Summarisation and Relevance Evaluation Techniques for Big Data Exploration: The Smart Factory Case Study

Ada Bagozi(✉), Devis Bianchini, Valeria De Antonellis, Alessandro Marini, and Davide Ragazzi

Department of Information Engineering, University of Brescia,
Via Branze, 38, 25123 Brescia, Italy
adabagozi@gmail.com

Abstract. The increasing connections of systems that produce high volumes of real time data have raised the importance of addressing data abundance research challenges. In the Industry 4.0 application domain, for example, high volumes and velocity of data collected from machines, as well as value of data that declines very quickly, put Big Data issues among the new challenges also for the factory of the future. While many approaches have been developed to investigate data analysis, data visualisation, data collection and management, the impact of Big Data exploration is still under-estimated. In this paper, we propose an approach to support and ease exploration of real time data in a dynamic context of interconnected systems, such as the Industry 4.0 domain, where large amounts of data must be incrementally collected, organized and analysed on-the-fly. The approach relies on: (i) a multi-dimensional model, that is suited for supporting the iterative and multi-step exploration of Big Data; (ii) novel data summarisation techniques, based on clustering; (iii) a model of relevance, aimed at focusing the attention of the user only on relevant data that are being explored. We describe the application of the approach in the smart factory as a case study.

Keywords: Data exploration · Big data · Multi-dimensional data model · Industry 4.0 · Cyber physical systems

1 Introduction

The research challenges raised by the abundance of real time data in Cyber-Physical Systems (CPS) have focused the attention of researchers on the collection, organisation and exploration of data as produced by interconnected systems, enabled by the widespread diffusion of IoT technologies [11]. Collected data are featured by high volumes and velocity and have outgrown the ability to be stored and processed by many traditional systems. Moreover, their value declines very quickly, making organisations' success more and more dependent on how efficiently they can turn collected data into actionable insights. For instance,

E. Dubois and K. Pohl (Eds.): CAiSE 2017, LNCS 10253, pp. 264–279, 2017.
DOI: 10.1007/978-3-319-59536-8_17

advanced Industry 4.0 capabilities, namely self-awareness, self-configuration and self-repairing, as well as *manufacturing servitization*, defined as the strategic innovation of organisations' capabilities and processes to shift from selling products to selling integrated product and service offerings, rely on data collection and sharing [10], according to the emerging "data-driven innovation paradigm" [7].

In this context, many approaches have addressed issues related to data collection and management, data analysis, data visualisation and rendering. Neverthless, Big Data exploration issues have been under-estimated. In this paper, we discuss the ingredients to enable exploration of real time data in a dynamic context of interconnected systems, where large amounts of data must be incrementally collected, organized and analysed on-the-fly. Firstly, we envision exploration as a multi-step process, where data can be browsed through iterative refinements over a set of dimensions, hierarchically modelled, that are used to organise data into a *multi-dimensional model*. Data modeling according to "facets" or "dimensions", either flat or hierarchically organized, has been recognised as a factor for easing data exploration, since it offers the opportunity of performing flexible aggregations of data [3]. On top of the multi-dimensional model, we developed a *data summarisation* approach, in order to simplify overall view over high volumes of data, and a *model of relevance*, aimed at focusing the attention of the user on relevant data only, also when the user is not able to specify his/her requirements through a query. The multi-dimensional model, the data summarisation approach and the model of relevance are the core components of our Big&Open Data Innovation framework (BODaI) and the main contributions of this paper. With respect to exploratory data analysis [13] and Data Mining [6], our approach aims at supporting exploration as a multi-step process, where the user may iteratively improve focus on relevant data, by receiving suggestions of the system based on the model of relevance. Compared to On Line Analytical Processing [5], we manage data that are incrementally collected, organized and analysed on-the-fly. Finally, with respect to traditional faceted search [14], we deal with high data volumes and velocity, that imply efficient techniques for storing and managing them. Given the importance of these research challenges in the Industry 4.0 domain, we describe the application of our approach in the smart factory as a case study.

The paper is organized as follows: Sect. 2 presents a motivating example, used to introduce the innovative aspects of our approach in the Industry 4.0 domain; in Sect. 3 we describe the multi-dimensional model and proposed data summarisation techniques; Sect. 4 provides details about the model of relevance and how this can be engaged within the multi-dimensional model in order to foster big data exploration; the architecture of BODaI framework and experimental evaluation are detailed in Sect. 5; Sect. 6 highlights cutting-edge features of our approach compared to the state of the art; finally, Sect. 7 closes the paper.

2 Motivating Example and Research Challenges

As a motivating example, we introduce here the application of our approach for exploring real time data collected from a machine produced by an Original Equipment Manufacturer (OEM). As shown in Fig. 1, the OEM produces multi-spindle machines, where spindles work independently each other 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) in order to complete different steps in the manufacturing cycle. For each unit, we can measure the velocity of the three axes (X, Y and Z) and 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). Hereafter, we will refer to the measured aspects as *features*.

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 we refer to 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 downtimes.

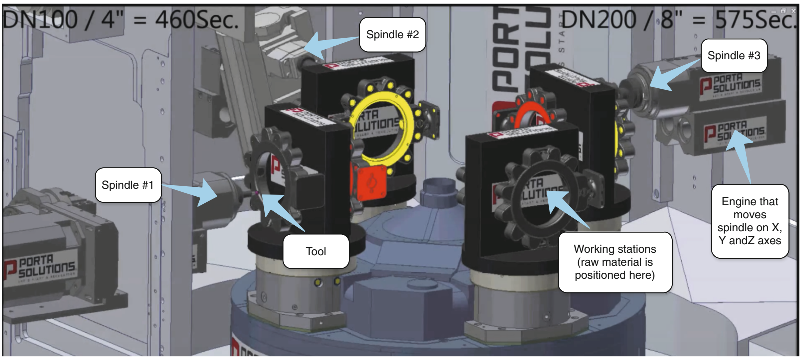

DN100 / 4" = 460Sec.
DN200 / 8" = 575Sec.
Spindle #2
Spindle #3
Spindle #1
Tool
Working stations (raw material is positioned here)
Engine that moves spindle on X, Y andZ axes
PORTA SOLUTIONS

Fig. 1. The multi-spindle machine from which real time data have been collected for exploration purposes.

This opens a set of issues, mainly related to data volumes and velocity and the considered application domain, that can be summarised as follows.

Data Modeling for Exploration. Data modeling according to "facets" (e.g., categories), evenly hierarchically organised, represents a powerful mean to enable incremental and on-the-fly data exploration. A multi-dimensional representation of data can be helpful, 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.

Data Summarisation. The ability of providing a compact view of the huge amount of data collected from the machine is strongly required. A data summarisation approach is recommended, where data should be observed in an aggregated way, instead of monitoring each single data record, that might be not relevant given the high level of noise in the working environment (slight variations in the measured variables). 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.

Data Relevance. The user who explores data needs an underlying data-model to enable fast exploration of the available data, guiding the user towards only those relevant measures that correspond to spindle hardening or tool 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.

3 A Multi-dimensional Model for Big Data Exploration

3.1 Basic Definitions

The basic concept of the multi-dimensional model, on which exploration relies, is the *feature*, that is, a monitored variable (e.g., measured through sensors and machine control). Features are defined as follows.

Definition 1 (Feature). *A feature represents a monitored variable that can be measured. A feature F_i is 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 $F = \{F_1, F_2 \dots F_n\}$ the overall set of features.*

Definition 2 (Measure). *We define a measure $X_i(t)$ a value for the feature F_i, expressed in terms of the unit of measure u_{F_i} and of the timestamp t, that represents the instant in which the measure has been taken. At a given time t, a set of measures can be identified, one for each considered feature. Therefore, we denote with vector $\boldsymbol{X}(t)$ a record of measures $\langle X_1(t), X_2(t), \dots X_n(t) \rangle$ obtained at a given time t and synchronised with respect to the acquisition timestamp.*

Examples. In the running example, velocity of the three axes X, Y and Z, electrical current, the value of spindle rpm and percentage of absorbed power are modelled as features.

3.2 Clustering-Based Data Summarisation

Records of measures collected at a given time interval Δt are clustered. Clustering offers a two-fold advantage: (a) it gives an overall view over a set of measure records, using a reduced amount of information; (b) it allows to depict the behaviour of the system better than single records, that might be affected by noise and false outliers, in order to observe a given physical phenomenon. When dealing with real time data, collected for example in Cyber Physical Systems, we face with data streams, where data are not all available since the beginning, but are collected in an incremental way. For these reasons, an incremental, data-stream clustering algorithm has been developed, in order to extract from records of measures in a time interval Δt a set of clusters aimed at summarising collected measures. The clustering algorithm is performed in two steps: (i) in the first one, a variant of Clustream algorithm [1] is applied, that incrementally processes incoming data to obtain a *set of syntheses*; (ii) in the second step, X-means algorithm is applied [12] in order to cluster syntheses obtained in the previous step. X-means does not require an a-priori knowledge on the number of output clusters. Syntheses are defined as follows.

Definition 3 (Synthesis). *We define a synthesis of records S as a tuple consisting of five elements, that is, $S = \langle N, \boldsymbol{LS}, SS, \boldsymbol{X}0, R\rangle$, where: (i) N is the number of records included into the synthesis (from $\boldsymbol{X}(t_1)$ to $\boldsymbol{X}(t_N)$, where $t_N = t_1 + \Delta t$); (ii) $\boldsymbol{LS}$ is a vector representing the linear sum of measures in S; (iii) SS is the quadratic sum of points in S; (iv) $\boldsymbol{X}0$ is a vector representing the centroid of the synthesis; (v) R is the radius of the synthesis. In particular:*

$$\boldsymbol{LS} = \sum_{k=1}^{N} \boldsymbol{X}(t_k) \quad SS = \sum_{k=1}^{N} \boldsymbol{X}^2(t_k) \tag{1}$$

$$\boldsymbol{X}0 = \frac{\sum_{k=1}^{N} \boldsymbol{X}(t_k)}{N} \tag{2}$$

$$R = \sqrt{\frac{\sum_{k=1}^{N} (\boldsymbol{X}(t_k) - \boldsymbol{X}0)^2}{N}} \tag{3}$$

The second step aims at clustering syntheses. Clustering is performed to minimise the distance between syntheses centroids within the same cluster and to maximise the distance between syntheses centroids across different clusters. Clusters give a balanced view of the observed physical phenomenon, grouping together syntheses corresponding to the same working status. Details about the algorithm for syntheses generation and clustering are out of the scope of this paper.

Definition 4 (Cluster). *A cluster C is defined as follows: $C = \langle \boldsymbol{C}_0, \mathcal{S}_C\rangle$, where $\boldsymbol{C}_0$ is the cluster centroid, $\mathcal{S}_C$ is the set of syntheses belonging to the cluster. We denote with SC the set of identified clusters.*

3.3 Dimensions

Clusters are associated with values of specific *dimensions.* Among dimensions, we mention *time*, *feature space*, *working mode* and other *domain-specific dimensions.*

Time. Time is the most important dimension. In fact, the clustering algorithm described in the previous section is computed incrementally over time. The minimum granularity of time dimension corresponds to the time interval over which clustering is performed. This means that, considering Δt as the time interval on which records of measures are grouped in syntheses, that in turn are clustered, every Δt seconds the clustering algorithm outputs a new cluster set SC built on top of the previous sets. Δt is chosen at configuration time such that $1/\Delta t$ is greater than the data acquisition frequency.

Feature Space. Feature spaces are used to represent different physical phenomena of a system that are being monitored. In the running example, the spindle hardening and the tool wear are feature spaces. A feature space conceptually represents a set of related features, whose measures are useful in order to describe the evolution over the time of monitored physical phenomena. Multiple feature spaces might be observed, and the observation of a feature might be useful to monitor more than one feature space. We denote with $FS = \{FS_1, FS_2, \dots FS_m\}$ the set of feature spaces, where $FS_j \subseteq F$ and $m \leq n$. Feature spaces can be monitored independently each others.

Working Mode. The working mode represents the conditions in which monitored cyber physical system operates. Working mode can be identified through one or more parameters. In our running example, working mode is identified by the kind of manufacturing task that is being processed, described within the Part Program of the machine, and by the machine model. Roughly speaking, working mode represents the *context* in which data analysis/comparison between collected measures might have sense. For example, comparison between the behaviour of two machines is meaningful only if two machines are executing the same Part Program and machine model is the same.

Domain-Specific Dimensions. Other dimensions can be considered depending on the specific domain of interest. In the running example, domain-specific dimensions are the monitored physical system (e.g., the spindle) and the tool used for the manufacturing process.

Dimensions can be organized in hierarchies, at different levels. Formally, we denote with $\mathcal{D} = \mathcal{D}_1 \times \mathcal{D}_2 \times \dots \times \mathcal{D}_p$ the multi-dimensional space created by p dimensions $\mathcal{D}_1, \mathcal{D}_2, \dots \mathcal{D}_p$. We denote with $\mathcal{D}_j^i$ the i-th level in the hierarchy of j-th dimension and with $d_i \in \mathcal{D}_i$ a single value of the dimension $\mathcal{D}_i$.

Example. The *time* dimension can be considered starting from the level of hour (if clustering is performed every hour), hours can be aggregated into days, days can be aggregated into months, that can be in turn aggregated into quarters, that is, `time[hour:days:month:quarters]`. Tools can be aggregated into tool types (`tool[tool:tool_type]`). Spindles can be aggregated into the machines they belong to (`monitored_system[spindle:machine]`).

3.4 Multi-dimensional Model

Our multi-dimensional model consists of an hypercube such as the one shown in Fig. 2 for the running example. Dimensions represent axes of the hypercube, that is defined as follows.

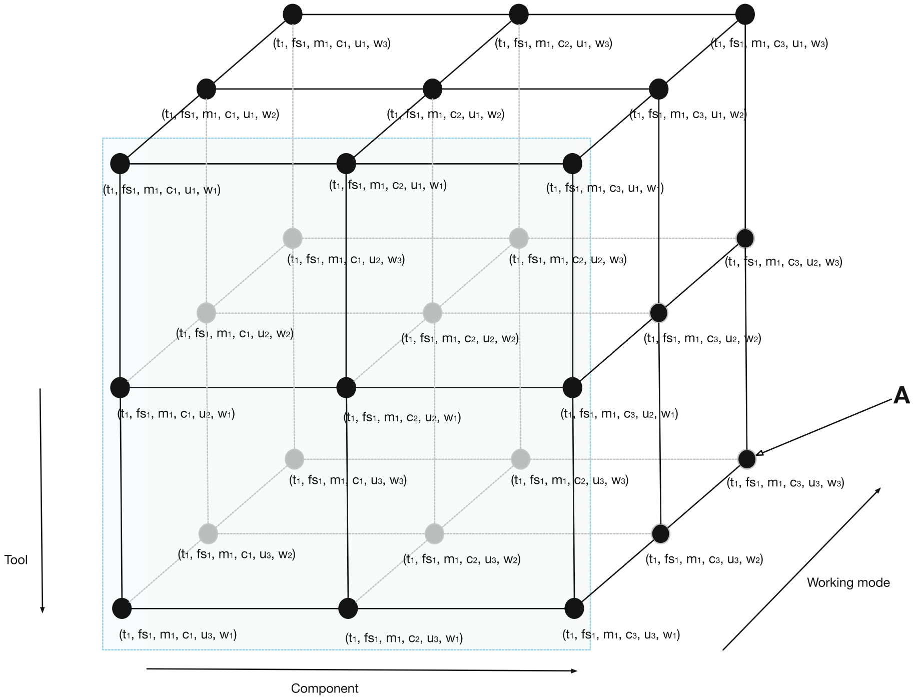

(t1, fs1, m1, c1, u1, w3)
(t1, fs1, m1, c2, u1, w3)
(t1, fs1, m1, c3, u1, w3)
(t1, fs1, m1, c1, u1, w2)
(t1, fs1, m1, c2, u1, w2)
(t1, fs1, m1, c3, u1, w2)
(t1, fs1, m1, c1, u1, w1)
(t1, fs1, m1, c2, u1, w1)
(t1, fs1, m1, c3, u1, w1)
(t1, fs1, m1, c1, u2, w3)
(t1, fs1, m1, c2, u2, w3)
(t1, fs1, m1, c3, u2, w3)
(t1, fs1, m1, c1, u2, w2)
(t1, fs1, m1, c2, u2, w2)
(t1, fs1, m1, c3, u2, w2)
A
(t1, fs1, m1, c1, u2, w1)
(t1, fs1, m1, c2, u2, w1)
(t1, fs1, m1, c3, u2, w1)
(t1, fs1, m1, c1, u3, w3)
(t1, fs1, m1, c2, u3, w3)
(t1, fs1, m1, c3, u3, w3)
Tool
(t1, fs1, m1, c1, u3, w2)
(t1, fs1, m1, c2, u3, w2)
(t1, fs1, m1, c3, u3, w2)
Working mode
(t1, fs1, m1, c1, u3, w1)
(t1, fs1, m1, c2, u3, w1)
(t1, fs1, m1, c3, u3, w1)
Component

Fig. 2. The multi-dimensional data model for big data exploration.

Definition 5 (Multi-dimensional model). *We describe the multi-dimensional model as a set $\mathcal{V}$ nodes. Each node $v \in \mathcal{V}$ is described as $v = \langle SC(d_1, d_2, \dots d_p) \rangle$, where $SC(d_1, d_2, \dots d_p)$ represents a cluster set, obtained at fixed values for each dimension $d_1 \in \mathcal{D}_1, d_2 \in \mathcal{D}_2 \dots\ d_p \in \mathcal{D}_p$.*

For example, in Fig. 2 the node identified as "A" represents the cluster set identified at time t_1 for machine $\mathtt{m}_1$ (spindle $\mathtt{c}_3$), that is using tool $\mathtt{u}_3$ and is working within the working mode $\mathtt{w}_3$, considering features in the feature space $\mathtt{fs}_1$. Exploration will be performed within this data structure as described in the next section.

4 Relevance-Based Big Data Exploration

The proposed approach enables exploration of real time data incrementally collected and organized, as well as aggregated on-the-fly. The user is guided by the multi-dimensional model through a set of steps according to data relevance aspects.

4.1 Model of Data Relevance

In Exploratory Computing (EC), during exploration steps data can be considered as *relevant* if they differ from an *expected status*. The latter one can be for example a normal distribution of values of a feature, as assumed in [3]. In our case, the expected status corresponds to the one of normal working conditions for monitored cyber physical systems. The expected status can be tagged by domain expert while observing the monitored system when operates normally. Let's denote with $\hat{SC}(d_1, d_2, \dots d_p)$ the cluster set identified during such condition, for dimension values fixed at $d_1, d_2, \dots d_p$.

The model of relevance adopted in our approach is based on the concept of *cluster distance*. The algorithm proposed here is inspired by [4] and has been adapted to the multi-dimensional model considered in this paper. Given two sets of clusters $SC_1 = \{C_1, C_2, \dots, C_n\}$ and $SC_2 = \{C'_1, C'_2, \dots, C'_m\}$, with size n and m respectively, we evaluate the distance between SC_1 and SC_2 by aggregating distances between each cluster belonging to SC_1 and the closest cluster belonging to SC_2 and viceversa, for symmetry purposes (see, for example, C_2 and C'_2 in Fig. 3). Formally, the distance is computed as:

$$\varDelta(SC_1, SC_2) = \frac{\sum_{i=1}^{n} d(C_i, SC_2) + \sum_{j=1}^{m} d(SC_1, C'_j)}{m + n} \tag{4}$$

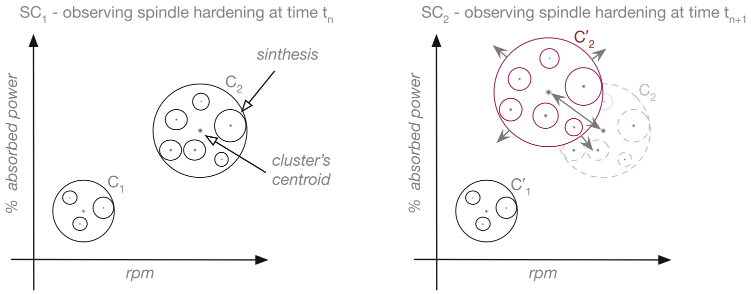

SC$_1$ - observing spindle hardening at time t$_n$
sinthesis
C$_2$
cluster's centroid
C$_1$
% absorbed power
rpm
SC$_2$ - observing spindle hardening at time t$_{n+1}$
C'$_2$
C$_2$
C'$_1$
% absorbed power
rpm

Fig. 3. Illustration of cluster's sets changes in time due to spindle hardening that may cause a decrease of rpm and an increase of the percentage of absorbed power. In the figure is showed how the cluster $C_2 \in SC_1$ changed its position, as well as its size, from time t_n to t_{n+1}; this changes may indicate an anomaly like the spindle hardening.

where $d(C_i, SC_2) = min_{j=1,\dots m} d_c(C_i, C'_j)$ and $d(SC_1, C'_j) = min_{i=1,\dots n}$ $d_c(C_i, C'_j)$ is the distance between clusters. To compute the distance between two clusters $d_c(C_i, C'_j)$, we combined different factors: (i) the distance between clusters centroids $d_{\boldsymbol{C}_0}(C_i, C'_j)$, to verify if C'_j moved with respect to C_i (or viceversa); (ii) the *intra-cluster distance* $d_c^{intra}(C_i, C'_j)$, to verify if there has been an expansion or a contraction of cluster C'_j with respect to C_i; (iii) the difference in number of syntheses contained in C_i and C'_j, denoted with $d_N(C_i, C'_j)$:

$$d_c(C_i, C'_j) = \alpha d_{\boldsymbol{C}_0}(C_i, C'_j) + \beta d_c^{intra}(C_i, C'_j) + \gamma d_N(C_i, C'_j) \tag{5}$$

where α, β and $\gamma \in [0,1]$ are weights such that $\alpha+\beta+\gamma = 1$, used to balance the impact of terms in Eq. (5). To set the optimal weights, a grid procedure can be performed over α and β (γ is set with $1-\alpha-\beta$), with the value of each weight varying from 0 to 1. In our preliminary experiments, we put $\alpha = \beta = \gamma = \frac{1}{3}$.

In particular, $d_{\boldsymbol{C}_0}(C_i, C'_j)$ is computed by applying the Euclidean distance ($D0$) between clusters' centroids, according to the following formula:

$$D0 = \sqrt{(\boldsymbol{C}_0^i - \boldsymbol{C}_0^j)^2} \tag{6}$$

where $\boldsymbol{C}_0^i$ and $\boldsymbol{C}_0^j$ are centroids of C_i and C'_j, respectively. The intra-cluster distance $d_c^{intra}(C_i, C'_j)$ is obtained by recursively computing $\Delta(\mathcal{S}_{C_i}, \mathcal{S}_{C'_j})$ on the sets of syntheses of C_i and C_j, that is:

$$d_c^{intra}(C_i, C'_j) = \frac{\sum_{k=1}^{n_1} d(S_k, C'_j) + \sum_{h=1}^{n_2} d(C_i, S_h)}{n_1 + n_2} \tag{7}$$

where $S_k \in \mathcal{S}_{C_i}$, $S_h \in \mathcal{S}_{C'_j}$, $|\mathcal{S}_{C_i}| = n_1$, $|\mathcal{S}_{C'_j}| = n_2$, $d(S_k, C'_j) = min_{h=1,\ldots n_2} d_s(S_k, S_h)$ and $d(C_i, S_h) = min_{k=1,\ldots n_1} d_s(S_k, S_h)$. Term $d_s(S_k, S_h)$ represents the average inter-syntheses distance ($D1$):

$$D1 = \sqrt{\frac{\sum_{i=1}^{N1} \sum_{j=N1+1}^{N1+N2} (\boldsymbol{X}(t_i) - \boldsymbol{X}(t_j))^2}{N1 N2}} \tag{8}$$

where $N1$ and $N2$ are the number of records in S_k and S_h, respectively.

4.2 Multi-step Guided Data Exploration

Starting the Exploration. To start the exploration, the user might specify a set d^r of preferred values for the dimensions he/she is interested in, where $d^r = \{d_1^r, d_2^r, \ldots d_p^r\}$ and $d_i^r \in \mathcal{D}_i$. The user might specify preferences on a subset of dimensions in $\mathcal{D}$. Let's denote as *bounded* the dimensions on which the user expressed a preference, as *unbounded* the other dimensions. The systems identifies a subset $\mathcal{V}' \subseteq \mathcal{V}$ of nodes within the multi-dimensional model, such that the values of bounded dimensions corresponds to the one specified in d^r. The exploration will start from nodes $v \in \mathcal{V}'$. We remark here that bounded dimensions must be considered starting from selected level in the hierarchy. This means that if the user selects a specific machine, the `monitored_system` dimension is bounded at machine level, but remains unbounded at spindle level, that is, no preferences are expressed on spindles and the user is enabled to browse data among all spindles that compose the selected machine. For example, if $d^r = \langle -, \mathtt{fs}_1, \mathtt{m}_1, -, -, \mathtt{w}_1 \rangle$, feature space, machine and working mode are the *bound* dimensions, while time, tool and spindle are the *unbound* ones: the front facade of hypercube shown in Fig. 2 groups the candidate nodes $v \in \mathcal{V}'$.

We assume that the user formulates d^r as an explicit, albeit vague exploration request, and expects the system to suggest some promising data to explore.

To this aim, we need a model of relevance to establish what data can be considered as relevant or interesting. The system uses the model of relevance in order to restrict the set of nodes from which to start the exploration among nodes $v \in \mathcal{V}'$, that is, the set of relevant data to be explored. For each node $v = \langle SC(d_1, d_2, \ldots d_p) \rangle \in \mathcal{V}$, the node is considered as relevant if the clusters distance with respect to the set of clusters $\hat{SC}(d_1, d_2, \ldots d_p)$ overtakes a predefined threshold, that is, $\Delta(SC(d_1, d_2, \ldots d_p), \hat{SC}(d_1, d_2, \ldots d_p)) \geq \delta$. Such a model of relevance enables the identification of relevant nodes also when the user does not specify any constraints in d^r, that is, he/she does not have any idea from which dimensions and data to start the exploration. In the latter case, the same relevance criteria is used, where the candidate nodes $v \in \mathcal{V}$ are all the ones in the hypercube.

How the Exploration Goes On. Starting from nodes selected in the previous step, exploration goes on through a set of different traversals that the user applies in order to move from one node to the other ones. We define a *traversal* as $\sigma(\tau_\sigma, v_i, v_j, \omega_\sigma)$, where: (i) τ_σ is the kind of traversal (among *drill-down*, *roll-up* and *sibling*), inspired by OLAP operators, as detailed below; (ii) $v_i \in \mathcal{V}$ is the starting node; (iii) $v_j \in \mathcal{V}$ is the destination node; (iv) ω_σ is a weight assigned to the traversal, computed according to the model of relevance. By using traversals it's possible to move in all directions.

Using a *drill-down* traversal the user moves towards a node $v_j \in \mathcal{V}$ by specialising any of the dimensions in $v_i \in \mathcal{V}$. An example of drill-down traversal is to move from a node labeled with $\langle t_1, \mathtt{fs}_1, \mathtt{m}_1, \mathtt{u}_1, \mathtt{w}_1 \rangle$ towards a node labeled with $\langle t_1, \mathtt{fs}_1, \mathtt{c}_2, \mathtt{u}_1, \mathtt{w}_1 \rangle$, where $\mathtt{c}_2$ (spindle) specialises $\mathtt{m}_1$ (machine) in the hierarchy of `monitored_system` dimension. Note that this means to include the spindle among the *bounded* variables and therefore to restrict the exploration space.

The *roll-up* traversal is similar. Using a *roll-up* traversal the user moves towards a node $v_j \in \mathcal{V}$ by generalising any of the dimensions in $v_i \in \mathcal{V}$. An example of roll-up traversal is to move from a node labeled with $\langle t_1, \mathtt{fs}_1, \mathtt{c}_2, \mathtt{u}_1, \mathtt{w}_1 \rangle$ towards a node labeled with $\langle t_1, \mathtt{fs}_1, \mathtt{m}_1, \mathtt{u}_1, \mathtt{w}_1 \rangle$. This also means to include the spindle among the *unbounded* variables and therefore to expand the exploration space.

Using a *sibling* traversal the user moves towards a node $v_j \in \mathcal{V}$ by changing the value of one of the dimensions in $v_i \in \mathcal{V}$. An example of sibling traversal is to move from a node labeled with $\langle t_1, \mathtt{fs}_1, \mathtt{m}_1, \mathtt{u}_1, \mathtt{w}_1 \rangle$ towards a node labeled with $\langle t_1, \mathtt{fs}_1, \mathtt{m}_2, \mathtt{u}_1, \mathtt{w}_1 \rangle$, where $\mathtt{m}_1$ and $\mathtt{m}_2$ are two machines, that is, values of the same level in the hierarchy of `monitored_system` dimension. This traversal does not change the sets of *bounded* and *unbounded* variables and therefore does not change in size the exploration space.

The model of relevance can be used here by the system to suggest more relevant nodes to move on: in particular, nodes $v_j \in \mathcal{V}$ are suggested such as $\Delta(SC(d_1, d_2, \ldots d_p), \hat{SC}(d_1, d_2, \ldots d_p)) \geq \delta$, where $v_j = \langle SC(d_1, d_2, \ldots d_p) \rangle$.

5 Implementation and Experiments

5.1 Architecture of the BODaI Framework

Figure 4 depicts the functional architecture of the BODaI framework. The framework has been developed in Java as a modular infrastructure composed of:

- `BODaI_BigData`, that is based on NoSQL technology (MongoDB) and stores records of measures, incrementally provided by monitored physical system; the composition of a record is defined within a *Config file*; different records are processed in parallel;
- `BODaI_model`, that contains all metadata the framework relies on (hierarchies of dimensions, organisation of features within feature spaces, features metadata such as names and unit of measures), as well as cluster sets, syntheses information and computed distances used in the model of relevance for guiding the exploration; the size of this information is much lower than the total amount of collected measures and MySQL technology has been used; both the `BODaI_BigData`, and the `BODaI_model`, are accessed through the BDAO (*BODaI Data Access Objects*);
- *BSB level* (BODaI Service Bus), that manages the interactions between BDAO and the framework services;
- *Data Acquisition Service*, in charge of collecting records of measures, synchronising timestamps and storing acquired data within the `BODaI_BigData`, according to feature spaces as specified in `BODaI_model`; during acquisition data processing is strongly minimised to avoid bottlenecks in data acquisition; costly data elaboration steps are postponed in a second step, where other services (clustering, data control, cluster distance computation) are invoked in parallel;
- *Data Control Service*, *Clustering Service* and *Cluster Distance Service*, in charge of performing controls on collected records, clustering and cluster sets distance computation, respectively;
- *Notification Service*, in charge of sending a notification when an unexpected variation between distances of cluster sets has been identified; it also manages notifications raised when data control is executed.

5.2 Real Use Cases

We applied the approach described in this paper to the Industry 4.0 application domain. We considered a factory producing multi-spindle machines for various industrial sectors: automotive, aviation, water industry, etc. Specifically, the multi-dimensional model enabled to monitor axle hardening by observing changes in the values of energy consumption (spindle engine charge coefficient) for similar rpm, with reference to the tool that has been used. By detecting energy consumption differences using different tools, we identified spindle hardening as the possible anomaly that increases the energy request to perform the

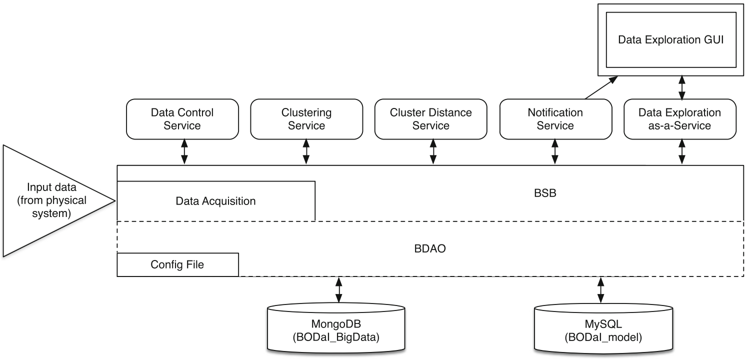
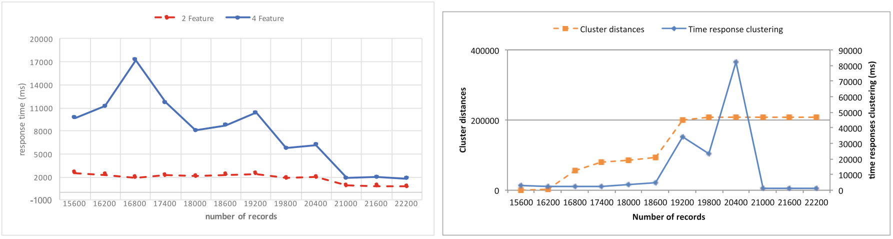


Fig. 4. The functional architecture of the BODaI framework.

manufacturing operations. If the increase in energy consumption is related only to the usage of a particular tool, this has been recognised as a symptom of a possible excessive tool wear. Next step will focus on monitoring of other variables like the absorbed electrical current on the axes X, Y, Z. The level of change in these variables may help in measuring the degree of tool wear and learning the best moment to change it before suffering a tool break and a machine downtime.

Experiments. We performed experiments in order to demonstrate the feasibility of our approach in terms of processing time and its effectiveness in providing summarised data for exploration purposes. Our evaluation focuses mostly on system performance. We collected real data from three machines, each one equipped with three spindles and different tools. On each spindle, we monitored the features listed in the motivating example: the velocity of the three axes (X, Y and Z) and 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). We collected 140 millions of records from the three machines. All records present a timestamp, and have been collected every 200 ms (5 records per second). We run experiments on an Intel Core i7-6700HQ, CPU 2.60 GHz, 4 cores, 8 logical cores, RAM 16 GB. As suggested in [2], during acquisition phase data processing is strongly minimised to avoid bottlenecks, by delaying clustering in a second phase. Collected records of measures have been saved within MongoDB as JSON documents grouped into collections. Each document contains a record $\boldsymbol{X}(t)$ of measures, labeled with the values of dimensions $d_1, d_2, \ldots d_p$. The structure of documents is maintained very simple, with at most one level of depth, and collection have been organised considering the time as main dimension, in order to speed up both data storage and data extraction for clustering, that is applied to records grouped with respect to the timestamps. This enabled to storage all 140 millions of records in 1 h and 14 min, with an acquisition rate of ∼31,531 records per second. Experimental results depicted in Fig. 5(a) show how these tasks can be addressed given the data acquisition rate. We recall here that clustering is applied on slots of records on a time internal Δt. We tested clustering and hypercube generation on real data considering average values on 2 and 3 features. The worse response time corresponds to the case where we performed clustering and distance computation tasks when no previous syntheses had been generated. Also in that case, these tasks are able to process ∼15,600 records in 11.5 s, that is able to process ∼1,356 records per second. Through the tasks of syntheses generation and clustering, the processed set of records is reduced to 7,2% on average. In Fig. 5(b) we tested the effectiveness of model of relevance by simulating strong variations in collected measures. We observed an evident variation in distance between cluster sets at the cost of decreasing the processing time to ∼255 records per second, that is acceptable.



(a) (b)

Fig. 5. Tests on efficiency of clustering and hypercube generation (a) And on the effectiveness of the model of relevance, introducing a variation in collected records (b). Number of records on X axis represent different incremental steps.

5.3 Considerations

The approach revealed to be useful in order to extract information for supporting production operator (i.e., the user of our system in this case study) in taking better decisions, thus preventing failures or increasing production efficiency. These observations are performed by operators to provide prompt maintenance services, thus avoiding long downtime periods. Using our model the operator is able to fully explore the multi-dimensional model, i.e., all data nodes can be explored using the three types of traversals introduced in Sect. 4, and it is possible to use traversals in any order and in a sequence of any length. The traversals are also intuitive, since they are inspired by the rollup, drill down and pivot operations of data cube. In addition here, we exploit the model of relevance to further reduce the exploration space. Operators can focus their attention on some relevant measures, explore them, verify the machine working conditions also according to their experience and decide to activate or not a maintenance activity. In this way, explorative approach can be used to adjust planning of maintenance interventions as scheduled through traditional, offline data mining techniques, that use historical data for their purposes. In fact, several latent factors might influence manufacturing operations and might have an impact on maintenance schedule. These factors cannot be easily detected through measured variables and the role of human actor is still of paramount importance for avoiding useless maintenance interventions, that are costly both for the OEM and for the OEM's client. The data exploration viewpoint enables to improve this task also for unexperienced maintenance operators through decisions supported by the system.

6 Related Work

Other approaches have been specifically focused on data exploration and exploratory computing research fields. Comparison criteria in this case include data characteristics (structured/semistructured/unstructured data, traditional vs big data, OLAP vs OLTP), the way data are collected (incrementally or one-step collection before starting data processing), the adopted exploration techniques, the model of relevance (if any), application of data mining or query approximation techniques, technological issues (e.g., the DBMS technology among SQL-based, NoSQL, NewSQL). The presentation of Exploratory Computing as a comprehensive approach that includes the notions of "exploration as a multi-step process", model of relevance, data summarisation, multi-dimensional data modeling is given in [3]. In this paper, authors proposed a model of relevance based on statistical distribution of data. Compared to them, our approach has a model of relevance based on clustering aimed at detecting deviations from the normal working conditions of a monitored physical system. In [9] cube exploration is discussed, in order to give OLAP-based exploration facilities that help users in navigating multi-dimensional data. No model of relevance is proposed and the aim is at foreseeing user's explorative actions in order to properly apply techniques of query approximation. Authors in [15] propose the application of query approximation techniques to big data that are incrementally collected. Here approximation methods are based on the analysis of user's action previously performed and on statistical properties of data, no model of relevance is proposed and the concept of exploration as a multi-step process has not been addressed.

In [8] an approach operating on structured data stored within a PostgreSQL database is proposed. Data are grouped according to specific criteria (e.g., all data in a given time interval, or all geographical data in the same area). These groups are referred to as *semantic windows*. The user is supported in formulating query where selection criteria and ranges of data are required. Query by sampling is applied and samples are compared against user's query to check their compliance. If sampled data are relevant with respect to the query, all data in the same semantic window are presented to the user and next queries are performed on the same data. With respect to this approach, we proposed a model of relevance for enabling exploration also when the user is not able to specify his/her requirements through a query. Moreover, we focused on big data incrementally collected and summarised.

7 Concluding Remarks

In this paper, we discussed the ingredients to enable exploration of real time data in a dynamic context of interconnected systems, where large amounts of data must be incrementally collected, organized and analysed on-the-fly: (i) a multi-dimensional model, that is suited for supporting the iterative and multi-step nature of data exploration; (ii) efficient data summarisation techniques, based on clustering, in order to simplify overall view over high volumes of data; (iii) a model of relevance, aimed at focusing the attention of the user on relevant data only, also when the user is not able to specify his/her requirements through a query. Given the importance of these research challenges in the Industry 4.0 domain, we applied our approach in the smart factory as a case study. Future development efforts will be devoted to a parallelisation of data clustering, in order to further speed up data elaboration in the multi-dimensional model, the study of data visualisation techniques, automate and operationalise knowledge extracted from data produced by the system and the development of a GUI specifically meant for data exploration. With reference to the case study, the migration of the BODaI infrastructure onto the Niagara IoT framework[1] is being implemented.

References

1. Aggarwal, C., Han, J., Wang, J., Yu, P.: A framework for clustering evolving data streams. In: Proceedings of VLDB 2003, pp. 81–92 (2003)
2. Biswas, S., Sen, J.: A proposed architecture for big data driven supply chain analytics. Int. J. Supply Chain Manag. (2016)
3. Buoncristiano, M., Mecca, G., Quintarelli, E., Roveri, D.S., Tanca, L.: Database challenges for exploratory computing. SIGMOD Rec. **44**(2), 17–22 (2015)
4. Goldberg, M., Hayvanovych, M., Magdon-Ismail, M.: Measuring similarity between sets of overlapping clusters. In: Proceedings of 2nd IEEE International Conference on Social Computing, pp. 303–308 (2010)
5. Golfarelli, M., Rizzi, S.: Data Warehouse Design: Modern Principles and Methodologies. McGraw-Hill, New York (2009)
6. Han, J., Kamber, M.: Data Mining: Concepts and Techniques. Morgan Kaufmann Publisher, Burlington (2006)
7. Hou, Z., Wang, Z.: From model-based control to data-driven control: survey, classification and perspective. Inf. Sci. **235**, 3–25 (2013)
8. Kalinin, A., Cetintemel, U., Zdonik, S.: Interactive data exploration using semantic windows. In: Proceedings of ACM SIGMOD 2014, pp. 505–516 (2014)
9. Kamat, N., Jayachandran, P., Tunga, K., Nandi, A.: Distributed and interactive cube exploration. In: Proceedings of ICDE 2014 (2014)
10. Lee, J., Kao, H.A.: Service innovation and smart analytics for industry 4.0 and big data environment. In: 6th Conference on Industrial Product-Service Systems (2014)
11. Monostori, L.: Cyber-physical production systems: roots, expectations and R&D challenges. In: 47th CIRP Conference on Manufacturing Systems, pp. 9–13 (2014)

[1] https://www.tridium.com/en/products-services/niagara4.

12. Pelleg, D., Moore, A.: X-means: extending K-means with efficient estimation of the number of clusters. In: 17th International Conference on Machine Learning, pp. 727–734 (2000)
13. Tukey, J.: Exploratory Data Analysis. Reading (1977)
14. Tunkelang, D.: Faceted Search (Synthesis Lectures on Information Concepts, Retrieval and Services). Morgan and Claypool Publishers, San Rafael (2009)
15. Wasay, A., Athanassoulis, M., Idreos, S.: Queriosity: automated data exploration. In: Proceedings of the IEEE International Congress on Big Data (2015)