model based on data stream clustering. *Cluster Computing*, page 1, August 2017.

[24] Carla Sauvanaud, Guthemberg Silvestre, Mohamed Kaâniche, and Karama Kanoun. Data stream clustering for online anomaly detection in cloud applications. In *Dependable Computing Conference (EDCC), 2015 Eleventh European*, pages 120–131. IEEE, 2015.

[25] Z. Dafir, Y. Lamari, and S.C. Slaoui. A survey on parallel clustering algorithms for Big Data. *Artificial Intelligence Review*, 2020.

[26] A. Hadian and S. Shahrivari. High performance parallel k-means clustering for disk-resident datasets on multi-core cpus. *Journal of Super Computing*, 69(2):845–863, 2014.

[27] S. Cuomo, V. De Angelis, G. Farina, L. Marcellino, and G. Toraldo. A gpu-accelerated parallel k-means algorithm. *Computers & Electrical Engineering*, 75:262–274, 2017.

[28] F. Jia, C. Wang, X. Li, and X. Zhou. SAKMA: specialized FPGA-based accelerator architecture for data-intensive k-means algorithms. In *Algorithms and architectures for parallel processing*, pages 106–119, 2015.

[29] A. Banharnsakun. A mapreduce-based artificial bee colony for large-scale data clustering. *Pattern Recognition Letters*, 93:78–84, 2017.

[30] R. Liu, X. Li, L. Du, S. Zhi, and M. Wei. Parallel implementation of density peaks clustering algorithm based on Spark. *Procedia Computer Science*, 107:442–447, 2017.

[31] R. Azimi, H. Sajedi, and M. Ghayekhloo. A distributed data clustering algorithm in P2P networks. *Applied Soft Computing*, 51:147–167, 2017.

[32] P. Huang, X. Li, and B. Yuan. A Parallel GPU-Based Approach to Clustering Very Fast Data Streams. In *Proceedings of the 24th ACM International Conference on Information and Knowledge Management (CIKM)*, pages 23–32, 2015.

[33] J. Dong, F. Wang, and B. Yuan. Accelerating BIRCH for clustering large scale streaming data using CUDA dynamic parallelism. In *Proceedings of International Conference on Intelligent Data Engineering and Automated Learning*, pages 409–416, 2013.

[34] A. Alazeez, S. Jassim, and H. Du. TPICDS: A Two-Phase Parallel Approach for Incremental Clustering of Data Streams. In *Proc. of Workshops of European Conference on Parallel Processing (Euro-Par 2018)*, pages 5–16, 2018.

[35] P. Karunaratne, S. Karunasekera, and A. Harwood. Distributed stream clustering using micro-clusters on Apache Storm. *Journal of Parallel and Distributed Computing*, 108:74–84, 2017.

[36] O. Backhoff and E. Ntoutsi. Scalable Online-Offline Stream Clustering in Apache Spark. In *Proceedings of IEEE 16th International Conference on Data Mining Workshops (ICDMW)*, pages 37–44, 2016.

[37] X. Wang and Q. Sun. Research on Clustream Algorithm Based on Spark. In *Proceedings of 10th International Symposium on Computational Intelligence and Design (ISCID)*, pages 219–222, 2017.

[38] A. Alazeez, S. Jassim, and H. Du. EINCKM: An Enhanced Prototype-based Method for Clustering Evolving Data Streams in Big Data. In *Proc. of 6th International Conference on Pattern Recognition Applications and Methods*, pages 173–183, 2017.

[39] G. Morales and A. Bifet. SAMOA: Scalable Advanced Massive Online Analysis. *Journal of Machine Learning Research*, 16(1):149–153, 2015.

[40] Mahsa Salehi and Lida Rashidi. A survey on anomaly detection in evolving data: [with application to forest fire risk prediction]. *SIGKDD Explor. Newsl.*, 20(1):13–23, May 2018.

[41] Danai Koutra, Neil Shah, Joshua T. Vogelstein, Brian Gallagher, and Christos Faloutsos. Deltacon: Principled massive-graph similarity function with attribution. *ACM Trans. Knowl. Discov. Data*, 10(3), February 2016.

[42] Lida Rashidi, Andrey Kan, James Bailey, Jeffrey Chan, Christopher Leckie, Wei Liu, Sutharshan Rajasegarar, and Kotagiri Ramamohanarao. Node re-ordering as a means of anomaly detection in time-evolving graphs. In Paolo Frasconi, Niels Landwehr, Giuseppe Manco, and Jilles Vreeken, editors, *Machine Learning and Knowledge Discovery in Databases*, pages 162–178, Cham, 2016. Springer International Publishing.

[43] T. Hanamori and T. Nishimura. *Real-time Monitoring Solution to Detect Symptoms of System Anomalies*. FUJITSU Sci. Tech. Journal, 2016.

[44] Marco Huber, Martin Voigt, and Axel Cyrille Ngonga Ngomo. Big data architecture for the semantic analysis of complex events in manufacturing. 09 2016.

[45] Subutai Ahmad, Alexander Lavin, Scott Purdy, and Zuha Agha. Unsupervised real-time anomaly detection for streaming data. *Neurocomputing*, 262:134 – 147, 2017. Online Real-Time Learning Strategies for Data Streams.

[46] Liangwei Zhang, Jing Lin, and Ramin Karim. Adaptive kernel density-based anomaly detection for nonlinear systems. *Knowledge-Based Systems*, 139:50 – 63, 2018.

[47] L. Cai, N. F. Thornhill, S. Kuenzel, and B. C. Pal. Real-time detection of power system disturbances based on *k* -nearest neighbor analysis. *IEEE Access*, 5:5631–5639, 2017.

[48] Milad Chenaghlou, Masud Moshtaghi, Christopher Leckie, and Mahsa Salehi. Online clustering for evolving data streams with online anomaly detection. In Dinh Phung, Vincent S. Tseng, Geoffrey I. Webb, Bao Ho, Mohadeseh Ganji, and

Lida Rashidi, editors, *Advances in Knowledge Discovery and Data Mining*, pages 508–521, Cham, 2018. Springer International Publishing.

[49] A. Bagozi, D. Bianchini, V. De Antonellis, A. Marini, and D. Ragazzi. Big Data Summarisation and Relevance Evaluation for Anomaly Detection in Cyber Physical Systems. In *Proc. of 25th Int. Conference on Cooperative Information Systems (CoopIS 2017)*, pages 429–447, 2017.

[50] D. Ratasich, F. Khalid, F. Geissler, R. Grosu, M. Shafique, and E. Bartocci. A Roadmap Toward the Resilient Internet of Things for Cyber-Physical Systems. *IEEE Access*, 7:13260–13283, 2019.

[51] A. Musil, J. Musil, D. Weyns, T. Bures, H. Muccini, and M. Sharaf. *Multi-Disciplinary Engineering for Cyber-Physical Production Systems*, chapter Patterns for Self-Adaptation in Cyber-Physical Systems, pages 331–368. 2017.

[52] J. Moura and D. Hutchison. Game Theory for Multi-Access Edge Computing: Survey, Use Cases, and Future Trends. *IEEE Communication Surveys and Tutorials*, 21(1):260–288, 2019.

[53] G. Murino, A. Armando, and A. Tacchella. Resilience of cyber-physical systems: an experimental appraisal of quantitative measures. In *2019 11th International Conference on Cyber Conflict (CyCon)*, volume 900, pages 1–19, 2019.

[54] R. Barenji, A. Barenji, and M. Hashemipour. A multi-agent RFID-enabled distributed control system for a flexible manufacturing shop. *Internationa Journal of Advanced Manufacturing Technology*, 71(9):1773–1791, 2014.

[55] B. Vogel-Hauser, C. Diedrich, D. Pantförder, and P. Göohner. Coupling heterogeneous production systems by a multi-agent based cyber-physical production system. In *Proc. of 12th IEEE International Conference on Industrial Informatics (INDN)*, pages 713–719, 2014.

[56] Nadia Galaske and Reiner Anderl. Disruption management for resilient processes in cyber-physical production systems. *Procedia CIRP*, 50:442 – 447, 2016. 26th CIRP Design Conference.

[57] Context-Active Resilience in Cyber Physical Systems (CAR) European Project. `http://www.msca-car.eu`, 2018.

[58] N. Bicocchi, G. Cabri, F. Mandreoli, and M. Mecella. Dynamic digital factories for agile supply chains: An architectural approach. *Journal of Industrial Information Integration*, 15:111–121, 2019.

[59] E. Baralis, S. Paraboschi, and E. Teniente. Materialized Views Selection in a Multidimensional Database. In *Proceedings of International Conference on Very Large Databases (VLDB)*, pages 156–165, 1997.

[60] A. Bagozi, D. Bianchini, V. De Antonellis, M. Garda, and A. Marini. A relevance-based approach for big data exploration. *Future Generation Computer Systems*, 101:51 – 69, 2019.

[61] Tian Zhang, Raghu Ramakrishnan, and Miron Livny. Birch: An efficient data clustering method for very large databases. In *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data*, SIGMOD '96, page 103–114, New York, NY, USA, 1996. Association for Computing Machinery.

[62] S. Biswas and J. Sen. A proposed architecture for big data driven syupply chain analytics. *International Journal of Supply Chain Management*, 2016.

[63] Ada Bagozi, Devis Bianchini, Valeria De Antonellis, and Alessandro Marini. A relevance-based data exploration approach to assist operators in anomaly detection. In *Proc. of 26th Int. Conference on Cooperative Information Systems (CoopIS 2018)*, pages 354–371, Valletta, Malta, 2018.

[64] Ada Bagozi, Devis Bianchini, Valeria De Antonellis, and Massimiliano Garda. Risk monitoring services of discharged sars-cov-2 patients. In *Web Information Systems Engineering – WISE 2020*, pages 578–590, 2020.

[65] Ada Bagozi, Devis Bianchini, and Valeria De Antonellis. Designing context-based services for resilient cyber physical production systems. In *Web Information Systems Engineering – WISE 2020*, pages 474–488, 2020.

[66] Gloria Pumpuni-Lenss, Timothy Blackburn, and Andreas Garstenauer. Resilience in complex systems: An agent-based approach. *Systems Engineering*, 20(2):158–172, 2017.