A Relevance-Based Data Exploration Approach to Assist Operators in Anomaly Detection

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

Department of Information Engineering, University of Brescia,
Via Branze, 38, 25123 Brescia, Italy
devis.bianchini@unibs.it

Abstract. Data is emerging as a new industrial asset in the factory of the future, to implement advanced functions like state detection, health assessment, as well as manufacturing servitization. In this paper, we foster Industry 4.0 data exploration by relying on a relevance evaluation approach that is: (i) flexible, to detect relevant data according to different analysis requirements; (ii) context-aware, since relevant data is discovered also considering specific working conditions of the monitored machines; (iii) operator-centered, thus enabling operators to visualise unexpected working states without being overwhelmed by the huge volume and velocity of collected data. We demonstrate the feasibility of our approach with the implementation of an anomaly detection service in the Smart Factory, where the attention of operators is focused on relevant data corresponding to unusual working conditions, and data of interest is properly visualised on operator's cockpit according to adaptive sampling techniques based on the relevance of collected data.

Keywords: Data exploration · Data relevance · Data summarisation Clustering · Big data · Anomaly detection · Industry 4.0

1 Introduction

Big data management is an ever-growing research topic given the emerging data-intensive applications of the Smart Factory. In order to improve operation process performance, monitoring, control and health assessment [11], big data streams, generated by embedded systems (RFID technology, sensors, mobile and wearable devices) are collected and processed in the cyber space (edge and cloud computing). In this context, human operators still play a crucial role to recognise critical situations that have not been encountered before, based on their long-term experience, but they must be supported in the identification of relevant data without being overwhelmed by the huge amount of information. In the so-called "Human in the Loop Cyber Physical Systems (CPS)", human actions and machine actuations go hand-by-hand and can often complement each other [14].

H. Panetto et al. (Eds.): OTM 2018 Conferences, LNCS 11229, pp. 354–371, 2018.
https://doi.org/10.1007/978-3-030-02610-3_20

As an example of CPS, let's consider a multi-spindle machine, designed to perform flexible manufacturing tasks. The machine is equipped with multiple spindles (e.g., from three to five), that work independently each other on the raw material. Spindles use different tools (that are selected according to the instructions specified in the part program executed by the numerical control of the machine) in distinct steps of the manufacturing process. Spindle precision, working performance, as well as minimisation of tool breaks and machine downtimes are critical factors in these kinds of systems. Therefore, monitoring activities might be very complex, checking several kinds of events in multiple conditions in order to identify anomalies. Anomalies can be discovered when incoming data goes beyond or below an expected range or with the occurrence of unexpected data patterns [13]. Traditional anomaly detection solutions (e.g., [8,9]) apply machine learning techniques to train proper models using historical data and use them to predict the future behaviour of monitored systems. The occurrence of unknown working states, never used before to train machine learning models, can be recognised and managed by operators according to their expertise. To this aim, operators must be supported in the effective exploration of data streams. For example, in the multi-spindle machine the 'spindle rolling friction torque increase' and the 'tool wear' should be promptly detected and avoided. The former one may happen for lack of lubrication or other mechanical wears like bearings damage. The latter one may lead to long downtimes as well and is managed through tool usage optimisation in order to balance the trade-off between the tools wear and the risk of tool breaking during manufacturing. Several working conditions must be considered, with a high likelihood of finding behaviours never met before. On the other hand, the increasing importance of human-machine interactions [7] calls for new models and techniques to organise collected data according to different exploration perspectives and to attract the attention of operators on relevant data only.

In this paper, we propose a novel approach where multi-dimensional data modelling, data summarisation and relevance evaluation techniques are proposed to implement big data exploration and anomaly detection based on data streams. In particular: (i) collected data are organised according to different dimensions, in order to meet distinct system monitoring requirements; (ii) a clustering algorithm for big data streams is applied to provide a comprehensive view over collected data and to enable data exploration using a reduced amount of information; (iii) data relevance techniques focus the attention of operators on relevant data only, thus increasing the effectiveness and efficiency of the data exploration process. The proposed model and techniques have been tested in the Smart Factory context for anomaly detection. Nevertheless, they have to be intended as a general approach for Big Data exploration. Proposed model and techniques aim at preparing data to address "Human in the Loop" issues.

The proposed approach relies on the IDEAaS (Interactive Data Exploration As-a-Service) framework [3]. Specifically, we extend the research presented in [3] with the following novel contributions:

(i) we introduce a mechanism to adapt data monitoring (e.g., for anomaly detection) based on the relevance evaluation;
(ii) we address relevance-driven adaptive sampling for visualisation purposes on the operator's cockpit;
(iii) we expand the experimental results, performing additional experiments to test effectiveness and response times of data relevance evaluation.

The paper is organised as follows: in Sect. 2 we introduce the research challenges; in Sect. 3 we provide an overview of the relevance-based data exploration approach and of the IDEAaS framework; Sect. 4 contains the description of the multi-dimensional model on which the approach relies; in Sects. 5 and 6 relevance-based techniques and adaptive data visualisation are described; Sect. 7 presents experimental evaluation; in Sect. 8 related work are discussed; finally, Sect. 9 closes the paper.

2 Research Challenges

To support big data exploration in dynamic contexts of interconnected systems, such as the considered application scenario, several research challenges raise and must be addressed.

Flexibility. Exploration depends on different analysis requirements. For example, in the considered application scenario the 'spindle rolling friction torque increase' and 'tool wear' events must be monitored to manage maintenance activities and purchase of new tools. Since many unknown situations may occur, due to the complexity of monitored system, analysts and operators must be supported in the identification of possible invisible problems [12]. Multi-dimensional data modelling represents a powerful mean to enable organisation of data according to different perspectives, in turn related to distinct observed problems and requirements. Data modelling according to "facets" or "dimensions", either flat or hierarchically organised, has been recognised as a factor for easing data exploration, since it offers the opportunity of performing flexible aggregations of data [18]. Moreover, a definition of *relevance* is required to attract the operator's attention on relevant data only, corresponding to an unexpected status. Also the concept of *unexpected status* must be defined as well.

Context-Awareness. The detection of relevant data may also depend on the specific working conditions of the observed system. For example, the machine performance may change with respect to the specific part program that is being executed. In different conditions, the range of tolerance for a given measure may be different. Relevance evaluation algorithms and visualisation tools must reflect this difference.

Operator-Centered Visualisation. Operators must be able to visualise unexpected working states and relevant data without being overwhelmed by the huge volume and velocity of collected data. The ability of providing a compact view over data is strongly required. Data summarisation and sampling techniques are recommended, where data is processed and observed in an aggregated way, instead of monitoring each single record [1].

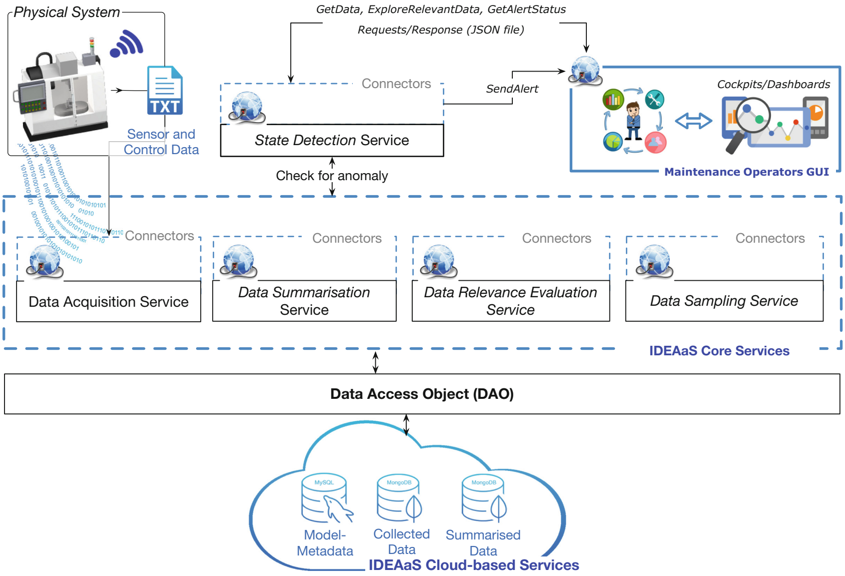


Fig. 1. The IDEAaS framework architecture.

3 Approach Overview

Figure 1 presents the IDEAaS framework modular architecture. The framework is implemented according to a service-oriented architecture, where *Core Services* implement data acquisition, data summarisation, sampling and relevance evaluation, and extensible services, built upon core ones, implement data-intensive functionalities for different application domains, such as the Industry 4.0 one. Among these data-intensive functionalities, in this paper we describe the *State Detection Service*.

As shown in the figure, data coming from the physical system, collected through sensors and IoT technologies, is sent to the *Data Acquisition Service* to be stored in the cyber space. Data is collected according to a set of features. Examples of features for the considered multi-spindle machines are spindle velocity (nm/min), the absorbed electric current (Amp) on X, Y and Z axes, the spindle rotation speed (rpm) and the percentage of absorbed power (%). We refer to measures as the values collected for each feature, associated with a given timestamp. Let's denote with $F = \{F_1, F_2 \dots F_n\}$ the overall set of features. We formally define a measure for the feature F_i as a scalar value $X_i(t)$, expressed in terms of the unit of measure u_{F_i}, taken at the timestamp t. The Data Acquisition Service operates in order to minimise time spent for data acquisition. Specifically, measures are first saved as JSON documents within a NoSQL database (*Collected Data*), using MongoDB technology. Measures are associated with other information about the physical system and the working conditions in which measures have been collected, for example, the tool used for manufacturing or the part program that is being executed by the machine. This information is modelled through analysis dimensions, resulting in a multi-dimensional data model that is detailed in Sect. 4. The *Model-MetaData* relational database (MySQL) contains metadata about dimensions of the model.

The *Data Summarisation Service* is in charge of summarising collected measures. This service applies clustering to aggregate measures that are closely related in the multi-dimensional space and ideally correspond to the same behaviour of the monitored system. Clustered measures are stored using MongoDB technology as well (*Summarised Data*) and processed by the *Data Relevance Evaluation* service, that helps identifying relevant data. Finally, the *Data Sampling* service applies relevance-based sampling techniques 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 data exploration is detailed in the next sections, with focus on the anomaly detection issues. The IDEAaS framework has been implemented in Java, on top of a Glassfish Server Open Source Edition 4.

3.1 State Detection Service in a Nutshell

The State Detection Service is in charge of detecting current status of the monitored system and managing the interaction with visualisation tools, such as cockpits and dashboards, on which operators can explore data.

We consider four different values for the *status* of the monitored system, (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, we introduced the notion of *contextual bounds.* 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. If the measure overtakes warning bounds, but not the error ones, then the feature status is `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 every state change. In fact, the State Detection Service provides the following methods, as remarked in Fig. 1:

- `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 Service and adapts the anomaly detection frequency according to the data relevance, as detailed in Sect. 5.2;
- `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 service; data is transferred as clusters of aggregated measures (as shown in Sect. 4) and visualised according to the multi-dimensional model described in the next section; this method has been designed to support operators to focus on relevant data only, without specifying any data search and filtering criteria, since operators do 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, as described in Sect. 6.

4 Clustering Based Multi-dimensional Model

In the multi-dimensional model used within the IDEAaS framework, *measures* are organised through the feature spaces and the domain-specific dimensions.

A feature space conceptually represents a set of related features, that are jointly measured to observe a physical phenomenon. In the example domain, the set composed of spindle power absorption and rpm features is a feature space used to monitor spindle rolling friction torque increase. In fact, spindle rolling friction torque increase may be identified when the rpm value decreases and, at the same time, the power absorption increases. Therefore, these two features must be monitored jointly. Given a feature space $FS_j = \{F_1, \dots F_h\}$, we denote with $\mathbf{X}_j(t)$ a record of measures $\langle X_1(t), \dots\ X_h(t)\rangle$ for the features in FS_j, synchronised with respect to the timestamp t.

Domain-specific dimensions organise records according to different "facets", such as the observed machine, the tool used during manufacturing, the part program that is being executed by the numerical control of the monitored system. Domain-specific dimensions can be organised in hierarchies: tools can be aggregated into tool types, while monitored physical components (e.g., spindles) can be aggregated into the machines they belong to, in turn organised into plants and enterprises. Therefore, a record $\mathbf{X}_j(t)$ is always associated with: (i) the timestamp at which measures in the record have been collected; (ii) the monitored feature space FS_j; (iii) the values of domain-specific dimensions. Once the feature space and domain-specific dimensions have been fixed, the stream of records over time can be used to monitor the evolution of the feature space for the considered dimensions.

Data summarisation is used here to provide an overall view over a set of records using a reduced amount of information and allows to depict the behaviour of the system better than single records, that might be affected by noise and false outliers. In our approach, data summarisation is based on clustering-based techniques. The application of the clustering algorithm to the stream of records incrementally produces a set of *syntheses* $S = \{s_1, s_2, \ldots, s_n\}$, providing a lossless representation of records.

A synthesis conceptually represents a working behaviour of the monitored system, corresponding to a set of records, with close values for each feature. Please refer to [4] for more details about the incremental clustering algorithm. Formally, we define a synthesis of records as:

$$s_i = \langle id_i, N_i, \mathbf{LS}_i, SS_i, \mathbf{X}_i^0, R_i \rangle \tag{1}$$

where: (i) id_i is the unique identifier of s_i; (ii) N_i is the number of records included into the synthesis; (iii) $\mathbf{LS}_i$ is a vector representing the linear sum of measures in s_i; (iv) SS_i is the quadratic sum of points in s_i for each feature; (v) $\mathbf{X}_i^0$ represents the centroid of the synthesis in the feature space; (vi) R_i is the radius of the synthesis.

The clustering algorithm at a given time t produces a set of syntheses $S(t)$ starting from records collected from timestamp $t - \varDelta t$ to timestamp t and built on top of the previous set of syntheses $S(t - \varDelta t)$ for a given feature space FS_j and domain-specific dimensions. Therefore, we formally define the multi-dimensional model as a set $\mathcal{V}$ of nodes within an hypercube structure, where time, feature spaces and domain-specific dimensions represent hypercube axes and each node $v \in \mathcal{V}$ is described as

$$v = \langle S(t), FS_j, d_1, d_2, \ldots d_p \rangle \tag{2}$$

where $S(t)$ is the set of syntheses at time t, for the feature space FS_j and the values $d_1, d_2, \ldots d_p$ of domain-specific dimensions $\mathcal{D}_1, \ldots \mathcal{D}_p$.

For example, an arbitrary node $v_A = \langle S(t_1), FS_1, m_1, c_2, u_2, pp_a \rangle$, represents the set of syntheses obtained by summarising records collected from time $t_1 - \varDelta t$ to t_1 for machine m_1 (spindle c_2), while using tool u_2 and executing part program pp_a, considering features in the feature space FS_1. Data exploration is performed over dimensions and is guided by data relevance evaluation techniques as described in the following.

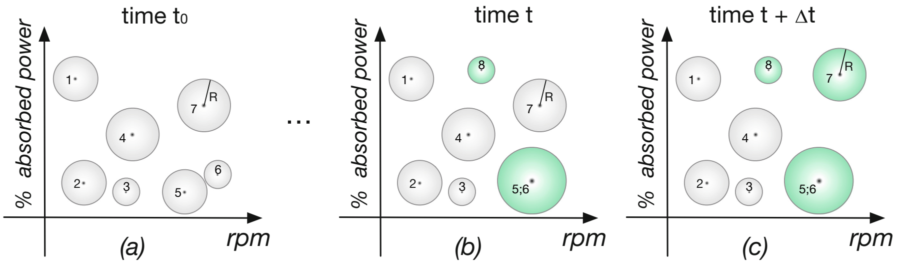


Fig. 2. Evolution of summarised data (syntheses) over time. Feature space and domain-specific dimensions are fixed and not shown here.

5 Relevance-Based Data Exploration

We define data relevance as the *distance* of the physical system behaviour from an *expected status*. This status corresponds to the normal working conditions of the system and is represented by the set of syntheses $\hat{S}(t_0)$. $\hat{S}(t_0)$ can be tagged by the domain experts while observing the monitored system when operates normally. Data relevance at time t is based on the computation of *distance* between the set of syntheses $S(t) = \{s_1, s_2, \ldots, s_n\}$ and $\hat{S}(t_0) = \{\hat{s}_1, \hat{s}_2, \ldots, \hat{s}_m\}$, where n and m represent the number of syntheses in $S(t)$ and $\hat{S}(t_0)$, respectively, and n and m do not necessarily coincide. We denoted this distance with $\mathit{\Delta}(S(t), \hat{S}(t_0))$, computed as:

$$\mathit{\Delta}(\hat{S}(t_0), S(t)) = \frac{\sum_{\hat{s}_i \in \hat{S}(t_0)} d(\hat{s}_i, S(t)) + \sum_{s_j \in S} d(\hat{S}(t_0), s_j)}{m + n} \tag{3}$$

where $d(\hat{s}_i, S(t)) = min_{j=1,\ldots n} d_s(\hat{s}_i, s_j)$ is the minimum distance between $\hat{s}_i \in \hat{S}(t_0)$ and a synthesis in $S(t)$. Similarly, $d(\hat{S}(t_0), s_j) = min_{i=1,\ldots m} d_s(\hat{s}_i, s_j)$. To compute the distance between two syntheses $d_s(\hat{s}_i, s_j)$, we combined different factors: (i) the euclidean distance between syntheses centroids $d_{\boldsymbol{X}_0}(\hat{s}_i, s_j)$, to verify if s_j moved with respect to $\hat{s}_i$ and (ii) the difference between syntheses radii $d_R(\hat{s}_i, s_j)$, to verify if there has been an expansion or a contraction of synthesis s_j with respect to $\hat{s}_i$. Formally:

$$d_s(\hat{s}_i, s_j) = \alpha d_{\boldsymbol{X}_0}(\hat{s}_i, s_j) + \beta d_R(\hat{s}_i, s_j) \tag{4}$$

where α, $\beta \in [0, 1]$ are weights such that $\alpha + \beta = 1$, used to balance the impact of terms in Eq. (4). Weights α and β can be set by operators according to their domain knowledge. For preliminary experiments we equally weighted the two terms of Eq. (4), that is, $\alpha = \beta = \frac{1}{2}$. Future efforts will be devoted to automatically identify the best values to set-up α and β.

Roughly speaking, the relevance techniques allow to identify what are the syntheses that changed over time (namely, appeared, have been merged or removed) for a specific feature space and given values of domain-specific dimensions. Let's denote with $\overline{S}(t) = \{\overline{s}_1, \overline{s}_2, \ldots, \overline{s}_k\}$ such syntheses at time t, where

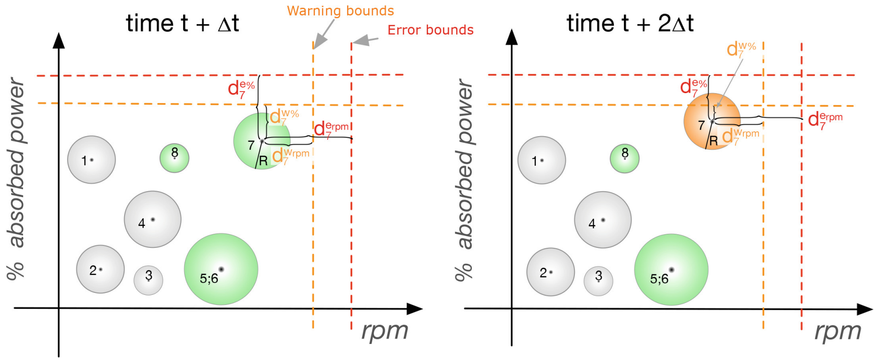


Fig. 3. Anomaly detection through data exploration based on relevance evaluation: data relevance techniques detect changes in syntheses set due to spindle rolling friction torque increase, that may be identified when the rpm value decreases and, at the same time, the power absorption increases.

$k \leq n$ and n is the number of syntheses $\in S(t)$. These syntheses are considered as relevant and will be proposed to the operators to start the exploration. For example, let's consider Fig. 2. Figure 2(a) corresponds to the normal working conditions, as labelled by domain experts according to their expertise, therefore $\hat{S}(t_0) = \{\hat{s}_1, \hat{s}_2, \ldots, \hat{s}_7\}$. At time t, shown in Fig. 2(b), a new synthesis 8 is identified while syntheses 5 and 6 have been merged, that is, $S(t) = \{s_1, s_2, \ldots, s_{[5,6]}, s_7, s_8\}$ and $\overline{S}(t) = \{\overline{s}_{[5,6]}, \overline{s}_8\}$. Finally, in Fig. 2(c) the synthesis 7 moved and $\overline{S}(t + \Delta t) = \{\overline{s}_{[5,6]}, \overline{s}_7, \overline{s}_8\}$.

5.1 Relevance-Based Data Exploration for Anomaly Detection

For anomaly detection purposes, for each synthesis $\overline{s}_c \in \overline{S}(t)$, the distance of synthesis centroid from the warning and error bounds is computed. In the following, we will consider features bounds, but the same considerations hold for the contextual ones. We denote with $\mathbf{d}_c^w$ the record of distances between the centroid of the synthesis $\overline{s}_c$ and the warning bounds and with $\mathbf{d}_c^e$ the record of distances between the centroid of $\overline{s}_c$ and the error bounds. 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. Both $\mathbf{d}_c^w$ and $\mathbf{d}_c^e$ are records having as components the distance for each feature. For example, $d_7^{e\%}$ represents the distance of the centroid of the synthesis 7 from the error bound of the percentage of absorbed power (see Fig. 3). Each relevant synthesis in $\overline{s}_c$ is described as:

$$\overline{s}_c = \langle id_c, N_c, \mathbf{LS}_c, SS_c, \mathbf{X}_c^0, R_c, \mathbf{d}_c^w, \mathbf{d}_c^e \rangle \tag{5}$$

Every Δt seconds, when the syntheses set $S(t)$ is updated, data is analysed to check for anomalies.

For example, in Fig. 3 synthesis 7 moved over time getting closer to the boundaries. Note that distance also helps to detect *potential* state changes. In fact, at time $t + \Delta t$ synthesis 7 still remains inside the wealth zone (`ok` status), but its movement is detected through relevance-based techniques. Therefore, synthesis 7 is recognised as relevant and monitored to promptly detect potential `warning` or `error` status occurrences. After Δt seconds, synthesis 7 moved again and crosses the warning bound of the percentage of absorbed power feature, causing 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 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). When a synthesis moved closer to bounds, the IDEAaS framework reacts by reducing the interval time Δt to check data for anomalies as described in the following.

5.2 Adaptive Relevance Evaluation

The State Detection Service checks the system status by relying on Data Relevance Evaluation Service and after the application of the Data Summarisation Service. If the relevance evaluation detects changes in data compared to the expected working behaviour, the State Detection Service identifies the new status of the system. 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.

Therefore, setup of Δt parameter influences the performances of the system. Small Δt values increase the promptness in identifying relevant syntheses, 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 Sect. 7). The rationale behind our approach is to change Δt as syntheses 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 synthesis $\overline{s}_c \in \overline{S}(t)$ 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 synthesis 7 in Fig. 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 synthesis centroid is between warning bounds and error bounds (see for example synthesis 7 in Fig. 3 at time $t{+}2\Delta t$),

the feature status is set to `warning`, Δt is reduced as $\Delta t = \Delta t(\frac{d_c^{e\text{-}min}}{R} - 1)$ until Δt = minimum value supported by the framework (see experimental evaluation in Sect. 7);

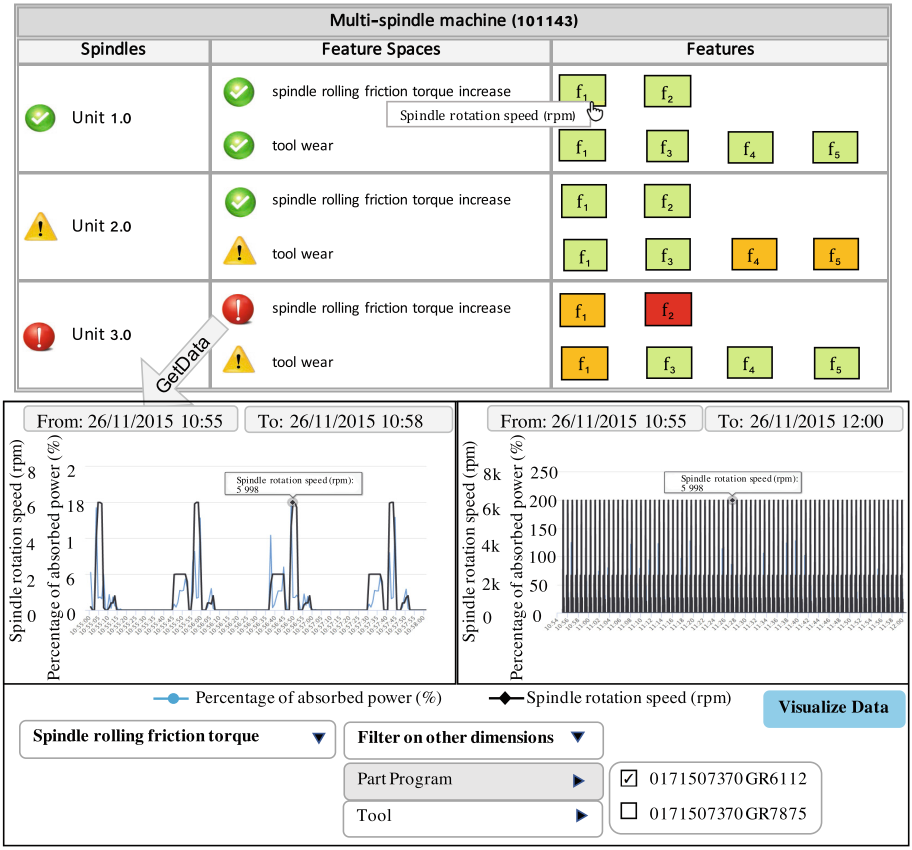
- if $\frac{d_c^{e\text{-}min}}{R} <= 1$ the synthesis centroid is beyond error bounds, the feature status is set to `error`, Δt is set to the minimum supported value (that is, checks are made as more frequently as possible).

6 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 our approach, 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 syntheses set $\overline{S}(t_i)$, processed at time t_i ($t_i <= t$). Each synthesis $\overline{s}_c \in \overline{S}(t_i)$ is marked with the corresponding status and with additional information about whether the synthesis moved, changed (expansion or contraction) or has been removed. All syntheses in $\overline{S}(t_i)$ recognised as anomalous are visualised as shown in Fig. 3.

Exploration of Relevant Syntheses. Once relevant syntheses have been identified, the operator may request to explore in detail records that have been clustered within relevant syntheses. 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 our 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 extracted from the database, when $n >> max_n$ a sampling technique is applied selecting only max_n data among the n data 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 acts 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.



Multi-spindle machine (101143)		
Spindles	Feature Spaces	Features
✓ Unit 1.0	✓ spindle rolling friction torque increase	f_1 f_2
	✓ tool wear	f_1 f_3 f_4 f_5
⚠ Unit 2.0	✓ spindle rolling friction torque increase	f_1 f_2
	⚠ tool wear	f_1 f_3 f_4 f_5
❗ Unit 3.0	❗ spindle rolling friction torque increase	f_1 f_2
	⚠ tool wear	f_1 f_3 f_4 f_5

From: 26/11/2015 10:55 To: 26/11/2015 10:58

From: 26/11/2015 10:55 To: 26/11/2015 12:00

☑ 0171507370GR6112

☐ 0171507370GR7875

Spindle rotation speed (rpm)
GetData
Spindle rotation speed (rpm)
Percentage of absorbed power (%)
Spindle rotation speed (rpm): 5 998
Percentage of absorbed power (%)
Spindle rotation speed (rpm)
Visualize Data
Spindle rolling friction torque ▼
Filter on other dimensions ▼
Part Program ▶
Tool ▶

Fig. 4. Visualisation of relevant data on operator's cockpit in the anomaly detection application scenario (`GetData` method).

Figure 4 shows an implementation 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 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, syntheses calculated for features `"f4"` and `"f5"` are detected as relevant and associated to the `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 detail through the `GetData` method, as shown in Fig. 4. Moreover, the operator may further explore data by setting the time interval of data to be plotted and the other dimensions (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. This value has been chosen considering the device on which the operator is navigating. On the left part of Fig. 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 of data to be visualised is under the max_n value, therefore the sampling techniques are not applied. In the right part of Fig. 4 the operator selected a wider time interval for the same feature space and dimensions, that, in our example 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 is not valuable for the operator.

7 Experimental Evaluation

We performed experiments on the State Detection Service in order to test its performance in terms of processing time and its effectiveness in promptly detecting anomalies. We collected measures from three multi-spindle machines, each of them mounting three spindles. For each spindle the values of 8 features have been collected every 500 ms. Globally we faced an acquisition rate of 144 measures per second. After six months of monitoring on the three machines 630,720,000 measures have been collected. We run experiments on a MacBook Pro mounting MacOS High Sierra, 2.8 GHz Intel Core i7, RAM 16 GB.

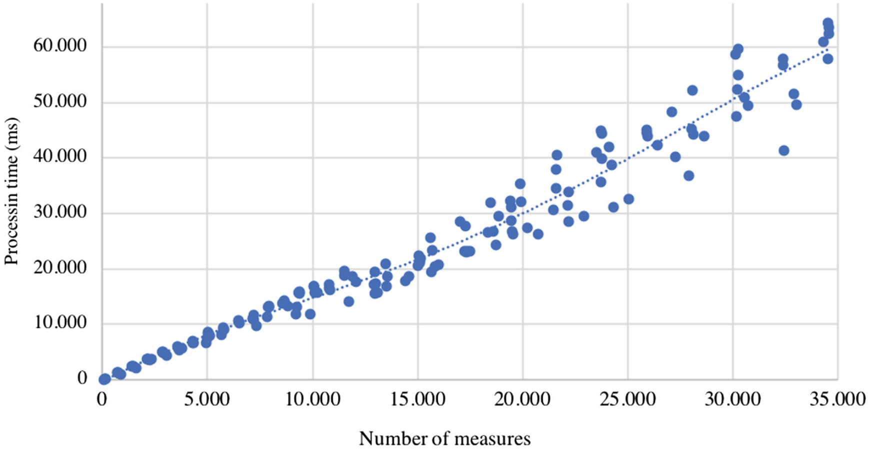


Fig. 5. Response times of the State Detection Service with respect to the number of processed measures.

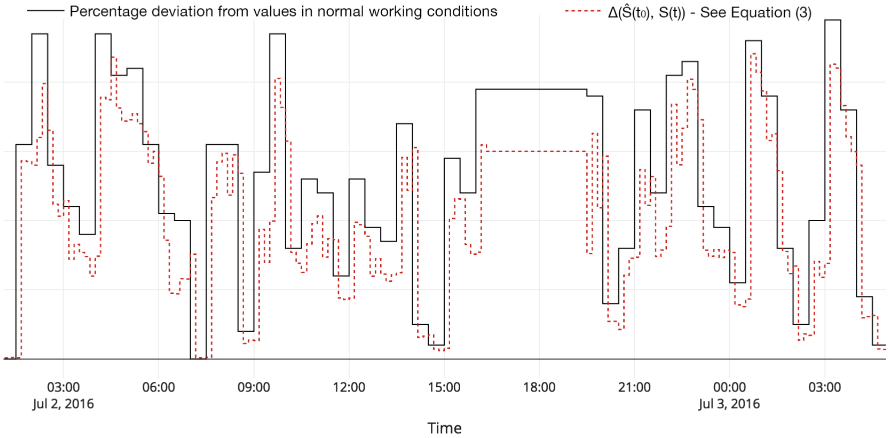


Fig. 6. Correlation between the value of the percentage deviation from the values of rpm and absorbed power features in normal working conditions (black line) and the value of $\Delta(\hat{S}(t_0), S(t))$ computed according to Eq. (3) (dashed red line) (Color figure online).

Figure 5 plots response times 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 Fig. 5 our State Detection Service can process 35000 measures in 60 s on average, corresponding to ~583 measures per second. Therefore, our State Detection Service can successfully cope with the acquisition rate.

To test effectiveness of the service to detect anomalies, we artificially introduced a percentage of values for rpm and absorbed power features with respect to their value in normal working conditions. Further evaluation in an actual production environment with real faults is being performed. Figure 6 shows how our relevance evaluation techniques promptly react to the introduced variations. For this experiment, we set the weights $\alpha = \beta = \frac{1}{2}$ in Eq. (4).

In order to quantify the correlation between the two curves in Fig. 6, we used the Pearson Correlation Coefficient (PCC) $\in [-1, +1]$. In the experiment, the value of PCC is higher than 0.85, that represents a strong correlation.

Figure 7 shows the average time required by the IDEAaS techniques to process a single record for different Δt values. In figure is shown how lower Δt values require more time to process data. In fact, every time clustering is applied, some initializations have to be performed (e.g., opening/closing connection to database, access to the set of syntheses previously computed). Therefore, lower Δt values lead to more frequent initializations. On the other hand, higher Δt values decrease the promptness in identifying anomalous situations, as shown in Fig. 6.

As a final remark, for what concerns the efficacy of the cockpit to support domain experts during data exploration, sampling techniques offer doubtless

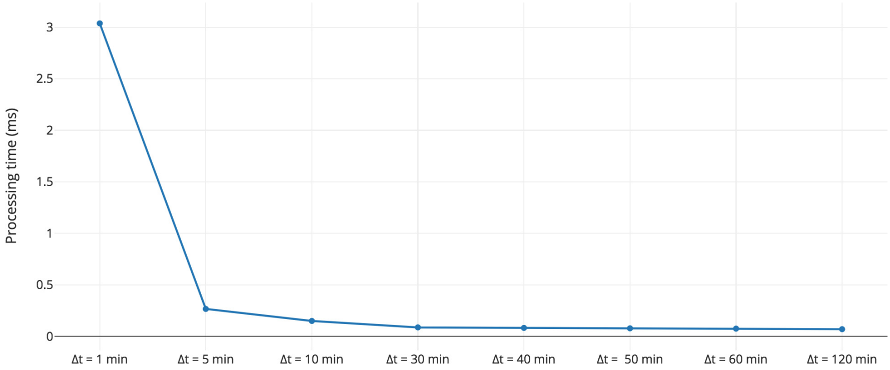


Fig. 7. Response time for IDEAaS algorithm processing time with respect to $\varDelta t$ value.

advantages to ease exploration of data through the proposed implementation of visualisation cockpit. It is straightforward that visualising all the data, without adaptive sampling techniques, is not valuable for the operators and will prevent them to easy inspect and identify incoming anomalies.

8 Related Work

The IDEAaS approach we described in this paper can be classified among approaches that have been proposed to address anomaly detection in presence of big data streams (please refer to [16] 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 framework also differs from solutions for anomaly detection in presence of evolving graphs [10,15], that are characterized by causal/non-causal relationships between measurements.

Among the approaches for anomaly detection on evolving data, the authors in [16] 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 neighbors-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 [8] 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 [9] 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 [2] 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 neighbors-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 [5, 19]. 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 [17] 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 [6] 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 recognized 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, we rely 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 framework from the approaches described in [16] and from traditional Complex Event Processing (CEP) approaches, that are mainly based on pre-defined queries and event detection rules.

9 Conclusions and Future Work

In this paper, we proposed a general-purpose framework that relies on relevant-based data exploration to support domain experts in the inspection and identification of critical situations, out from the large amount of available measure taken from a monitored system. In particular, the framework relies on the combined use of different techniques: (i) an incremental clustering algorithm, to provide summarised representation of collected data; (ii) data relevance evaluation techniques, to attract the experts' attention on relevant data only; (iii) a multi-dimensional organisation of summarised data and adaptive sampling, to enable effective visualisation of data for operators.

The proposed framework has been tested in the Smart Factory context for anomaly detection. Nevertheless, it must be intended as a general approach for Big Data exploration. In fact, the framework can be generalised by defining the dimensions of the multi-dimensional model for different case studies and domains. Summarisation and data relevance evaluation techniques are designed to be applied in any domain that is based on numeric measures collected from a monitored system.

Although preliminary experiments are promising, future development will be focused on further improving the approach using technologies for streaming and parallel processing, such as Spark/Storm. Moreover, the State Detection Service, on which we focused to test the relevance-based data exploration, will be enhanced by introducing pattern recognition techniques to learn from the syntheses evolution. Further usability studies are being performed on the operator's cockpit. This would in principle enable the implementation of health assessment strategies, on top of the ecosystem of services and techniques described in this paper.

References

1. Agrawal, R., Kadadi, A., Dai, X., Andres, F.: Challenges and opportunities with big data visualization. In: Proceedings of the 7th International Conference on Management of Computational and Collective intElligence in Digital EcoSystems (MEDES), pp. 169–173 (2015)
2. Ahmad, S., Lavin, A., Purdy, S., Agha, Z.: Unsupervised real-time anomaly detection for streaming data. Neurocomputing **262**, 134–147 (2017)
3. Bagozi, A., Bianchini, D., De Antonellis, V., Marini, A., Ragazzi, D.: Big data summarisation and relevance evaluation for anomaly detection in cyber physical systems. In: Panetto, H. (ed.) OTM 2017. OTM 2017 Conferences, vol. 10573, pp. 429–447. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-69462-7_28
4. Bagozi, A., Bianchini, D., De Antonellis, V., Marini, A., Ragazzi, D.: Summarisation and relevance evaluation techniques for big data exploration: the smart factory case study. In: Dubois, E., Pohl, K. (eds.) CAiSE 2017. LNCS, vol. 10253, pp. 264–279. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-59536-8_17
5. Cai, L., Thornhill, N.F., Kuenzel, S., Pal, B.C.: Real-time detection of power system disturbances based on k -nearest neighbor analysis. IEEE Access **5**, 5631–5639 (2017)
6. Chenaghlou, M., Moshtaghi, M., Leckie, C., Salehi, M.: Online clustering for evolving data streams with online anomaly detection. In: Phung, D., Tseng, V.S., Webb, G.I., Ho, B., Ganji, M., Rashidi, L. (eds.) PAKDD 2018. LNCS (LNAI), vol. 10938, pp. 508–521. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-93037-4_40
7. Gorecky, D., Schmitt, M., Loskyll, M., Zuhlke, D.: Human-machine interaction in the industry 4.0 era. In: IEEE International Conference on Industrial Informatics (INDIN), pp. 289–294 (2014)
8. Hanamori, T., Nishimura, T.: Real-time monitoring solution to detect symptoms of system anomalies. FUJITSU Sci. Tech. J. **52**, 23–27 (2016)
9. Huber, M., Voigt, M., Ngomo, A.C.N.: Big data architecture for the semantic analysis of complex events in manufacturing, pp. 353–360 (2016)
10. Koutra, D., Shah, N., Vogelstein, J.T., Gallagher, B., Faloutsos, C.: DELTACON: principled massive-graph similarity function with attribution. ACM Trans. Knowl. Discov. Data **10**(3), 28:1–28:43 (2016)
11. Lee, J., Ardakani, H., Yang, S., Bagheri, B.: Industrial big data analytics and cyber-physical systems for future maintenance and service innovation. In: Proceedings of Conference on Intelligent Computation in Manufacturing Engineering (CIRP), vol. 38, pp. 3–7 (2015)

12. Lee, J., Lapira, E., Bagheri, B., Kao, H.: Recent advances and trends in predictive manufacturing systems in big data environment. Manuf. Lett. **1**(1), 38–41 (2013)
13. Lopez, F., et al.: Categorization of anomalies in smart manufacturing systems to support the selection of detection mechanisms. IEEE Robot. Autom. Lett. **2**(4), 1885–1892 (2017)
14. Nunes, D., Silva, J.S., Boavida, F.: A Practical Introduction to Human-in-the-Loop Cyber-Physical Systems. Wiley IEEE Press, Hoboken (2018)
15. Rashidi, L., et al.: Node re-ordering as a means of anomaly detection in time-evolving graphs. In: Frasconi, P., Landwehr, N., Manco, G., Vreeken, J. (eds.) ECML PKDD 2016. LNCS (LNAI), vol. 9852, pp. 162–178. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46227-1_11
16. Salehi, M., Rashidi, L.: A survey on anomaly detection in evolving data: [with application to forest fire risk prediction]. SIGKDD Explor. Newsl. **20**(1), 13–23 (2018)
17. Stojanovic, L., Dinic, M., Stojanovic, N., Stojadinovic, A.: 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), pp. 1647–1652 (2016)
18. Wongsuphasawat, K., Moritz, D., Anand, A., Mackinlay, J., Howe, B., Heer, J.: Voyager: exploratory analysis via faceted browsing of visualization recommendations. IEEE Trans. Vis. Comput. Graph. **22**(1), 649–658 (2016)
19. Zhang, L., Lin, J., Karim, R.: Adaptive kernel density-based anomaly detection for nonlinear systems. Knowl.-Based Syst. **139**, 50–63 (2018)