Journal of Manufacturing Systems 43 (2017) 214–224

Contents lists available at ScienceDirect

Journal of Manufacturing Systems

journal homepage: www.elsevier.com/locate/jmansys

Technical Paper

Interpretative identification of the faulty conditions in a cyclic manufacturing process

Dominik Kozjek[a,*], Rok Vrabič[a], David Kralj[b], Peter Butala[a]

[a] Faculty of Mechanical Engineering, University of Ljubljana, Aškerčeva 6, Ljubljana, Slovenia
[b] ETI Elektroelement d.d., Obrezija 5, Izlake, Slovenia

ARTICLE INFO

Article history:
Received 28 September 2016
Received in revised form 24 February 2017
Accepted 3 March 2017
Available online 22 March 2017

Keywords:
Production process
Fault identification
Root cause analysis
Decision support
Big Data

ABSTRACT

The intensive development of information and communication technologies in recent years has led to an increase in data size and complexity. Conventional approaches, with associated methods of analysis based on descriptive and inductive statistics, may no longer be suitable for extracting the valuable information that is hidden in the available data.

Computer-controlled manufacturing systems are becoming rich sources of data. Plastic injection moulding and die casting systems are typical examples of such manufacturing systems where the parts are produced by repeating the same sequence of steps that make up a manufacturing cycle. For each cycle, similarly structured data is generated.

In this work a method for systematic data analysis for cyclic manufacturing processes is presented. The proposed data-analysis method integrates well-known heuristic algorithms, i.e., decision trees and clustering, with the purpose of identifying types of faulty operating conditions. The result of the analysis is an interpretable model for decision support that can be used for fault identification, to search for root causes, and to develop prognostic systems. A holistic approach of applying the proposed data-analysis method, along with suggestions and guidelines for implementation, is presented. A case study is presented in which the proposed method is applied to real industrial data from a plastic injection-moulding process.

1. Introduction

With the development of technology, the complexity of manufacturing systems is steadily increasing. In turn, the complexity of generated data is also increasing, which is reflected in incompleteness and inaccuracy of data, heterogeneous and dynamically changing data structures, large and fast increasing data volumes, etc. All these facts make analysis of manufacturing data for extracting useful information and knowledge very challenging.

For the above reasons, advanced online and real-time monitoring, identification and prognostics are becoming increasingly difficult. Insufficient data quality, lack of holistic databases that integrate operational and decision-support data, and a lack of models for analysis are some of the recent, most pressing challenges in industry [1]. In the future, new applications for management and use of data will need to be provided [2].

Current issues related to data management have led to the formation of what is known as the concept of Big Data. Big Data is generally related to the interplay of information- and communication-technology developments that enables an increase in storage capacities and computing power, as well as the advanced integration of traditional and novel methods for the acquisition, storage, integration, analysis, modelling, visualisation, and other kinds of management of data that is too large or too complex to allow the extraction of value using traditional approaches [1–6]. Data complexity can arise due to diversity, variability, data distribution, lack of structure, shortcomings, non-credibility, speed, or automation requirements [7,8]. A typical Big Data system is divided into four stages that sequentially form the value chain: (1) generation, (2) acquisition, (3) storage, and (4) analysis [9,10]. Each of these phases includes elements, techniques, tools, methods and concepts, which conjointly form efficient systems for extracting the value from large and complex data.

Heuristic methods of analysis turn out to be very efficient for fault-diagnosis problems in complex manufacturing processes, where the states are described by complex combinations of many parameters. Recently, in the manufacturing domain, a lot of research was carried out regarding fault-diagnostics problems by

* Corresponding author.
E-mail addresses: dominik.kozjek@fs.uni-lj.si (D. Kozjek), rok.vrabic@fs.uni-lj.si (R. Vrabič), david.kralj@eti.si (D. Kralj), peter.butala@fs.uni-lj.si (P. Butala).

http://dx.doi.org/10.1016/j.jmsy.2017.03.001
0278-6125/

incorporating machine-learning methods, data mining, and other evolving techniques, targeting the problems of sensing and describing multiple and dynamic faults, monitoring, real-time processing, searching for the best features, failure-condition identification, early prediction of failures, and assessing the severity of failures [11].

Many of so-far untouched challenges and opportunities remain for the development and introduction of useful heuristic techniques in manufacturing systems that would improve the process quality through an intelligent use of the available data [12].

The paper presents a holistic data-analysis method for the interpretative identification of faulty conditions in a cyclic manufacturing process (IIFC). The objective here is to demonstrate the usability of large volumes of data generated during a widespread type of the so called cyclic manufacturing processes, such as plastic injection moulding, for extracting valuable information and new knowledge models for better understanding of the manufacturing process and its faulty operating conditions. The proposed approach uses well known data-mining methods and addresses the arising issues of increasing data size and complexity.

Cyclic-manufacturing processes are defined as manufacturing processes in which the parts are produced repeating the same sequence of steps that forms the manufacturing cycle. The data that is generated in these processes describes the state of the process for each cycle. This is suitable for applying machine-learning and other advanced techniques of data analysis, because minimal effort is required to define the common features for each cycle due to the repetitive nature of the cycle's steps. Although machine-learning techniques are well known, relatively little effort has been made towards an effective implementation in real, every-day cases in manufacturing systems.

The proposed IIFC method is intended to systematically analyse the empirical data that are usually generated in the cyclic manufacturing process. The aim of applying the IIFC method is to improve the quality of an already well-optimised process through the identification of rare faulty operating conditions, and by learning about their types and characteristics. A two-phase data-analysis workflow is suggested. In the first phase, rules,[1] describing the process conditions are extracted with the use of a decision-tree heuristic algorithm and the expert knowledge of the observed manufacturing process. Bit vectors (containing only 0 and 1) are obtained, describing the combination of rules that are true (1) or false (0) for each faulty cycle of the manufacturing process. In the second phase, different types of faulty conditions are revealed and described using a clustering technique and the extracted rules. The output of IIFC analysis is an interpretative model, which is understandable to people such as operators, process engineers, plant supervisors, etc., that interact daily with the observed manufacturing process.

The paper is structured as follows. Section 2 describes the proposed IIFC method. The IIFC method's role in a general workflow of data-analytics, a holistic description of the combination and the sequence of steps that form the workflow of the IIFC method, including suggestions for how to organise and apply the data as well as the properties related to implementation and applicability of the IIFC method, are given. The content of this paper continues with an application of the proposed method in a real industrial scenario of plastic injection moulding. The procedure for applying the method is presented together with the results and an interpretation that demonstrates the usability in a real industrial environment.

2. Interpretative identification of faulty conditions

The proposed IIFC method suggests a combination and sequence of data-analysis steps together with guidelines for applying them to usually available data of cyclic-manufacturing processes.

The role of the proposed data-analysis steps of the IIFC method according to the general workflow of data analytics is shown in Fig. 1. Heterogenous data can be generated at different locations by work systems, the manufacturing execution system, etc. Using appropriate methods for data acquisition, we can reduce the data size and dimensionality, with a tendency to minimise the loss of information. Large amounts of data are then managed by advanced databases and cloud-computing systems that enable high throughput, reliability, and availability [9,10].

The object of the IIFS analysis is a set of manufacturing process cycles. A cycle can be *normal*, *faulty* or *other*. The method assumes that the cycle can be faulty for various reasons, e.g., deviations from the required dimensional tolerances of the products, an unplanned machine stop, etc. Other cycles are those that do not belong to the group of normal cycles nor to the group of faulty ones, e.g., cycles that occur in the vicinity of the faulty ones, cycles with missing data, etc. Each cycle c_i is described in terms of a unique ID, a date and time, a work system, process parameters, etc. (Eq. (1)).

$$\begin{aligned} c_i = \{ & c_i^{id}, c_i^{dateTime}, c_i^{workSystem}, \ldots, c_i^{processParameter1}, c_i^{processParameter2}, \\ & c_i^{processParameter3}, \ldots, c_i^{inputMaterial}, \ldots, c_i^{machineAlarm1}, \\ & c_i^{machineAlarm2}, \ldots, c_i^{outsideHumidity}, c_i^{outsideTemperature}, \\ & c_i^{workingShift}, \ldots \} \end{aligned} \quad (1)$$

The approach of IIFC method is (1) to describe each faulty cycle with a combination of (a) *fault-specific rules*,[2] i.e., rules that describe faulty operating conditions, and (b) *other rules*,[3] i.e., rules that are not necessarily related to the physics of the manufacturing process or faults, but their inclusion could potentially lead to the discovery of the root causes for the emergence of faulty operating conditions, (2) to reveal similar subgroups of faulty cycles based on combinations of fault-specific rules and (3) to create detailed descriptions of these identified subgroups based on all extracted rules.

Data that are usually available are in different ways related to the manufacturing process and the faulty conditions, thus they need to be treated differently when extracting the rules which are used for the identification and the description of the faulty conditions. The data types (*id*, *dateTime*, *workSystem*, etc.) are of different kinds, e.g., integers, real values, descriptors, etc., and are classified into four groups: (1) metadata (*G*1), (2) process-specific (*G*2), (3) fault-specific (*G*3) and (4) other (*G*4) data. Metadata (*G*1) are used to support computing operations and enables efficient data storage and retrieval. From process-specific (*G*2) and fault-specific (*G*3) data, fault-specific rules which are used to identify different types of faulty operating conditions, are extracted. To describe properties of identified faulty-condition types, fault-specific rules, together with other rules extracted from other (*G*4) data, are used. Classification of input data types is presented in Table 1 and explained below.

Metadata (G1). Group *G*1 includes the data that identify the manufacturing cycle time- and location-wise. The unique ID of the cycle, the date and time of the cycle, the work system, and the cycle quality (*normal*/*faulty*/*other*), etc. are the types of data that belong to this group. A unique ID is given to a cycle in the step of data storage. A

[1] E.g., "*Temperature of a cylinder exceeded 251 °C.*", "*Day of the week is Saturday.*", etc.

[2] E.g., "*Temperature of a cylinder exceeded 251 °C.*"

[3] E.g., "*Day of the week is Saturday.*"

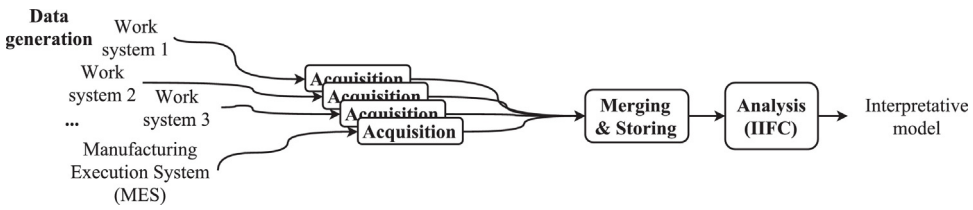

Fig. 1. General workflow of data analytics.

Table 1
Classification of input data types.

Metadata (*G*1)	Process-specific (*G*2)	Fault-specific (*G*3)	Other (*G*4)
id	*processParameter*1	*machineAlarm*1	*dateTime*
dateTime	*processParameter*2	*machineAlarm*2	*machineID*
workSystem	*processParameter*3	*machineAlarm*3	...
quality	*processParameter*4	...	
...	...	*defectDescription*	
	inputMaterial	...	
	...		

well-thought-out prescription of the cycle's ID can enable efficient processing in the subsequent steps of the data analysis.

The cycles' c_i^{id} are integer values defined sequentially for each individual work system (Eq. (2)).

$$\forall i,j \in [1,N] \wedge \forall k,l \in [1,NWS]:\ (c_i^{workSystem} = ws_k \wedge c_j^{workSystem} = ws_l \wedge k < l) \Rightarrow c_i^{id} < c_j^{id} \quad (2)$$

where $\mathbf{WS} = [ws_1, ws_2, ws_3, \ldots, ws_{NWS}]$ is the set of work systems and *NWS* is the number of all work systems. Within each range for the individual work system, the c_i^{id} values are consecutively arranged according to the chronological sequence of cycles (Eq. (3)).

$$\forall i,j \in [1,N]:\ (c_i^{dateTime} < c_j^{dateTime} \wedge c_i^{workSystem} = c_j^{workSystem}) \Rightarrow c_i^{id} < c_j^{id} \quad (3)$$

Process-specific (G2) data. Group *G*2 contains the data that are directly connected to the process, but from which faulty conditions cannot be directly revealed, e.g., numerical values of the process parameters, categorical or numerical data of the input material, etc.

Fault-specific (G3) data. Typical examples of fault-specific (*G*3) data are warnings (alarms) from the machine's controller, descriptions of product defects, etc. The fault-specific (*G*3) data are obviously different for cycles that occurred in faulty operating conditions.

Other (G4) data. These data are included in the analysis for the purpose of identifying the relations from the data that are not necessarily related to the physics of the manufacturing process, but their inclusion could potentially lead to discovering the root causes of the emergence of faulty operating conditions.

2.1. Workflow of the IIFC analysis

Fig. 2 shows a diagram of the workflow for the analysis. The input dataset *cycles* consists of an ordered series of data for individual cycles (Eq. (4)).

$$cycles = \{c_1, c_2, c_3, \ldots, c_i, \ldots, c_N\} \quad (4)$$

From process-specific (*G*2), fault-specific (*G*3) and other (*G*4) data, sets E^{G2}, E^{G3} and E^{G4} respectively, are generated. Sets E^{G2}, E^{G3} and E^{G4} consist of bit vectors,[4] which denote the combination of rules for each faulty cycle. Based on the combinations of fault-specific rules (E^{G2} and E^{G3}), homogenous groups of faulty cycles (η_1, η_2, η_3, etc.) are hierarchically revealed in the step *identification of faulty operating-condition types*. In the final step – *enriching the hierarchic model*, the identified hierarchy of faulty-cycle groups *H* is enriched with the faulty-type characteristics, i.e., the relative frequencies of rules ($\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \boldsymbol{\alpha}_3$, etc.). The output of the analysis is the *interpretative model* in a form of a hierarchic categorisation of faulty-cycle groups representing faulty-condition types, with corresponding characteristic combinations of rules.

The analysis workflow consists of two general phases. In the first phase, combinations of rules for faulty cycles are extracted. The second phase includes the identification and description of different types of faulty conditions. The second phase is based on the idea of *predictive clustering trees* (PCTs).

PCTs are partitions of the data into a cluster hierarchy with respect to the number of observable properties [13]. This method has been successfully applied to conceptual clustering, the simultaneous prediction of multiple parameters and ranking tasks. The general idea is to recursively partition a set of data into clusters, with the tendency to minimise intra-cluster variation. The algorithm for inducing PCTs is a standard TDIDT (Top-Down Induction of Decision Trees) [14].

Below, a more detailed description of the analysis' steps is given.

2.1.1. Transformation G2

The purpose of this step is to transform process-specific (*G*2) data to a set of bit vectors that describe the combination of fault-specific rules for each faulty cycle. This transformation consists of two successive parts: (1) *calculation of features* and (2) *heuristic extraction of rules*.

Calculation of features. The cycles are performed one after the other on a work system. The process state of the current cycle may reflect the previous one or be reflected in the subsequent ones. The features should therefore contain some information about the previous or subsequent cycles. It is important that the observed process is well understood in order to define the features that reflect faults and anomalies. In the case of numerical values (e.g., process parameters, such as temperatures, pressures, and speeds), the features could be the process parameter values of the observed cycle, local rises or falls in average values, local changes of standard deviations, slopes of rising and falling, etc.

On the basis of process-specific (*G*2) data, a set of feature vectors *X* and a corresponding set of target-class values *Y*, are generated (Eqs. (5) and (6)).

$$X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \ldots, \boldsymbol{x}_n, \ldots\} \quad (5)$$

$$Y = \{y_1, y_2, y_3, \ldots, y_n, \ldots\} \quad (6)$$

Each feature vector $\boldsymbol{x}_n$ consists of features ($x_{n,1}, x_{n,2}, x_{n,3}$, etc.), which on the basis of process-specific (*G*2) data describe the process conditions at the cycle c_i (Eq. (7)). Each feature vector $\boldsymbol{x}_n$ is associated

[4] *Bit vector* – an ordered set of values that can be either 0 or 1.

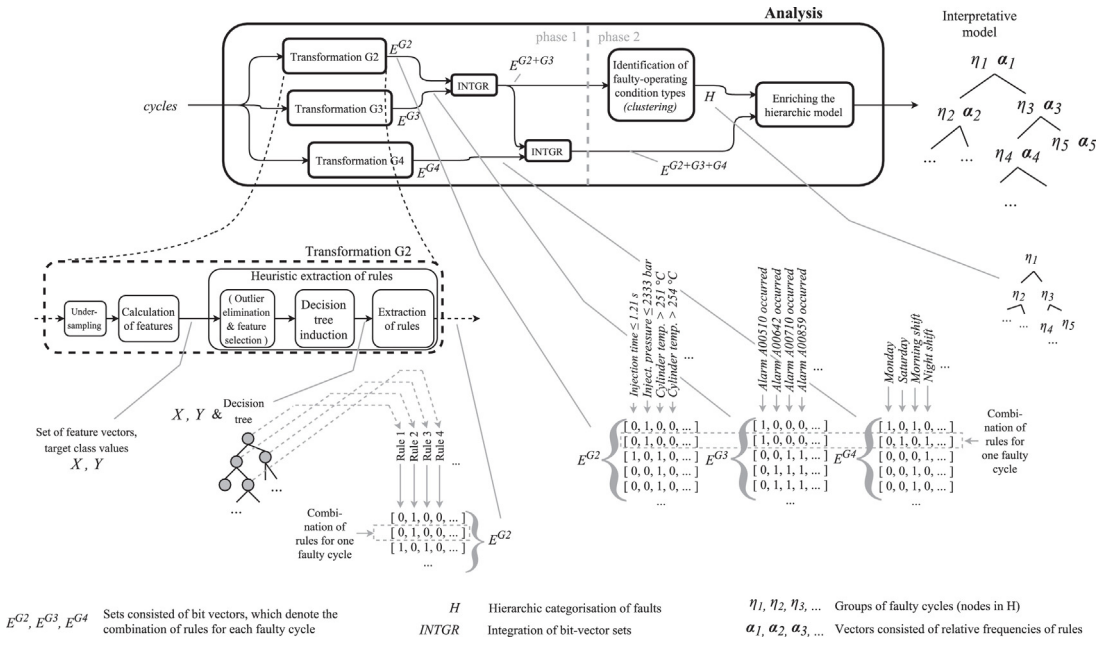

Fig. 2. Workflow of IIFC analysis.

with a target class y_n, which denotes whether the feature vector corresponds to a normal ($y_n = 0$) or a faulty ($y_n = 1$) cycle (Eq. (7)).

$$\boldsymbol{x}_n = [x_{n,1}, x_{n,2}, x_{n,3}, \ldots] \rightarrow y_n \tag{7}$$

Features of the vector $\boldsymbol{x}_n$ are determined by the function f^{fv}, which takes the c_i^{id} value of the corresponding cycle c_i as an input (Eq. (8)).

$$\boldsymbol{x}_n = f^{fv}(c_i^{id}) \tag{8}$$

When calculating the features which require the data of previous or subsequent cycles on the work system, the given prescription of c_i^{id} values (as suggested in Eqs. (2) and (3)) enables an efficient access to the data.

Due to an expected imbalance between the numbers of normal and faulty cycles and a large size of the input dataset, the *undersampling*[5] method on the group of normal cycles is suggested to be used before the step *calculation of features*.

Heuristic extraction of rules. Set of feature vectors X and corresponding target class values Y are with the use of heuristic algorithms transformed to a set E^{G2}, which consists of combinations of fault-specific rules for each faulty cycle (Eq. (9)).

$$E^{G2} = \{\boldsymbol{e}_1^{G2}, \boldsymbol{e}_2^{G2}, \boldsymbol{e}_3^{G2}, \ldots, \boldsymbol{e}_k^{G2}, \ldots\}, \tag{9}$$

where $\boldsymbol{e}_k^{G2}$ is a bit vector, that determines the combination of corresponding rules for the k-th faulty cycle (Eq. (10)).

$$\boldsymbol{e}_k^{G2} = [e_{k,1}^{G2}, e_{k,2}^{G2}, e_{k,3}^{G2}, \ldots, e_{k,j}^{G2}, \ldots]; \quad e_{k,j}^{G2} \in \{0, 1\} \tag{10}$$

The value of $e_{k,j}^{G2}$ determines whether j-th rule corresponds ($e_{k,j}^{G2} = 1$) or does not correspond ($e_{k,j}^{G2} = 0$) to the k-th faulty cycle.

Fault-specific rules can be extracted from the *decision-tree* model, induced by a heuristic to distinguish faulty examples from the normal ones. *Decision-tree* algorithms can handle both categorical and numerical data, and they are normally used for classification tasks. Due to the relatively good interpretative ability of the induced models, they can be used to define anomalies corresponding to faulty process conditions in the form of a combinations of rules

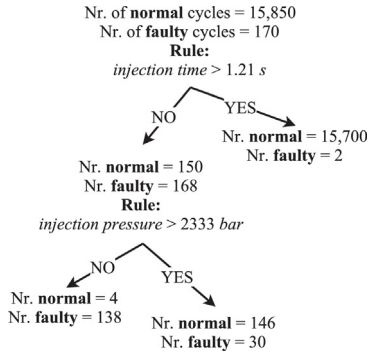

Fig. 3. Decision tree example.

derived from the nodes of the decision-tree. An example which indicates how the rules can be extracted from the structure of the decision tree is given below.

Fig. 3 shows an example of a decision tree. The numerical values of process parameters *injection time* and *injection pressure* are used to induce the model. Extracted rules from the nodes of the decision tree would be (1) "*injection time* $\leq$ 1.21 *seconds*" $\rightarrow e_{k,1}^{G2}$ and (2) "*injection pressure* $\leq$ 2333 *bar*" $\rightarrow e_{k,2}^{G2}$. If at k-th faulty cycle the *injection time* $\leq$ 1.21 *s* then $e_{k,1}^{G2} = 1$, else $e_{k,1}^{G2} = 0$, and if the *injection pressure* $\leq$ 2333 *bar* then $e_{k,2}^{G2} = 1$, else $e_{k,2}^{G2} = 0$.

When implementing the *decision-tree* algorithms, the appropriate method of limiting the size of the decision tree to prevent over-fitting should be used. The expected imbalance (the ratio of normal to faulty cycles) of the training dataset should be taken into account. The performance of model may be improved using the methods of *outlier elimination* on the set of *normal* examples and *feature selection* before the model induction procedure.

2.1.2. Transformation G3

Because fault-specific ($G3$) data directly correspond to faulty conditions, there is no need to perform the above-described extraction of rules, which is based on the differences between normal and faulty conditions. The output of *transformation* $G3$ is a set E^{G3} that

[5] *Undersampling* as a data-analysis technique to reduce the imbalance of classes by under-sampling the majority class.

holds the combinations of fault-specific rules ($\boldsymbol{e}_1^{G3}, \boldsymbol{e}_2^{G3}, \boldsymbol{e}_3^{G3}$, etc.) of individual faulty cycles (Eqs. (11) and (12)).

$$E^{G3} = \{\boldsymbol{e}_1^{G3}, \boldsymbol{e}_2^{G3}, \boldsymbol{e}_3^{G3}, \ldots, \boldsymbol{e}_k^{G3}, \ldots\} \tag{11}$$

$$\boldsymbol{e}_k^{G3} = [e_{k,1}^{G3}, e_{k,2}^{G3}, e_{k,3}^{G3}, \ldots, e_{k,j}^{G3}, \ldots]; \quad e_{k,j}^{G3} \in \{0, 1\} \tag{12}$$

Each bit vector $\boldsymbol{e}_k^{G3}$ corresponds to the same k-th faulty cycle as $\boldsymbol{e}_k^{G2}$ (in Eq. (10)). How exactly this transformation is conducted depends on the case. Features must be defined by incorporating knowledge about the observed manufacturing process, e.g., we need to include descriptions about what has been happening at the work system in the recent history of the observed cycle. A rule that is based on fault-specific ($G3$) data can be, e.g., "*Alarm about the lack of input material has occurred 1 cycle before the observed one*."

2.1.3. Transformation G4

The output of this transformation is the set E^{G4}, which consists of combinations of rules that are derived from the other ($G4$) data (Eqs. (13) and (14)).

$$E^{G4} = \{\boldsymbol{e}_1^{G4}, \boldsymbol{e}_2^{G4}, \boldsymbol{e}_3^{G4}, \ldots, \boldsymbol{e}_k^{G4}, \ldots\} \tag{13}$$

$$\boldsymbol{e}_k^{G4} = [e_{k,1}^{G4}, e_{k,2}^{G4}, e_{k,3}^{G4}, \ldots, e_{k,j}^{G4}, \ldots]; \quad e_{k,j}^{G4} \in \{0, 1\} \tag{14}$$

Each bit vector $\boldsymbol{e}_k^{G4}$ corresponds to the same k-th faulty cycle as $\boldsymbol{e}_k^{G2}$ and $\boldsymbol{e}_k^{G3}$ (in Eqs. (10) and (12)). Set E^{G4} can be provided similarly as described above for the *transformation G2* or *transformation G3*. Rules, derived from other ($G4$) data are, e.g., "*Day of the week = Saturday.*", "*Working shift = night shift.*", etc.

2.1.4. Identification of faulty-operating condition types

In this step, E^{G2+G3} (joint sets E^{G2} and E^{G3}) is the input set, in which an individual bit vector $\boldsymbol{e}_k^{G2+G3}$ is a combination of fault-specific rules of an individual faulty cycle (Eqs. (15) and (16)).

$$E^{G2+G3} = \{\boldsymbol{e}_1^{G2+G3}, \boldsymbol{e}_2^{G2+G3}, \boldsymbol{e}_3^{G2+G3}, \ldots, \boldsymbol{e}_k^{G2+G3}, \ldots\} \tag{15}$$

$$\boldsymbol{e}_k^{G2+G3} = [e_{k,1}^{G2}, e_{k,2}^{G2}, e_{k,3}^{G2}, \ldots, e_{k,1}^{G3}, e_{k,2}^{G3}, e_{k,3}^{G3}, \ldots] \tag{16}$$

The input set E^{G2+G3} is hierarchically divided into subgroups using a *clustering* algorithm. Before applying the *clustering* algorithm, all of the faulty-cycle examples are contained in one group. This initial group is repeatedly divided into two subgroups using the *clustering* algorithm. Without a stopping criterion, the number of the identified subgroups would grow until each instance of a faulty cycle belonged to a separate group. An appropriate stopping criterion is needed to avoid the excessive growth of the clustering tree. Statistical methods can be used to test the equality of the groups in order to solve this problem: during each iteration of the separation, it is first checked whether the separated groups would differ significantly by at least one rule, and if they do, the separation is considered meaningful. Otherwise, the growth of the current branch is finished before the separation is performed. The rules can be differently weighted, e.g., according to their relative importance. The output of this step is a hierarchic categorisation H of nodes[6] η_1, η_2, η_3, etc., representing different types of faulty conditions.

2.1.5. Enriching the hierarchic model

The role of this step is to reveal the characteristics of the identified faulty-condition types in a form of combinations of frequent rules for individual groups of faulty cycles. This is done with a use of an integrated set $E^{G2+G3+G4}$ (Eqs. (17) and (18)) and hierarchical categorisation of faulty-cycle groups H.

$$E^{G2+G3+G4} = \{\boldsymbol{e}_1^{G2+G3+G4}, \boldsymbol{e}_2^{G2+G3+G4}, \boldsymbol{e}_3^{G2+G3+G4}, \ldots, \boldsymbol{e}_k^{G2+G3+G4}, \ldots\} \tag{17}$$

$$\boldsymbol{e}_k^{G2+G3+G4} = [e_{k,1}^{G2}, e_{k,2}^{G2}, e_{k,3}^{G2}, \ldots, e_{k,1}^{G3}, e_{k,2}^{G3}, e_{k,3}^{G3}, \ldots, e_{k,1}^{G4}, e_{k,2}^{G4}, e_{k,3}^{G4}, \ldots] \tag{18}$$

Knowing the hierarchic categorisation of groups of faulty cycles H and integrated combinations of rules $E^{G2+G3+G4}$, specific characteristics of identified faulty-condition types can be revealed as combinations of rules, which have high values of *relative frequency* for the faulty cycles inside an individual group of revealed hierarchic categorisation. Thus, for all identified groups of faulty cycles (η_1, η_2, η_3, etc.), vectors of relative frequencies ($\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \boldsymbol{\alpha}_3$, etc.) are calculated (Eq. (19)).

$$\boldsymbol{\alpha}_i = [\alpha_{i,1}, \alpha_{i,2}, \alpha_{i,3}, \ldots, \alpha_{i,j}, \ldots] \rightarrow \eta_i \quad , \tag{19}$$

where $\boldsymbol{\alpha}_i$ is the vector of relative frequencies for the i-th group of faulty cycles (η_i) and $\alpha_{i,j}$ is the relative frequency of j-th rule for the i-th group of faulty cycles. The final *interpretative model* is the hierarchic categorisation H of faulty-condition types (η_1, η_2, η_3, etc.) with corresponding characteristics in a form of vectors of relative frequencies of rules ($\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \boldsymbol{\alpha}_3$, etc.).

2.2. Implementation and applicability

The operational properties of this method support the use of the benefits that are derived from typical Big Data technologies for storing and retrieving the data – the benefits of high-speed querying the data from large quantities, with the advantage of not keeping all the observed data in the computer memory at once while performing memory-critical steps of the analysis. The IIFC method can be used in an adaptive manufacturing system to predict faulty conditions thanks to a high degree of autonomy. This could contribute to an improvement in the manufacturing process performance through additional decision support by avoiding faulty cycles, preparing early action plans, or determining preventative actions in real time.

IIFC as a holistic approach, including suggestions on the way of organising and applying commonly available data, is directly applicable to cyclic manufacturing processes such as plastic injection moulding and die casting. Applying the IIFC method to some other manufacturing process[7] requires the redefinition of the object of analysis, and a more comprehensive overview and rethinking about the way of organising and applying the data under consideration.

3. Case study of plastic injection moulding

The proposed IIFC method is applied to industrial data collected from a plastic injection moulding (PIM) process.

3.1. Plastic injection moulding

PIM is a widespread manufacturing process for the production of plastic products. It is a cyclic manufacturing process in which a hot melt is injected into the tool[8] under high pressure. Finally, the injected material is cooled and ejected from the tool [15].

Unplanned machine stops (UMSs) are highly undesirable during the process. UMS is an unplanned process interruption that causes reduced availability of the equipment. UMSs indirectly

[6] Groups of faulty cycles.

[7] E.g., some more general discrete manufacturing process or continuous-type manufacturing process.

[8] Also called the *mould*.

Table 2
Records of process parameters of PIM.

Date	Time	Cycle no.	Cycle time (s)	Inject. time (s)	Max. inject. pressure (bar)	...
...	...	...	...	...	...	...
12/08/15	11:01:48	3922	21.84	1.35	2553	...
12/08/15	11:02:10	3923	21.83	1.35	2553	...
12/08/15	11:02:32	3924	21.81	1.34	2553	...
12/08/15	11:02:54	3925	21.84	1.35	2553	...
12/08/15	11:03:16	3926	21.83	1.35	2553	...
...	...	...	...	...	...	...

Table 3
Records of alarms occurred on a PIM system.

Date	Time	Cycle no.	Alarm code	...
...	...	...	...	...
18/07/15	04:44:50	1985	A 00510 Z1	...
18/07/15	04:44:58	1985	A 00510 Z11	...
18/07/15	04:45:33	1985	A 00510 Z3	...
18/07/15	04:45:56	1985	A 00510 Z2	...
...	...	...	...	...
19/07/15	06:48:43	2003	A 00145	...
19/07/15	07:01:36	2038	A 00306	...
...	...	...	...	...

affect the quality of the process. When a UMS occurs, the process needs to be restarted. The start-up pieces are usually faulty and therefore discarded, the variability of the product's characteristics is increased, and some time is needed to restore satisfactory operating conditions.

There are several reasons why UMSs occur. While the information about whether there were some UMSs is directly revealed from the empirical data, the information about the types and the root causes is hidden in the data and turns out to be complex to reveal.

The objective is to minimise the number of UMSs. This can be achieved by knowing the root causes or anomalies that are reflected in the given data. The IIFC method is used with the aim of revealing the types and the root causes of the UMSs.

3.2. Data description

Data are derived from a successful plastic-parts producer. They are obtained from machine controllers and from the Manufacturing Execution System (MES). In the observed period of 6 months and on the five observed, modern, PIM machines, approximately 2.2 million cycles of 83 different products were recorded. All these records are collected and merged into a dataset on which the IIFC analysis is performed.

The collected data can be divided into three groups: (1) process parameters, (2) alarm occurrences, and (3) products and tools.

Process-parameters data. Process-parameters data can be seen as a table that holds the numerical values of the process parameters, the hour, the date, etc., for each cycle (Table 2). These data are accessible from the machine controllers. Due to the finite memory capacities of the controllers, the raw collected data are saved in several hundreds of files. The structure of an individual file's content varies according to the selection and the order of the process parameters that were selected on the controller screen at the time of the data export. This inconsistency needs to be handled in the steps of pre-processing or circumvented by a smart planned system for data acquisition and storage. However, in these files a common set of 26 process parameters is present. The common process parameters are considered for further analysis. These process parameters are the cycle time (s), injection time (s), plastification time (s), plastification stroke (mm), pressure at the point of switching to pressure control (bar), etc.

Alarms data. Several different alarms can occur during each cycle. The data about the alarms is accessible through the machine controllers. Individual alarms are connected to the process-parameters data through the date, the time, and the serial number of the cycle for the current data export (Table 3). Generally, alarms are classified into two groups: (1) 1st degree alarms that result in a machine stop, and (2) 2nd degree alarms or warnings that just draw attention to overruns, threats, or anomalous process states.

Data about products and tools. These data are manual entries of the workers into the MES. The product and used-tools tables hold information about the machine ID, the tool ID, the product code, the date, the approximate time of mounting the tool, etc. (Table 4). These data can be connected to the other two types of data, knowing the approximate times of mounting the tools and the name of the machine.

By merging these three types of data and knowing the locations of their generation, each cycle c_i is described in terms of date and time, work system, machine, the common set of 26 process parameters, corresponding alarms, and the corresponding tool (Eq. (20)). c_i^{id} values were prescribed as defined in Eqs. (2) and (3).

$$\begin{aligned} c_i = \{ & c_i^{id}, c_i^{dateTime}, c_i^{workSystem}, c_i^{machineID}, \\ & c_i^{processParameter1}, c_i^{processParameter2}, c_i^{processParameter3}, \ldots, \\ & c_i^{processParameter26}, c_i^{machineAlarm1}, c_i^{machineAlarm2}, \\ & c_i^{machineAlarm3}, \ldots, c_i^{toolID} \} \end{aligned} \tag{20}$$

Table 4
Records of tool exchange.

Mach. ID	Tool ID	Product code	Date and time of mounting	Date and time of dismounting	...
...	...	...	...	...	...
KM80	685	2630614	11/09/2015 12:10	14/09/2015 20:50	...
KM80	0955	2650084	14/09/2015 20:50	22/09/2015 16:43	...
KM80	0933	2570904	22/09/2015 16:43	23/09/2015 19:29	...
...	...	...	...	...	...

Table 5
Categorisation of input data types.

Metadata (*G*1)	Process-specific (*G*2)	Fault-specific (*G*3)	Other (*G*4)
id	*processParameter*1	*machineAlarm*1	*dateTime*
dateTime	*processParameter*2	*machineAlarm*2	*machineID*
workSystem	*processParameter*3	*machineAlarm*3	
quality	*processParameter*4	*machineAlarm*4	
machineID	*processParameter*5	*machineAlarm*5	
toolID	*processParameter*6	...	
	...		

Table 6
Designations of process-parameter feature types.

Design.	Feature type	Design.	Feature type
m-1	$\langle c_n\rangle_{1-1} - \langle c_n\rangle_{15-30}$	std-15	$(\sigma_n)_{1-15} - (\sigma_n)_{15-30}$
m-2	$\langle c_n\rangle_{1-2} - \langle c_n\rangle_{15-30}$	sl-1	$[s_n]_{1-1}$
m-3	$\langle c_n\rangle_{1-3} - \langle c_n\rangle_{15-30}$	sl-2	$[s_n]_{1-2}$
m-5	$\langle c_n\rangle_{1-5} - \langle c_n\rangle_{15-30}$	sl-3	$[s_n]_{1-3}$
m-10	$\langle c_n\rangle_{1-10} - \langle c_n\rangle_{15-30}$	sl-5	$[s_n]_{1-5}$
std-5	$(\sigma_n)_{1-5} - (\sigma_n)_{25-30}$	sl-10	$[s_n]_{1-10}$
std-10	$(\sigma_n)_{1-10} - (\sigma_n)_{20-30}$		

UMS identification problems. In the period of data collection, 890 machine stops occurred that were unplanned or not a consequence of the start-up conditions, i.e., UMSs. At the times when UMSs emerged, 328 different combinations of 197 different types of alarms occurred. In addition to the alarms that may indicate a root cause, other types of alarms were occurring that are just a consequence of the machine stop. Due to the complex combinations of the alarms that occurred and the data size, it is not possible to simply identify the different types of faulty operating conditions that result in an unplanned process interruption.

3.3. Analysis

The IIFC analysis was performed using a software system built in the Python programming language. To implement the proposed steps of analysis, programming libraries SciPy and Scikit-learn [16,17] were used.

The initial step is to discern the normal and the faulty cycles. In the case study, faulty cycles are defined as cycles when UMSs occurred during a settled, stable operating-conditions regime. Normal cycles are those that are sufficiently far apart from the cycles in which the machine stops or controller warnings occurred, or a process start-up was performed. The c_i^{id} values of the faulty and normal cycles can be identified knowing the cycles of the alarm occurrences and the process start-ups. The cycle of an UMS (faulty cycle) is defined when the appropriate 1st degree alarm occurred after 50 cycles, which do not belong to the start-up area. Not all the 1st degree alarms are indicating an UMS, e.g., the 1st degree alarm *A_00063 The number of produced pieces reached the set value* is not an alarm that indicates an UMS. The normal cycle is defined as a cycle that is sufficiently distanced (100 cycles) from any alarm occurrence or start-up conditions. The cycles of the start-up events can be identified knowing the 1st degree alarm occurrences, as well as the dates and times of replacing the tools.

Determination of input data classification. The different types of UMSs were identified on the basis of combinations of alarm occurrences and characteristic anomalies of the process-parameter values in the nearby cycles before a UMS occurs. The goal is to discover whether there are any connections between the identified types of faulty process conditions, different machines, or daily and weekly production cycles.

According to the above-listed objectives, the input data types were categorised as shown in Table 5.

Transformation G2. The set of normal cycles was undersampled uniformly according to their sequence on the corresponding machine from the initial number of 1,896,154 to a final 18,961 cycles.

Local changes to the averages and standard deviations, and the average slopes of the rise or fall of the process-parameter values for the recent cycles just before the observed cycle are chosen as the features. Let the average value and standard deviation of the process parameter of the starting k cycles and the ending j cycles before n-th cycle be denoted as $\langle c_n\rangle_{j-k}$ and $(\sigma_n)_{j-k}$ respectively, and let the average slope of the starting k cycles and the ending j cycles before n-th cycle be denoted as $[s_n]_{j-k}$ and defined by Eq. (21).

$$[s_n]_{j-k} = \sum_{i=j}^{k} (\langle c_n\rangle_{i-i} - \langle c_n\rangle_{(i+1)-(i+1)})/(k-j+1) \tag{21}$$

The calculation of feature vector for n-th cycle is defined by Eq. (22), in which the features are listed only for one process parameter. For other process parameters, the features are calculated in the same way.

$$\begin{aligned}\boldsymbol{x}_n = [\ldots, &\langle c_n\rangle_{1-1} - \langle c_n\rangle_{15-30}, \langle c_n\rangle_{1-2} - \langle c_n\rangle_{15-30}, \langle c_n\rangle_{1-3}\\ &-\langle c_n\rangle_{15-30}, \langle c_n\rangle_{1-5} - \langle c_n\rangle_{15-30}, \langle c_n\rangle_{1-10}\\ &-\langle c_n\rangle_{15-30}, (\sigma_n)_{1-5} - (\sigma_n)_{25-30}, (\sigma_n)_{1-10}\\ &-(\sigma_n)_{20-30}, (\sigma_n)_{1-15} - (\sigma_n)_{15-30}, [s_n]_{1-1}, [s_n]_{1-2},\\ &[s_n]_{1-3}, [s_n]_{1-5}, [s_n]_{1-10}, \ldots]\end{aligned} \tag{22}$$

In this way 338 features from 26 process parameters are defined. In Table 6, the designations of these process-parameter feature types are listed. The designations are used in the following paragraphs to interpret the models.

The empirical covariance method [17] was used to detect outliers on the basis of the Mahalanobis distance. 17% of the most decentralised examples were eliminated from the set of normal cycles.

Before applying the decision-tree algorithm, the features were filtered by applying the tree-based, feature-selection algorithm [17]. The result of this step was a reduced set of the 132 features.

An optimised version of the CART algorithm [17,18] was used to induce the decision-tree model. The Gini impurity splitting metric was used. The performance of the decision-tree model was estimated with a stratified 10-fold cross-validation for different combinations of algorithm parameters that determine the size of a decision-tree: the maximum depth and the minimum number of samples in a leaf. This decision-tree was induced from a highly imbalanced dataset, thus the F-measure of the faulty class is a more appropriate performance measure than the Accuracy. The calculated F-measure values at different size parameter values are shown in Fig. 4. The values of the final size parameters were chosen at the point where the F-measure value was the highest. The highest F-measure was 0.317, with the corresponding Precision 0.595 and Recall 0.216. Based on the maximum value of F-measure, the maximum depth was set to 20 and the minimum number of samples in a leaf was set to 15. 103 rules were identified from the nodes of the decision-tree model. The identified rules are described with the 65 features from 16 process parameters. In Table 7, these process parameters and corresponding designations for the interpretation, are listed. Fig. 5 provides a visualisation of the most informative part of the decision tree.

Transformation G3. In this transformation each faulty-cycle example is described with a combination of the alarms that

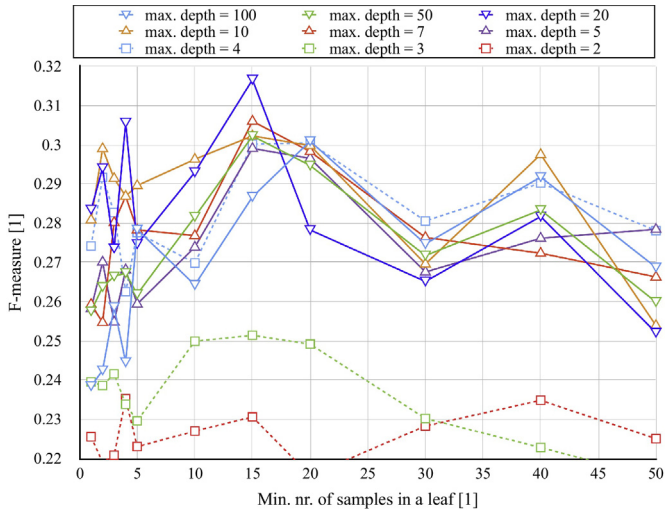

Fig. 4. Choosing the size parameters and evaluating the performance of the decision tree.

Table 7
Process parameters from the decision-tree nodes.

Process parameter	Unit	Designation
Cycle time	s	cycle_time
Injection time	s	injection_time
Plastification time	s	plastification_time
Remainder of the material in the cyl. after the Injection	mm	inj_material_remain
Pressure at the point of switching to pressure control	bar	switch_press
Maximum injection pressure in the cylinder	bar	max_inj_press
Maximum pressure of the cylinder pressure curve	bar	max_cyl_press
Time of opening the tool	s	tool_ope_time
Time of closing force increasing when closing the tool	s	clo_forc_incr_time
Maximum speed of injection	mm/s	max_inj_speed
Integral value of injection speed curve	(mm/s) s	intgr_inj_speed
Integral value of injection stroke	mm s	intgr_inj_stroke
Integral value of injection pressure in the cylinder	bar s	intgr_inj_press
Temperature of the cylinder heating zone 1	°C	cyl_heat_zn_1
Temperature of the cylinder heating zone 2	°C	cyl_heat_zn_2
Temperature of the cylinder heating zone 11	°C	cyl_heat_zn_11

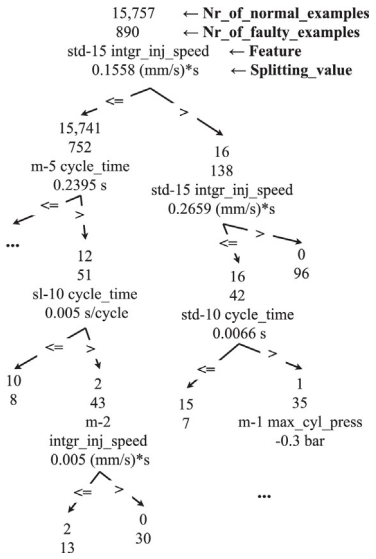

Fig. 5. The most informative part of the decision tree.

Table 8
Designations of alarm features.

Designation	Feature
I-z	1st degree alarm in the observed cycle
II-z	2nd degree alarm in the observed cycle
II-1	2nd degree alarm one cycle before the observed one
II-2	2nd degree alarm two cycles before the observed one
II-3	2nd degree alarm three cycles before the observed one
...	...
II-29	2nd degree alarm 29 cycles before the observed one
II-30	2nd degree alarm 30 cycles before the observed one

occurred in the observed cycle as well as in the previous 30 cycles. The designations of these features are indicated in Table 8.

Transformation G4. Other (*G4*) data were transformed to a set of bit vectors that for an individual faulty cycle hold the facts about the machine ($M_1/M_2/M_3/M_4/M_5$), the day of the week (*Sunday*/*Monday*/*Tuesday*/...*/Saturday*), and the part of the day (*night*/*morning*/*afternoon*/*evening*).

Identification of faulty operating-condition types. The agglomerative clustering algorithm [17] was used, where for each step the group of examples was divided into two clusters. The Ward linkage criterion was used. The rules were weighted in a manner, that the overall information gained from process parameters had an equal importance as the overall information gained from alarms. The statistical methods of the unpaired *t*-test and Levene's test were used to stop the growth of the clustering tree.

3.4. Results

The result of the analysis is the interpretative model in the form of a hierarchical categorisation of the underlying UMS types with corresponding characteristic combinations of the rules. The visualisation of this model is shown in Fig. 6. The hierarchic classification starts with node 0. Examples[9] of this node are divided into examples of node 1 and node 2. Then node 1 is separated into nodes 3 and 4, and node 2 into nodes 5 and 6, etc. The identified type properties are described with information about frequent rules about process-parameter anomalies, alarm occurrences, machines, days of the week, and parts of the day. These descriptions can be given in the form of a list for an individual node. On these lists, thresholds can be defined to filter the rules in such a way that only the most frequent are shown. As examples, lists of the properties for nodes 46 and 14 are given in Tables 9 and 10.

To demonstrate the interpretative power of the model, two of the identified types of UMSs (*Lack of input material UMS* and *Saturday-night UMS*) are interpreted in this section.

Lack of input material UMS. Node 46 (list of frequent rules in Table 9) holds 51 examples. All the examples have in common the occurrence of the 1st degree alarm: the lack of input material. This event is, as a frequent event, present nowhere else in the induced model.

The examples of this node have in common a relatively large number of anomalies that are based on process parameters. This node has frequent occurrences of (1) a local increase of the standard deviations of the cycle time before the UMS emerges, (2) increased local averages and standard deviations of the plastification time, (3) a decrease of the remainder of the material in the cylinder after the injection, (4) a decrease in the values of the pressure at the point of switching to pressure control, etc. The combination of the identified frequent anomalies and the machine-controller alert type makes a physical sense. Due to the lack of input material in the plastification cylinder, more time is needed to collect a sufficient amount of

[9] *Example* – the cycle when it came to an UMS.

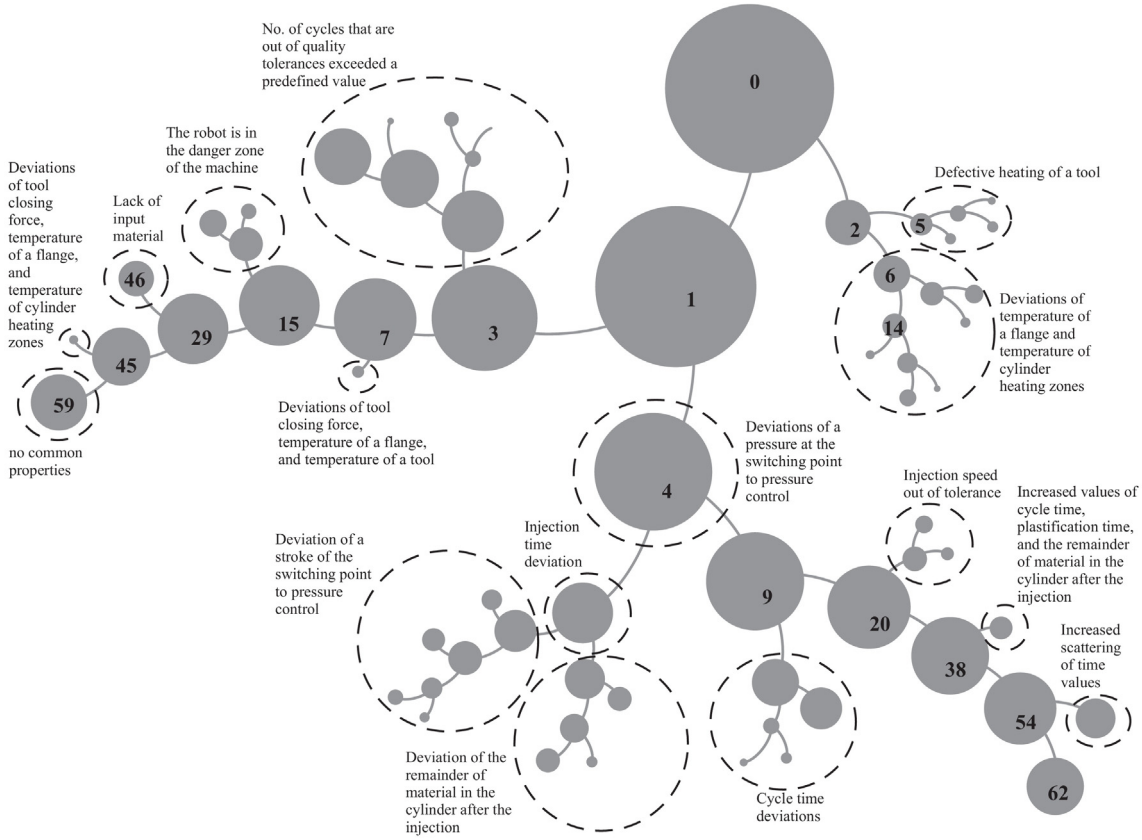

Fig. 6. Interpretative model – hierarchic categorisation of unplanned machine stops (UMSs). *The size of nodes is proportional to the number of examples in a node.*

material for the injection. Due to the smaller amount of material in the cylinder after the injection, less material is delivered into the cylinder. The decreasing pressures also make sense in a situation when the material is lacking, etc.

These types of UMSs only emerged on three of the five observed machines. This observation could already trigger a further investigation as to why these types of UMSs only emerged on these three machines.

If the root causes of this type of UMS cannot be preventively eliminated, a prediction system could be built to alert the operator before a UMS occurs. A relatively large number of frequent process-parameter anomalies suggests that building a prediction system is feasible, and gives directions about which parameters and types of anomalies could be used for prediction. If the UMS is predicted in time, it could be relatively simply avoided with brief increases in the input material flow rate when the predictive system detects anomalies of this type of UMS.

Saturday-night UMS. Node 14 (list of frequent rules in Table 10) holds 28 examples. The UMSs of this node occur on two machines (with one exception). Controller alarms connected to this type of UMS are related to the temperatures of the hydraulic oil, the flange, and the cylinder heating zones.

Due to frequent process-parameter anomalies it is concluded that the temperatures exceeded the upper tolerance limits. A higher temperature of the cylinder heating zones decreases the melt viscosity, the melt can flow into the mould more easily, and consequently the injection time is shorter. The increasing of the cycle time could be a consequence of the increasing of moulded part's cooling time because of the higher temperature of a part at the start of the cooling phase after its injection. Due to the low viscosity of the melt, the mould can be overfilled, and the remaining material in the cylinder after the injection is reduced.

The frequent rules about the time of occurrence reveal something very interesting. More than 92% of the UMSs occurred on a Saturday and 75% of the UMSs occurred at night.

General description of the final model. In the manner described above, the nodes of the induced interpretative model were examined. 16 groups were recognised that exhibit some common frequent rules. The groups are marked in Fig. 6 as the parts of the final hierarchical model. From the initial node 0, which contains all 890 UMS examples, the first clustering step separates the groups that represent UMSs related to the temperature deviations of the flange and the cylinder heating zones and defective heating of the tool. Other examples that are in node 1 are further separated into two larger branches. One of these branches (nodes 4–9–20–38–54–62) exhibits a common property, which is the deviation of pressure at the point of switching to pressure control. Out of the other larger branch (nodes 3–7–15–29–45–59), some of the more diverse types of UMSs are separated into further steps. Examples in node 59 had no common frequent rules that are more frequent than the given thresholds, which are listed at the bottom of the lists in Tables 9 and 10 (with an exception of rules about the machines).

3.5. Findings on the case study

A manageable number of UMS types was identified. The integration of the numerical values of the process parameters and the alarm occurrences provides a better understanding of the identified types of observed faulty conditions. The enrichment of the hierarchical categorisation of UMS types with other (*G*4) data allows for additional guidelines when searching for the root causes. The anomalies in the process-parameter values that separate the faulty from the normal operating conditions were identified. The

Table 9
List of rules – node 46.

Node 46		
Nr. of examples in the node = 51		
Relative freq. [%]	**RULE**	**Data type**
100.0	I-z A_00859_LACK_OF_INPUT_MATERIAL	G3
70.6	std-15 cycle_time >0.0008 [s]	G2
60.8	std-15 cycle_time >0.0029 [s]	G2
82.4	m-2 plastification_time >0.1375 [s]	G2
64.7	m-5 plastification_time >0.117 [s]	G2
80.4	m-10 plastification_time >0.0235 [s]	G2
64.7	std-5 plastification_time >0.0492 [s]	G2
80.4	std-5 plastification_time >0.0033 [s]	G2
60.8	std-15 plastification_time >0.0488 [s]	G2
62.8	m-3 inj_ material_remain <= -0.1357 [mm]	G2
84.3	sl-2 inj_material_remain <= -0.0125 [mm/cycle]	G2
66.7	std-5 inj_material_remain >0.0325 [mm/cycle]	G2
66.6	std-5 switch_press >0.8024 [bar]	G2
66.7	sl-2 max_inj_press <= -2.75 [bar/cycle]	G2
70.6	sl-10 max_cyl_press <-0.3 [bar/cycle]	G2
70.6	m-3 intgr_inj_speed >0.1167 [(mm/s)·s]	G2
66.7	std-15 intgr_inj_speed >0.0191 [(mm/s)·s]	G2
76.5	sl-2 intgr_inj_stroke <= -0.025 [(mm·s)/cycle]	G2
86.3	m-1 intgr_inj_press <= -1.25 [bar·s]	G2
86.3	sl-2 intgr_inj_press <= -0.25 [(bar·s)/cycle]	G2
68.6	sl-2 intgr_inj_press <= -4.75 [(bar·s)/cycle]	G2
70.6	sl-10 intgr_inj_press <= -0.95 [(bar·s)/cycle]	G2
66.7	std-10 intgr_inj_press >2.379 [(bar·s)/cycle]	G2
66.7	std-15 intgr_inj_press >1.8793 [(bar·s)/cycle]	G2
49.0	machine M_1	G4
0.0	machine M_2	G4
0.0	machine M_3	G4
23.5	machine M_4	G4
27.5	machine M_5	G4
Data types and the thresholds to show the rule on the list		
Process-specific (G2) data (threshold = 60%)		
Fault-specific (G3) data (threshold = 60%)		
Other (G4) data (thresholds: day of the week = 30%, part of the day = 50%, machine = 0%)		

Table 10
List of rules – node 14.

Node 14		
Nr. of examples in the node = 28		
Relative freq. [%]	**RULE**	**Data type**
96.4	I-z A_00642_MIN_MAX_HYDRAULIC_OIL_TEMP	G3
96.4	II-z A_00650_FLANG_TEMP_OUT_OF_TOL	G3
100.0	II-z A_00510_CYL_HEATING_ZONE_1_OUT_OF_TOL	G3
100.0	II-z A_00510_CYL_HEATING_ZONE_2_OUT_OF_TOL	G3
100.0	II-z,A_00510_CYL_HEATING_ZONE_3_OUT_OF_TOL	G3
100.0	II-z A_00510_CYL_HEATING_ZONE_4_OUT_OF_TOL	G3
100.0	II-z A_00510_CYL_HEATING_ZONE_11_OUT_OF_TOL	G3
75.0	II-z,A_00710_TOOL_HEATING_ZONE_1_OUT_OF_TOL	G3
75.0	II-z A_00710_TOOL_HEATING_ZONE_2_OUT_OF_TOL	G3
82.1	II-z A_00640_HYDRAULIC_OIL_TEMP_OUT_OF_TOL	G3
60.7	sl-5 cycle_time >0.001 [s/cycle]	G2
71.4	std-15 cycle_time >0.0008 [s]	G2
64.3	m-3 injection_time <= -0.0022 [s]	G2
64.3	std-5 plastification_time >0.0033 [s]	G2
60.7	sl-2 inj_material_remain <= -0.0125 [mm/cycle]	G2
60.7	std-10 max_inj_speed >0.0939 [mm/s]	G2
60.7	sl-2 intgr_inj_speed <= -0.025 [(mm·s)/cycle]	G2
75.0	night	G4
92.9	Saturday	G4
0.0	machine M_1	G4
64.3	machine M_2	G4
23.1	machine M_3	G4
0.0	machine M_4	G4
3.6	machine M_5	G4
Data types and the thresholds to show the rule on the list		
Process-specific (G2) data (threshold = 60%)		
Fault-specific (G3) data (threshold = 60%)		
Other (G4) data (thresholds: day of the week = 30%, part of the day = 50%, machine = 0%)		

resulting model can be interpreted relatively easily. It is not necessary that the end-user understands the process of interpretative model induction in detail in order to make use of the analysis.

4. Conclusions

Despite the high level of information and communication technologies development and advanced data-analysis methods, the data that are generated in manufacturing systems often remain unexploited. It is necessary to approach the new applications that use the value that is hidden in the available data.

This paper presents an innovative holistic approach to the analysis of the usually available data of a widespread type of a manufacturing process in a Big Data context. The goal of the paper is to propose a data-analysis method for interpretative identification of faulty operating conditions in a cyclic manufacturing process. The presented data-analysis method integrates well-known machine-learning techniques, i.e., decision trees and clustering, which are in the combination with conventional fault-diagnosis approaches used to extract the key information from the large amounts of complex manufacturing data. The result is a hierarchic categorisation of the underlying faulty-condition types with descriptions that can be used for decision support when determining actions to eliminate faults.

The applicability and usefulness of the presented method are shown by applying it to the real industrial data of a plastic injection-moulding process. The key results and conclusions are as follows.

(1) A manageable number of unplanned machine-stop types is identified, with corresponding characteristics that exhibit a physical sense.
(2) The data-integration approach enables a better understanding of the identified unplanned machine-stop types and a definition of additional guidelines to search for the root causes.
(3) It is not necessary that the end-user understands the process of model induction in detail in order to make use of the analysis.

In the future, quantities of data will become even larger and more complex. For the purpose of extracting value from the available data, data usage and management will need to be improved. This work represents an example of how advanced methods of analysis can be used in manufacturing practise to face the challenges and opportunities that arise due to the development of information and communication technologies.

Acknowledgment

This work was partially supported by the Ministry of Higher Education, Science and Technology of the Republic of Slovenia, grants no. 1000-15-0510 and C3330-16-529000, and by the Slovenian Research Agency, grant no. P2-0270.

References

[1] Esmaeilian B, Behdad S, Wang B. The evolution and future of manufacturing: a review. J Manuf Syst 2016;39:79–100.
[2] Wang L, Törngren M, Onori M. Current status and advancement of cyber-physical systems in manufacturing. J Manuf Syst 2015;37:517–27.

[3] Boyd D, Crawford K. Critical questions for big data: Provocations for cultural, technological, and scholarly phenomenon. Inf Commun Soc 2012;15(5):662–79.
[4] Villars RL, Olofson CW, Eastwood M. Big Data: what it is and why you should care. White Paper, IDC; 2011.
[5] Gartner. Gartner it glossary; 2016. URL: http://www.gartner.com/it-glossary/big-data [accessed 17.01.16].
[6] Hurwitz J, Nugent A, Halper F, Kaufman M. Big Data for dummies. John Wiley and Sons; 2013.
[7] Laney D. 3D data management: controlling data volume, velocity and variety. META Group Research Note; 2001.
[8] Hitzler P, Janowicz K. Linked data, Big Data, and the 4th paradigm. Semant Web 2013;4(3):233–5.
[9] Hu H, Wen Y, Chua TS, Li X. Toward scalable systems for Big Data analytics: a technology tutorial. IEEE Access 2014;2:652–87.
[10] Chen M, Mao S, Liu Y. Big Data: a survey. Mobile Netw Appl 2014;19.2:171–209.
[11] Precup RE, Angelov P, Costa BSJ, Sayed-Mouchaweh M. An overview on fault diagnosis and nature-inspired optimal control of industrial process applications. Comput Ind 2015;74:75–94.
[12] Babiceanu RF, Seker R. Big Data and virtualization for manufacturing cyber-physical systems: a survey of the current status and future outlook. Comput Ind 2016;81:128–37.
[13] Struyf J, Džeroski S, Blockeel H, Clare A. Hierarchical multi-classification with predictive clustering trees in functional genomics. Progress in Artificial Intelligence. EPIA 2005, 3808. Springer: Lecture Notes in Computer Science; 2005. p. 272–83.
[14] Blockeel H, De Raedt L, Ramon J. Top-down induction of clustering trees. In: Proceedings of the 15th International Conference on Machine Learning; 1998. p. 55–63.
[15] Rosato DV, Rosato MG. Injection molding handbook. Springer Science and Business Media; 2012.
[16] Jones E, Oliphant T, Peterson P, et al. SciPy: Open source scientific tools for Python; 2001. URL: http://www.scipy.org/ [Online; accessed 08.02.16].
[17] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: machine learning in Python. J Mach Learn Res 2011;12:2825–30.
[18] Breiman L, Friedman J, Stone CJ, Olshen RA. Classification and regression trees. The Wadsworth and Brooks-Cole statistics-probability series. Taylor & Francis; 1984.