Assessing product quality from the production process logs

Le Toan Duong, Louise Travé-Massuyès, Audine Subias, Nathalie Barbosa Roa

Noname manuscript No.
(will be inserted by the editor)

Assessing product quality from the production process logs

Le Toan Duong · Louise Travé-Massuyès · Audine Subias* · Nathalie Barbosa Roa

Received: date / Accepted: date

Abstract A real challenge for manufacturing industry is to be able to control not only the manufacturing process but also the production quality. Products that are suspected to be faulty are deviated from their nominal path in the production line and inspected more closely. The fact that some products deviate from the nominal path and others fail at some check operations can be used as an indicator of poor product quality. Based on this idea, this paper proposes a method to compute a product quality index or more exactly a penalty index taking into account both product path and production batches. The method relies on categorizing the products according to how they follow the production path and process mining techniques. The originality of the proposed index is to be built from advanced data analysis techniques enhanced by expert know-how. The quality index highlights risk of customer return, which is highly relevant information for the after sales service. The significance of the method is illustrated on a printed circuit board production line using surface mount technology at *Vitesco Technologies.* Data is collected from the real manufacturing execution system. The results obtained over more than 10000 single electronic boards show that 91.7% of the products are in good compliance with respect to the requirements. For the other products, the method identifies the root causes of poor quality that may call for maintenance or reconfiguration actions.

* Corresponding author

Le Toan Duong · Louise Travé-Massuyès · Audine Subias
LAAS-CNRS, Université de Toulouse, CNRS, INSA, Toulouse, France
E-mail: ltduong, subias, louise@laas.fr

Le Toan Duong · Louise Travé-Massuyès · Audine Subias · Nathalie Barbosa Roa
ANITI, Université Fédérale Toulouse Midi Pyrénées, France

Le Toan Duong · Nathalie Barbosa Roa
Vitesco Technologies France SAS, Toulouse, France

Keywords Industry 4.0 · Process mining · Event log · Predictive analysis · Product quality index

1 Introduction

Controlling and optimising the quality of produced parts is a major challenge in the industry today. There is a need for methods to verify that quality objectives are reached and to propose maintenance actions risen from this challenge.

Industry 4.0 is a digital and connected industry offering continuous information related to manufacturing processes and products. In this context, these methods must be able to deal with massive amounts of production data, notably during process execution. Production data are tracked and collected in real-time by Manufacturing Execution Systems (MES). These data allow virtual representations in the form of digital twins of the production process, from the manufacturing order to product delivery. Process mining is a family of process data analysis methods to discover, monitor and improve processes by extracting knowledge from event logs recorded by such systems [1]. The extracted knowledge is captured in a model that represents the process as it is and not as perceived by the expert. As a consequence, deviations between observed process behaviour (real behaviour) and expected behaviour (theoretical or planned) can be highlighted. Key indicators can be designed from these deviations to guide maintenance decision making.

At Vitesco Technologies, Printed Circuit Boards (PCB) are manufactured continuously 7 days a week, 24 hours a day under strict process checking. Products suspected to be faulty are deviated from their nominal path in the production line and inspected more closely. The fact that some products deviate from their nominal path and others fail at some check operations might indicate a risk of poor quality. This risk is related to the discrepancy between the expected, validated production flow and the actual observed production flow. Moreover, not only single product paths could be an indicator of product quality but the distribution of the production batch paths can also give a hint on this. Intuitively a "good" PCB produced within a batch of all "good" products, i.e. respecting the expected path and within the validated time constraints, is expected to have a better quality than a "good" PCB being the only "good" product of its production batch. The approach proposed in this paper is based on this expert intuition. It relies on categorizing products according to the path they follow along the process and on the results for their production batches. These information are formalized and ultimately used to compute a penalty index. This work can be related to virtual metrology that applies to the semiconductor manufacturing domain [2].

The contributions of this paper are twofold. First, an end-to-end log-based framework is developed to compute a penalty index as a relevant indicator of product quality. For this, different process mining approaches to discover and represent the underlying model and to check product path conformance are applied. Second, the framework is applied to a real PCB assembly process of Vitesco Technologies.

The originality of proposed index is that it is computed from the data collected on the PCB assembly process, taking into account the production

flow and time as a key factor. In this way it differs from quality indexes that are only based on measurements or checks directly on the products. Moreover, the proposed index not only allows to take into account product paths with related flow times but also production batches. Last but not least, it captures business knowledge and evaluates individual products as well as the overall process for every specific phase of the production.

The results obtained on an industrial use case demonstrate the significance of the proposed index and its value for diagnosing the health status of industrial processes and trigger the adequate actions.

The paper is organised as follows. Section 2 gives some background on process mining. Section 3 describes the PCB assembly process and data structure. Then, section 4 presents the mathematical concepts of process model and time constraints as well as their construction method. Section 5 explains how product populations are categorized and the concepts that are used to build a relevant product penalty index to assess product quality. Results on the industrial use case are given in section 6. Finally, section 7 provides some conclusions and perspectives.

2 Background on process mining

The field of process mining is concerned with extracting useful information about process execution, by analysing event logs [1]. The research area of process mining is large and leads to a wide range of applications. For example, process mining techniques have been applied to support the invoice handling process in a road construction and maintenance company [3], for organizational mining to discover social networks and to optimize the underlying processes [4], or in the healthcare environment to improve the quality of patient care [5]. This section presents an overview of this field and its applications, pointing at the own objectives of the work presented in this paper.

By extracting knowledge from event logs, process mining aims to discover, monitor and improve real processes. Process mining includes process discovery, conformance checking and process enhancement (i.e., model extension, repair, etc.). Process discovery and conformance checking are the most important tasks.

Process discovery consists of finding a process model by analysing a set of sequences or traces extracted from event logs. Many process discovery algorithms have been proposed in the literature. The α algorithm [4] is the first and simple discovery technique that constructs causal relationships observed between tasks. This algorithm was proven to be correct for a large class of processes [4]. However it has problems with noise and incompleteness [6]. Therefore, other advanced techniques were developed. Heuristic mining [7] uses causal nets [6] for process model representation. This algorithm takes frequencies of events and sequences into account when constructing a process model. The basic idea is that infrequent paths should not be incorporated into the model. Inductive miner [8] is an improvement of the α algorithm and of

the heuristic miner. The Inductive miner works by repeatedly finding a split in the event log into smaller event logs. The procedure is repeated until a base case (sub-log with only one activity) is reached. The first inductive miner was presented in [8], then [9] proposed an adaptation of this algorithm to deal with incomplete logs and [10] solved the problem of scalability with quality guarantees. Other algorithms like genetic process mining that use an iterative procedure were also proposed in [11]. The algorithm proposed in this paper relies on a so-called *Directly-Follows Graph*(DFG) for process representation and on a frequency-based filtering to tackle complexity problem.

Conformance checking refers to the analysis of the consistency between the behaviour of a process described in a process model and event logs that have been recorded during the execution of the process [12]. This work brings important notions [13], like:

- *fitness* measures the extend to which log traces can be associated with execution paths specified in the process model: a model fits a log if all traces in the log can be replayed by the model,
- *appropriateness* provides the degree of accuracy and clarity in which the process model describes the observed behaviour.

Conformance checking is useful in several tasks, among them performance analysis [14], high-level deviations [15] and alignment-based precision metrics [16], where *alignment* indicates the differences between the model paths and the log traces. The conformance checking approach proposed in this article is original in the sense that it uses a decision tree with various criteria to compare real behaviour with the process model taking into account time constraints.

Process mining was originally developed for Business Process Management (BPM), so there have been many applications in real business process. [17] processed event logs to configure process risk indicators (PRIs) that predict process delay. [18] presents an overview of existing discovery techniques for the construction of high-level Business Process Model and Notation (BPMN) models. There are also the applications of process mining in healthcare [19]. There have been less applications of process mining in the manufacturing industry. [20] proposed a system architecture to use both structured and unstructured data to discover process model and analyse the performance. [21] developed a data model which merges data from different information systems to create event logs for process mining in end-to-end order processing. Process mining techniques have also been applied in the analysis and prediction of manufacturing costs [22]. The heuristic miner algorithm was applied in [23] to obtain the process model, then used for maintenance inspection interval optimisation.

This paper proposes a process mining framework for the manufacturing sector. The objective of the framework is to characterise product quality based on event logs recorded from an end-to-end PCB assembly process. The approach follows a standard process mining framework depicted in Figure 1 [24]. This figure summarises the process mining framework in a manufacturing process environment. In the data preparation step (step 1), raw data is extracted in form of text messages from the MES databases and it is converted to event

logs. Then, the event logs are pre-processed by checking missing values, filtering out redundant data and duplicated values, etc. (step 2). For this use case, the manufacturing process mining and analysis step (step 3) focuses notably on process perspective that consists of discovering process model, executing conformance checking and performance analysis. Finally, results and evaluations are presented in step 4. The tasks involved in this process mining framework are presented in more detail all along the paper.

3 The PCB assembly process

The use case of this study concerns the PCB assembly process given in Figure 2. This process is divided into two main phases:

- Front End assembly (FE): Electronic components are placed and soldered onto the PCB.
- Back End assembly (BE): Connectors are added into the electronic boards and the whole is covered by the housing.

In the front end, the electronic boards are assembled using Surface Mount Technology (SMT). A schematic view of a SMT line is presented in Figure 2. In the first process, called Solder Paste Printing (SPP), solder paste is pressed through a stencil mask to create a film over the PCB that forms primary interconnection basis between the components and connection pads. Next, the solder deposit quality is verified by the Solder Paste Inspection (SPI) machine. In this process, several properties of the pad such as volume, area, height and position are measured and verified according to the standard

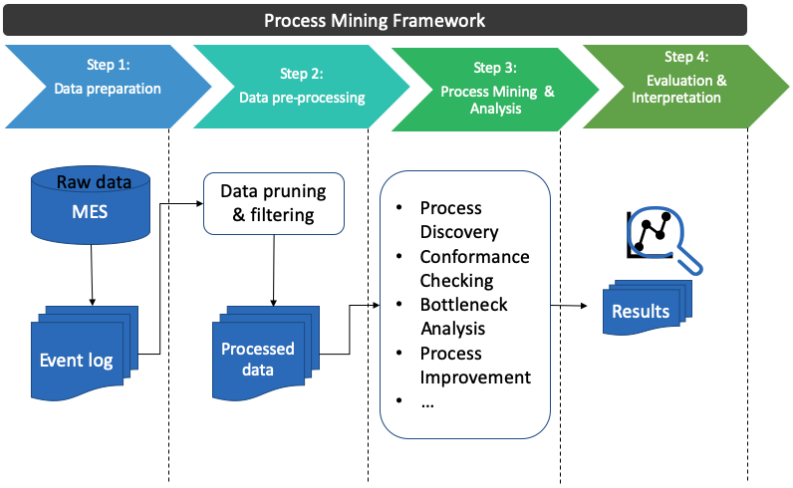

Fig. 1: Process mining framework

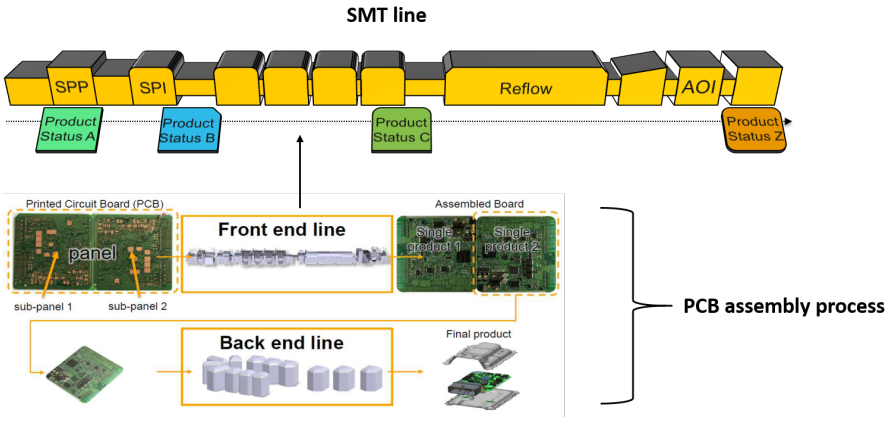

Fig. 2: A schematic view of PCB assembly process

limits. PCBs that are judged as good continue through the placement process. There, electronic components are mounted directly onto the surface of the PCB. At this point the joints are still in a liquid form so the components float in their position but are not permanently attached to the PCB. This happens in the next step, called reflow.

The reflow oven is divided into several sections allowing the PCBs to be conveyed in a specific temperature profile. Temperatures in each part are strictly controlled to guarantee the quality of the solder pads and the integrity of the placed components. The front end phase ends with an Automated Optical Inspection (AOI) that checks components final position and solders quality. The front end phase is then repeated on double-sided boards. In the rest of this paper,the front end process in the first side of PCB is denoted as $FE1$ and $FE2$ for the second side.

Once all components have been fixed to the PCB, an electrical test is performed. The In Circuit Test (ICT) checks the connections and verifies that the components correspond to their specifications. It is important to notice that each PCB can contain one or more sub-panels depending on the product type. In the front end phase the whole panel pass through processes and is modified or controlled at the same time.

The back end phase starts by cutting the PCB into single products. In this phase, the connectors that communicate with peripherals are fixed to the PCB and the set is assembled in a housing for protection and thermal dissipation. The final product is ready to be packed just after performing the functional tests that verify product operation. It is worth noting that, contrary to front end lines that have a standard configuration,back end lines are suited to the specific needs of each product, and in many cases, employ a combination of human operators and specialised robots [25].

Although several tests check the quality of solder pads and the function of electronic components during production process, i.e. SPI, AOI, ICT, etc.,

this is not enough to evaluate the global quality of final products. In fact, when assembling PCBs, processing times are one of the most important factors affecting product quality. Indeed, throughout the assembly process the PCBs are exposed to several factors affecting the reliability of electronics. Among the well known factors, there are temperature, relative humidity and dust. The impact of these factors on the reliability of electronics increases as the size of electronics reduces. Some moisture sensitive components can be damaged during reflow when moisture trapped inside the component expands. These sensitive components are always sealed in a air-tight packaging including the Moisture Sensitivity Level (MSL) information, i.e. the time period in which a moisture sensitive device can be exposed to ambient room conditions. Most of the damages caused by moisture, e.g. delamination, die damage, internal cracks, etc., are not visible on the component surface and therefore not detected by the visual inspection processes. In order to prevent moist related damage, the MSL should always be respected, and therefore the assembly processing times should be under control.

At *Vitesco Technologies*, production is organised in batches. The aim of the batch concept is that products within a batch are assembled under almost identical conditions and configuration. It is important to note that even if products in a batch are assembled under the same configuration, their production paths may be different because of operation failures. Batches can be retrieved automatically from event logs by identifying inactive periods. An inactive period is a time interval in which no product passes along the line.

All the products at *Vitesco Technologies* plants are tracked by a unique id in form of a data matrix marked on the PCB. This matrix is read each time the PCB goes through an operation. Information from all processes is collected in real-time by the Manufacturing Execution Systems (MES). The MES takes an important role in the smart factory transformation and the fourth industrial revolution. Indeed, a MES tracks, traces and controls the production, from the manufacturing order to product delivery.

4 Timed process model construction

The presented work takes place in a general process mining framework in which the event logs from the MES are leveraged to analyse the real process behaviour. For doing so, a process discovery step is performed leading to the construction of a *timed process model*. The aim of a timed process model is to represent the real process with time constraints that will be used for products quality characterisation. This type of model is then constructed from event logs in three steps. In the first step a model simply called *process model* is built without time consideration. In a second step, behavioural patterns are extracted from the initial model. Finally time constraints are added to obtain the *timed process model*.

4.1 Process model construction

The process model is an event based model. To go further in the description of the model construction, let us formally define the concept of *event* which relies on the concept of *event type* expressing the semantic label associated with an event. The set of all event types is denoted by $\mathcal{E}$.

Definition 1 (Event) An *event* is defined as a pair (e_i, t_i), where $e_i \in \mathcal{E}$ is an event type that identifies the event and $t_i \in T$ is the event date given by a timestamp.

Time representation relies on the time point algebra and time is considered as a linearly ordered discrete set of timestamped instants whose resolution is sufficient to capture the process dynamics. Events generally come in streams forming *sequences*.

Definition 2 (Sequence) An event sequence $\mathcal{S} \in (\mathcal{E} \times T)^*$ [1] of dimension h is a chronologically ordered set of events defined as $\mathcal{S} = \langle (e_i, t_i) \rangle_h = \{(e_i, t_i) / e_i \in \mathcal{E}, t_{i-1} < t_i, i = 1, \ldots, h \text{ and } t_0 = 0\}$.

Given a set of process activities, events are useful to represent the beginning and the end of each activity. Event types provide an identifier for the activity and the date places them in an ordered sequence.

A process instance is characterised by the sequence of activities that are executed, hence by an event sequence qualified as a *trace*. An *event log* gathers several traces.

Definition 3 (Trace) A *trace* is a specific event sequence $\sigma \in (\mathcal{E} \times T)^*$ corresponding to some process instance. The trace support $\mathcal{E}_\sigma$ is given by the sequence of event types present in the trace ordered according the trace.

Definition 4 (Event Log) An *event log* $\mathcal{L}$ is a multi-set over $(\mathcal{E} \times T)^*$, i.e. a trace can appear multiple times in an event log.

An event log contains data related to a process, i.e. the assembly process of electronic boards in the presented use case.

Each process gives rise to as many process instances as products. A process instance is characterised by a set of generated events that are gathered in a trace. The structure of event logs is presented in Figure 3.

Event logs are stored in a tabular format in which columns are different attributes and rows represent events. A fragment of event log used in this study is shown in Table 2 where an event is associated with a product (process instance), an operation, the machine being operated and a timestamp. The whole analysis is performed on tabular data. However, for the visualisation of the process model, a graphical representation that fits with the concept of directly follows graph in process mining is used.

[1] This notation is inspired by the Kleene star that, applied to a set of symbols, provides the set of all strings over these symbols, including the empty string [26].

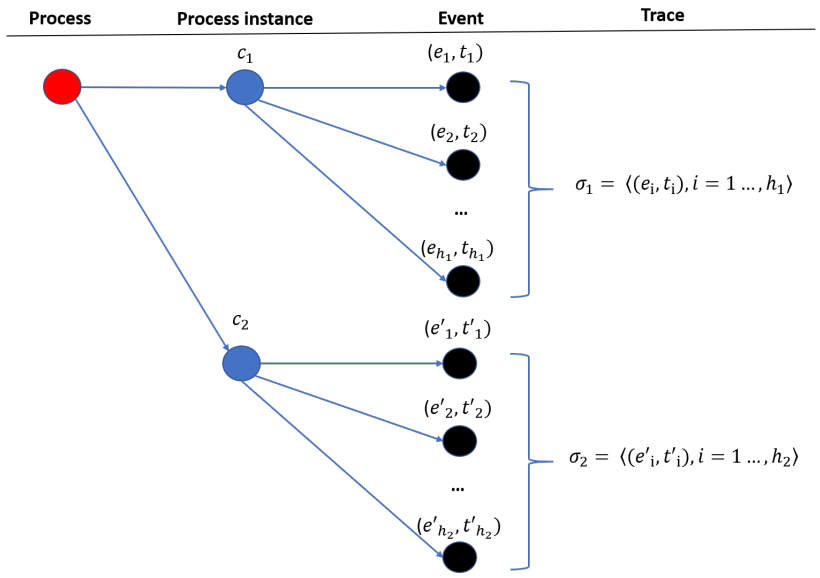

Fig. 3: Tree structure of an event log

Definition 5 (Directly Follows Graph) A *directly follows graph*, denoted as DFG is a pair (G, L) such that:

- G is a directed graph $G = (V, E)$ with vertex set V and edge set E.
- L is a set of labels, where l_{ij} is associated with the edge between vertex v_i and vertex v_j.

Definition 6 (Process model) A *process model*, denoted as PM is a DFG, where V is instantiated with the set of event types related to process operations and the edges of E represent the precedence relation according to the traces of the process. V includes two specific nodes, a source node v_{start} and a well node v_{end}. The edge labels in L can represent different information, such as transition counts, time durations, etc.

In this real use case, V is the set of event types occurring in the traces of production process, E is the set of all possible transitions representing product state changes, and L contains information about the frequency of these state changes stored in the event log. The DFG of production process obtained from all event logs data related to a product family collected in 2019 is shown in Figure 4. The graph was automatically constructed using the process mining library developed in python PM4Py [27]. In this graph, the event types associated to nodes are numeric identifiers of operations and are anonymized for confidentiality concerns. The v_{start} and v_{end} nodes ares in green and orange respectively. The edges refer to product state transitions between operations. Additionally, the label l_{ij} on top of each edge represents the number of products that have taken the corresponding transition. The

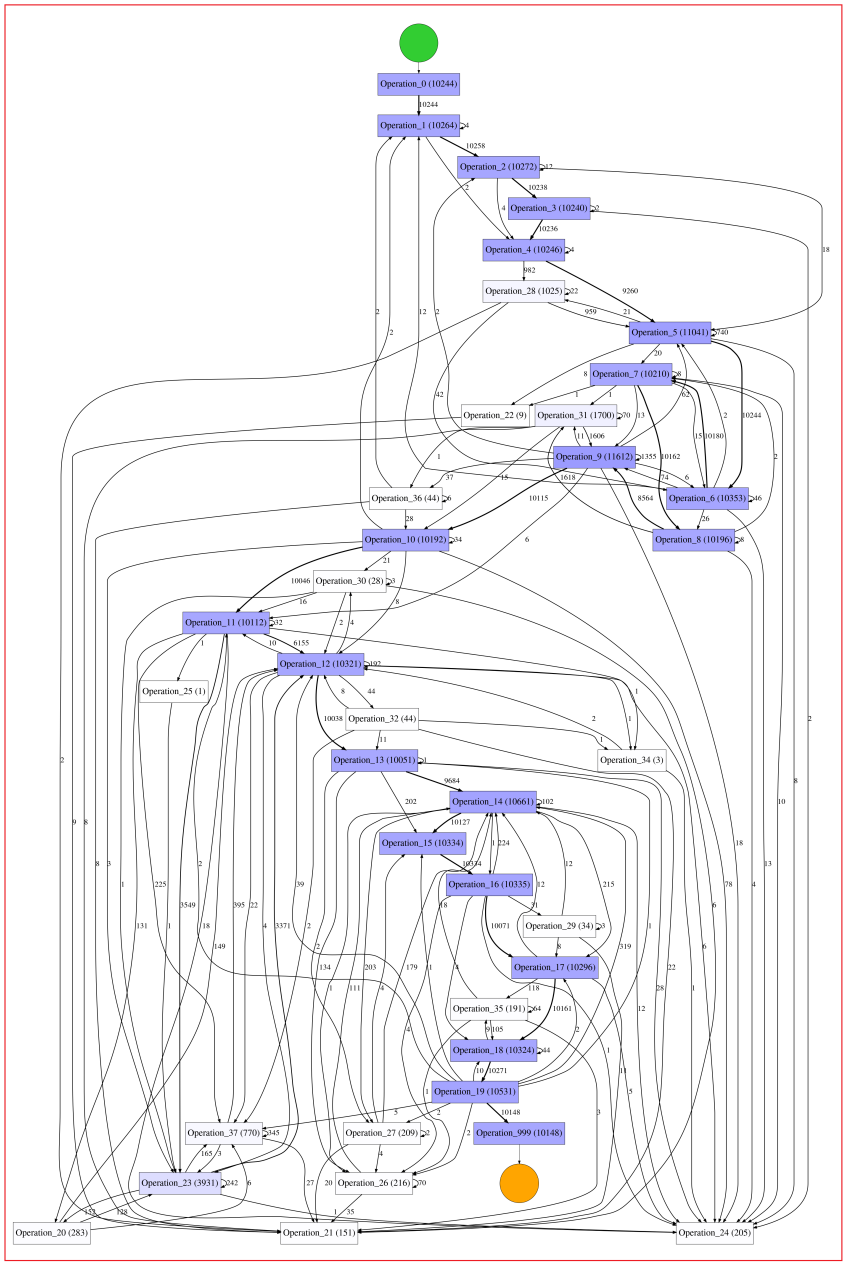

Fig. 4: Process model represented by a *DFG*

number of products that have gone through operations is reflected by the more or less dark purple color of the nodes. This number is reported between brackets in every node.

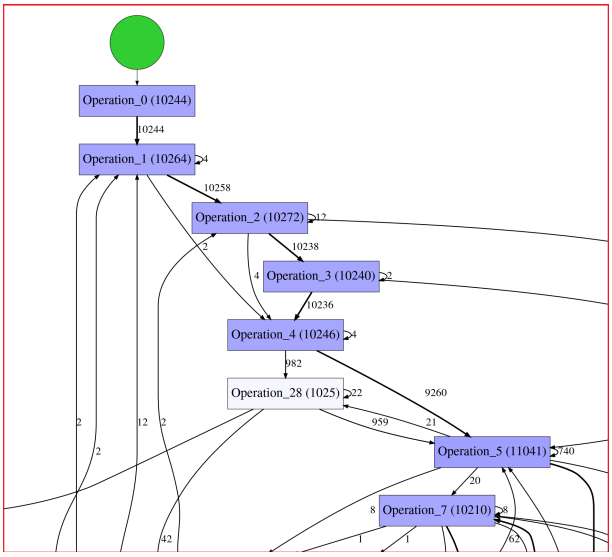

Fig. 5: Zoom of the DFG representing the process model on the *solder paste inspection side 1*

For example, there are 10258 product transitions between the node marked with the event type operation 1, i.e the *laser marking* operation and the node marked operation 2 that corresponds to the *solder paste inspection side 1*. There are 10264 products that have gone through operation 1 and 10272 that have gone through operation 2 (see Figure 5).

4.2 Pattern extraction from the process model

Since the construction exploits all the transitions that have been undertaken by the products in 2019, the resulting process model is unstructured and appears as a *spaghetti*. Therefore, this model needs to be pruned to reveal interesting underlying patterns and to facilitate the analysis. To do this, several process discovery techniques exist as mentioned in Section 2. In the first instance, a simple algorithm is developed to find the nominal process model or nominal pattern. Before describing this, notice that a trace as defined in definition 3 corresponds to a path between the two nodes v_{start} and v_{end} in the PM. The PM shown in Figure 4 includes such paths. The nominal process model is constructed based on the path that most products follow. This path is called the *nominal path* noted $\mathcal{P}^*$. Products of the same family follow the same path

if nothing wrong happens along the production chain. The nominal path $\mathcal{P}^*$ is then computed by the following formula:

$$\mathcal{P}^* = \underset{\mathcal{P} \in PM}{\operatorname{argmax}} \, freq(\mathcal{P}) \tag{1}$$

where $freq(\mathcal{P})$ returns the number of traces, i.e. process instances, that follow the path $\mathcal{P}$.

To apply formula (1), it is necessary to extract the list of all paths and their frequency from the event log. Then, the most frequent path in the process model is the nominal path which corresponds to the nominal process model. In Figure 4, the nominal path can be retrieved by following purple colour operation identifiers.

This work focuses on the normal behaviour of the process, nevertheless others criteria can be defined to extract particular paths from the PM and then to obtain other behaviours of interest i.e. others patterns.

The resulting model does not integrate any timed information. Nevertheless, as time constraints are quite indicative of process problems, the proposal is to integrate them in the model as presented in the next section.

4.3 Timed process model

By integrating time aspects in the process model, the objective is to retrieve the normal time constraints between process operations. For this, we introduce interval time labels that represent the time ranges of all products travelling between two operations for a set of process instances. The timed process model is thus the process model labeled with time intervals representing the time constraints for process instances.

Consider an event log $\mathcal{L}$ that gathers a set of traces $\mathcal{T}$ representing different instances of the same process.

Consider an edge between two adjacent vertices v_i and $v_j \in V$ corresponding to two event types e_i and e_j that belong to the support of all the traces in a subset $\mathcal{T}_k \subseteq \mathcal{T}$. The subset $\mathcal{T}_k$ gathers the traces that take a path through the consecutive operations represented by v_i and v_j. Let us also assume that in any trace $\sigma \in \mathcal{T}_k$, the timestamps of e_i and e_j are such that $t_j > t_i$, then the label l_{ij} is determined as follows:

$$l_{ij} = [\Delta t_{ij}^-, \Delta t_{ij}^+] \tag{2}$$

where:

$$\Delta t_{ij}^- = min_{e_i, e_j \in \mathcal{E}_\sigma, \sigma \in \mathcal{T}_k}(t_j - t_i),$$
$$\Delta t_{ij}^+ = max_{e_i, e_j \in \mathcal{E}_\sigma, \sigma \in \mathcal{T}_k}(t_j - t_i).$$

The time intervals bounds defined by equation (2) represent the inf and sup of the elapsed time between two consecutive product states in the process.

Definition 7 (Timed process model)
A *timed process model* (*t-PM*) is a process model PM for which edges are labeled according to the time constraints given by equation (2).

A process instance, or a trace $\sigma^* = \langle (e_1, t_1^*), (e_2, t_2^*), ..., (e_n, t_n^*) \rangle$ satisfies the timed process model if:

1. The sequence of event types $\langle e_1, e_2, ..., e_n \rangle$ can be replayed in the graph G of *t-PM*.
2. All event pairs (e_i, t_i^*) and (e_j, t_j^*) satisfy the time constraint $[\Delta t_{ij}^-, \Delta t_{ij}^+]$, i.e. $t_j^* - t_i^* \in [\Delta t_{ij}^-, \Delta t_{ij}^+]$.

5 Characterising production quality

The timed process model obtained in 4.3 provides a characterisation of the observed process that can be used as support to better evaluate the quality of final products. To achieve this, products are sorted into populations (i.e the set of traces related to the product) and each population is associated with a penalty index depending on the conformance of the paths with the obtained model. In fact, a classification algorithm is developed to categorize products based on their production path (i.e. process instance).

5.1 Sorting populations by conformance checking

Populations are defined by constructing a decision tree [28]. The primary advantage of using a decision tree is that it is easy to follow and understand. Decision trees have four main parts: a root node, internal nodes, leaf nodes and branches. The root node is the starting point of the tree, and both root and internal nodes contain a test based on an attribute. Each branch represents the answer to the test, and each leaf node represents a class label.

Let $\mathbf{Y} = \{y_\alpha, \alpha = 1, ..., m\}$ be a set of m class labels. The aim of the partition issued from the decision tree is to gather the products that share the same path. Classes $(y_\alpha)_{\alpha=1...m}$ are then associated with individual penalty indexes $(\tilde{p}_\alpha)_{\alpha=1...m}$ provided by process experts to score process instances.

Product clustering consists of checking the conformance of the corresponding trace with the timed process model. Several criteria are used, in particular order of operations, presence or absence of some important operations, consistency with the time constraints. These criteria or clustering rules are illustrated by the decision tree given in Figure 6. The building process of this decision tree involves several steps. First of all, products that have a path from the first operation to the end are selected. This means that those whose production path is incomplete are excluded. Then, among products that performed the first and last operation, the algorithm checks whether there are products that missed a *nominal operation*. The *nominal operation* is the operation in the nominal path. The next step is to check for order of nominal operations

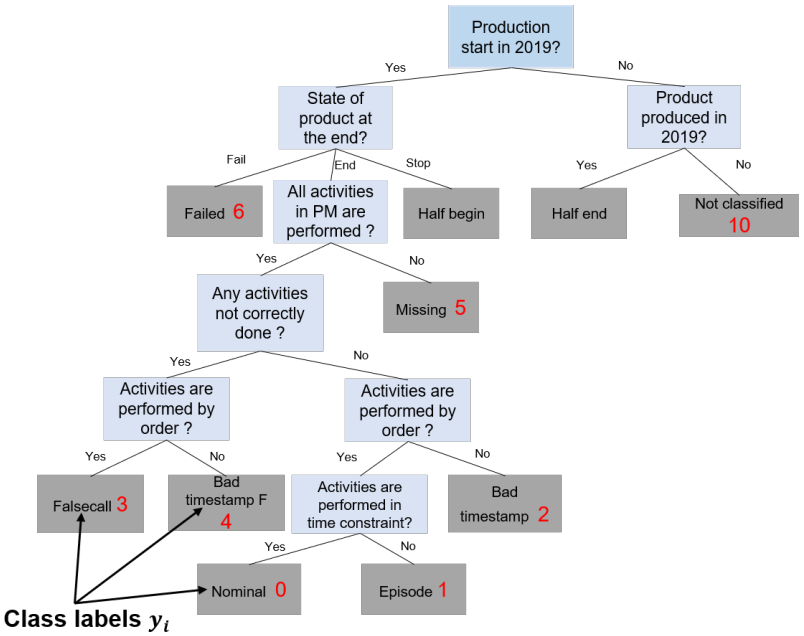

Fig. 6: Decision tree for product clustering

and the time constraints for transitions between these operations. The light blue boxes in Figure 6 represent internal nodes that contain split rules. The gray boxes are leaf nodes with corresponding class labels. Each class is associated with a penalty index, a real number in $[0, 10]$, depicted in red. Moreover, a description of all class labels is presented in Table 1.

5.2 Penalty index for products

As mentioned previously, each product is associated with a so-called *individual penalty index*, depending on the class it belongs to. This index aims at characterising product quality. As introduced in section 1, the quality of a product is evaluated based on its production path and on its production *batches*. Remind that a batch is composed by a set of products produced under the same configuration, within a given time frame.

Generally, products go through several phases from separate components to assembly and packaging. Due to the heterogeneity of configuration in the different phases, the partition of products into batches in each phase is different. This means that it is possible to have two products in the same batch in one phase and in different batches in the other phase. A given product hence belongs to several batches, one batch per phase. For a given phase, it is assigned a unique class indicated by its path along the operations of the phase. $y_\alpha, \alpha = 1, \ldots, m$.

Table 1: Description of class labels in the decision tree of Fig. 6

Class label	Description
Half begin	Products that lack information about the final state.
Half end	Products that lack start date information.
Not classified	Products that do not have start date and end state information.
Failed	Products that have failed an operation and are then discarded.
Missing	Products whose production has been completed but some nominal operations have been missed.
Falsecall	Products that have suffered some failures during production and have passed all nominal operations in order.
Bad timestamp F	Products that have suffered some failures during production and have not fulfilled the order of nominal operations.
Bad timestamp	Products that have completed production without failure but have not fulfilled the order of nominal operations.
Episode	Products that have completed production without failure and have passed all nominal operations in order but the time constraints have not been respected.
Nominal	Products that have completed production without failure and have passed all nominal operations in order and within the time constraints.

Let $\mathbf{X} = \{x_i, i = 1, ..., n\}$ be the set of all products. Let $\Phi = \{\phi_k, k = 1, \ldots, |\Phi|\}$ be the set of phases and $\mathbf{Y}_k = \{y_{\alpha|k}, \alpha = 1, ..., m_k\}$ the set of class labels for phase ϕ_k. Assume a set of batches $\mathbf{B} = \{b_j, j = 1, ..., |B|\}$, partitioned in $\mathbf{B} = \bigcup_{k=1}^{|\Phi|} \mathbf{B}_k$ according to the phases, where $\mathbf{B}_k = \{b_{j|k}, j = 1, ..., |B_k|\}$. The batch of product x_i in phase ϕ_k is denoted by $b_{j_i|k}$.

- **Individual penalty index of product x_i in a given phase ϕ_k**
  Let $\tilde{p}_k(x_i)$ denote the *individual penalty index* of product x_i in phase ϕ_k. This *penalty index* inherits the penalty index of the class it belongs to for the given phase. For example, a product x_i that belongs to class $y_{\alpha|k}$ for phase ϕ_k has individual penalty index $\tilde{p}_k(x_i) = \tilde{p}_{\alpha|k}$. As an example, the individual penalty index of products of the *nominal* class (see Figure 6) is equal to 0. Products in this class follow the nominal path without fail operations and respect the time constraints between operations.
- **Penalty index of a batch b_j of phase ϕ_k**
  The *penalty index* of a batch is computed based on the percentage of products in different classes for that batch. Given a batch b_j containing $|b_j|$ products, the ratio of products in the class $y_{\alpha|k}$ with the corresponding penalty index $\tilde{p}_{\alpha|k}$ is denoted by $r^j_{\alpha|k}$. The penalty index of a batch b_j of

phase ϕ_k is then given by:

$$p_k(b_j) = \sum_{\alpha=1}^{m_k} r^j_{\alpha|k} \times \tilde{p}_{\alpha|k} \tag{3}$$

As already explained, the quality of a product is influenced by the classes, i.e. the paths it follows along the production line for the different phases, and the batches to which it belongs. The different phases and the corresponding batches may impact product quality differently.

– **Global penalty index of product** x_i The final penalty index of a product x_i is hence obtained by combining its individual penalty indexes for each phase and the penalty indexes of the batches it belongs to weighted by their phases.

$$\begin{aligned} p_g(x_i) = \sum_{k=1}^{|\Phi|} \gamma_k \times \left(\lambda \times \tilde{p}_k(x_i) + (1-\lambda) \times p(b_{j_i|k})\right), \\ \lambda, \gamma_k \in [0,1], \quad \sum_{k=1}^{|\Phi|} \gamma_k = 1 \end{aligned} \tag{4}$$

Note that the value of individual penalty indexes $\tilde{p}_k(x_i)$ is in $[0, 10]$, the penalty index of batches and of final products given by equations (3),(4) are also in $[0, 10]$.

6 Quality evaluation of the PCB production Line

The proposed approach has been illustrated using a data set from a *Vitesco Technologies* plant.

6.1 Data description

Raw data considered in this use case are generated in real-time or near-real-time by the MES. In fact, machines generate data in form of messages that contain information about production process. An example of decoded messages from cloud storage service of *Vitesco Technologies* is presented in Figure 7. The data was anonymised for preserving confidentiality. Messages are generated from every single operation performing a modification or control over the product. Hence, messages contain features related to machine, operation and product. The main features of the messages are given below:

– **Type of message (red):** there are several types of messages, among which transitive messages and control messages are mainly used. While transitive messages notify that the PCB has entered or exited some operations, control messages give us information on whether or not the process pass as

expected. Hence, these messages have a feature of sanction that could be Pass (P) or Fail (F). An example of control messages is the one generated from the *AOI* machine.
- **Machine host name/Machine_ID (pink):** the identification of the machine or computer that performed an operation in the *PCB*.
- **Description of operation (purple):** description of the performed operation.
- **Family Code (green):** product family that is being produced.
- **Serial Number/Board_ID (orange):** the identification number of the product.
- **Operation Code/Operation_ID (blue):** the identification number of the operation being performed.
- **Sanction (brown):** for control message, the sanction given by the operation (F/P), the *F* means the operation has failed meanwhile the *P* letter means the operation has been performed successfully.
- **Timestamp (magenta):** the date and time an operation was performed.

```
P2|mid00002|Ope7_Line2|FamilyC128|Board_560|Operation_7|FACE2||006380|U55H4583|Line2-Config|2|1548866902|
PS|mid00003|Ope4_Line3|FamilyC128|Board_544|1548866935|
SF|mid00000|Ope2_Line0|FamilyC128|Board_566|Operation_2|002|P|1548858285|||||||||||||||||||||||||||||||
SS|mid00000|Ope15_Line0|FamilyC128|Board_730|Operation_15|P|1546860302|
```

Fig. 7: Snapshot of messages file from *MES*. Data are anonymised for preserving confidentiality.

To analyse the path of products and their transition time through event logs, the following features are selected: *Type of message*, *Serial Number*, *Operation Code*, *Sanction* and *Timestamp*. The experimentation was carried out on data related to a specific product family collected during 2019.

For the data preparation, the whole dataset was pre-processed and transformed to event log. The pre-processing consists of removing duplicated messages, filtering bad format messages and gathering messages related to the same product together. A fragment of the event log generated from cleaned messages is shown in Table 2. The data was anonymised for confidentiality issue.

The final data set corresponds to over 10000 single electronic boards, and over $98,5\%$ of them are successfully produced and delivered to clients. These products are denoted as *good product* and conversely products that were failed and rejected during production were denoted as *bad product*. The number of products by category in our dataset is presented in Table 3.

During production process, events tagged with a *fail* notification indicate that the corresponding operation has not been performed as supposed. Consequently, the product is taken to a diagnosis station and if the defects found to be non reproducible (*NRD* Non Reproductible Defect) it would return to the failed operation and be tested again. The distribution of number of fail

Table 2: A fragment of an event log generated from messages: an event per line.

Board_ID	*Operation_ID*	*Timestamp*	*Machine_ID*
Board_0	Operation_19	2019-01-21 11:20:49	Machine_0
Board_0	Operation_12	2019-01-21 11:20:50	Machine_26
Board_0	Operation_13	2019-02-07 23:52:37	Machine_19
Board_0	Operation_14	2019-02-08 00:24:21	Machine_11
Board_0	Operation_15	2019-02-08 00:31:33	Machine_12
Board_0	Operation_16	2019-02-08 00:31:35	Machine_12
Board_0	Operation_17	2019-02-08 00:39:39	Machine_20
Board_0	Operation_18	2019-02-08 00:42:49	Machine_14
Board_0	Operation_19	2019-02-08 00:43:00	Machine_14
Board_0	Operation_999	2019-02-08 00:43:01	Machine_0
Board_1	Operation_19	2019-01-21 11:20:33	Machine_0
Board_1	Operation_12	2019-01-21 11:20:34	Machine_26
Board_1	Operation_37	2019-01-30 23:55:46	Machine_10
Board_1	Operation_37	2019-01-30 23:57:54	Machine_10
Board_1	Operation_21	2019-01-30 23:57:55	Machine_0
...	...	...	...

Table 3: Summary of products produced in 2019

Product category	**Quantity**	**Percentage (%)**
Good product	9902	98,54
Bad product	146	1,56
Total	10048	100

events per product appearing during the production process for *good* and *bad* products is presented in Figure 8. There are more fail events generated in bad products than in good products, which as shown later, has been integrated in product quality characterising approach.

An analyse on the production duration for each product was also performed. The production duration corresponds to the time elapsed between the date when the product is put on the production line and the date when it finishes the last operation. The number of products per duration is presented in Figure 9. Most products, actually 79.75%, are manufactured within 2 days. Nevertheless, some of them stay in the production process much longer, even up to 6 months. For these products the industrial partner planned a more thorough analysis requiring the insights of process engineers.

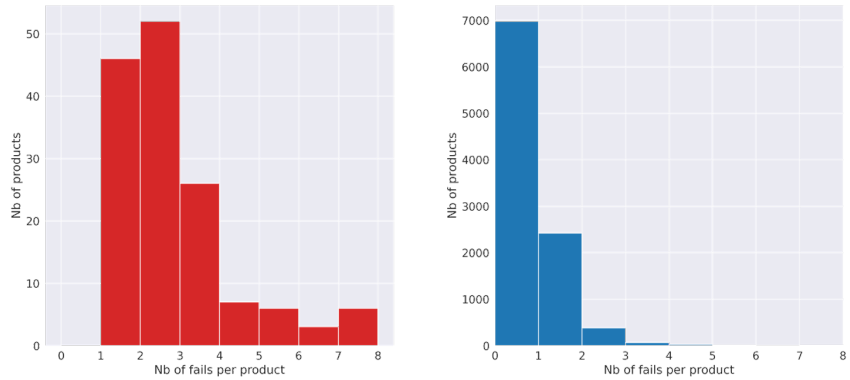

Fig. 8: Distribution of fail events per product. Left: Bad products, Right: Good products.

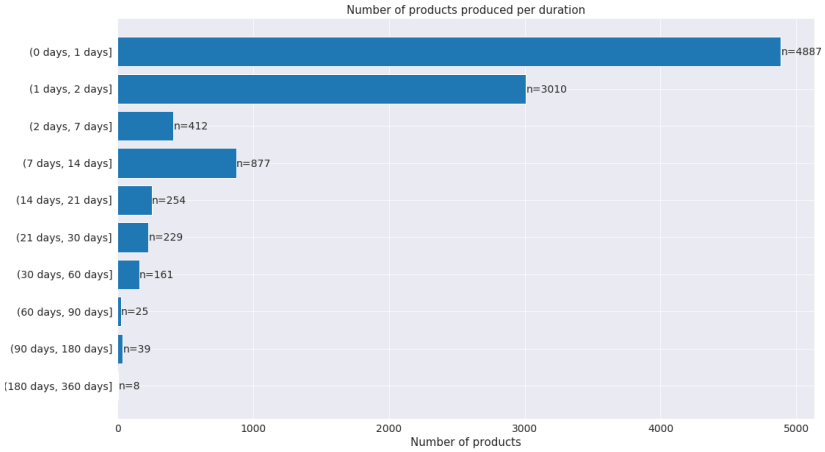

Fig. 9: Histogram of number of products produced in different time intervals.

6.2 Quality evaluation of electronic boards

The quality evaluation of electronics boards relies on the penalty index evaluated for each product i.e each PCB.

In the first step, instead of using the min and max value to define two bounds of the time constraints (see Section 4.3), statistical values, the 5^{th} and 95^{th} percentile are used to exclude outliers and extreme values. The timed process model obtained for this use case is shown in Figure 10. The process model

for this product family is quite simple. Indeed, operations are performed successively, one after the other. There is no deviation and there are no operations that perform at the same time (problem of concurrency).

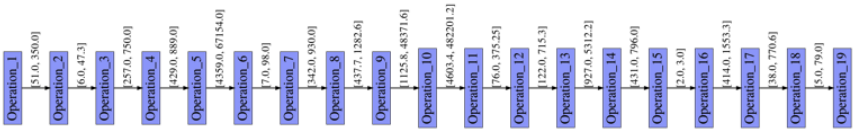

Fig. 10: The time process model

The next step is to categorize products by comparing their production path with the timed process model. A decision tree was used for this task (see Figure 6). The decision tree consists of split rules that branch on different classes. Each class is associated with a penalty index by process expert (presented as red number) between 0 an 10 which characterises the conformance level. Note that two classes *half begin* and *half end* are excluded because the paths of products in these classes are incomplete. In fact, as only data registered in 2019 were considered, products that were put in the production line before 2019 and appeared again in 2019 were classified in the *half end* class. Similarly, boards that have not finished production in 2019 were classified in the *half begin* class.

Once products are categorized, the penalty index for batches and global penalty index of products were computed (see equations (3),(4)). The results are presented below.

– **Penalty index obtained by batch:** as a reminder, a batch is a sub-set of products that are produced continuously and consecutively. Batches are extracted from event logs and the penalty index of a batch is computed by the equation (3). The relation between penalty index and batch size in the three phases $FE1$, $FE2$ and BE of production process is shown in Figure 11. Some statistics such as the standard deviation of batch size, the percentage of batches which have penalty index < 1 and the percentage of perfect batches with penalty index equal to 0 are also computed (see Table 4). More than half of the batches have a small penalty index, between 0 and 1 over the three phases (see 3^{rd} column in Table 4). Especially in the FE1 phase, 81.24% of batches respect the timed process model. However, batches produced in the FE2 and BE phases are less consistent than those in the FE1 phase. The results also show that there are no perfect batches i.e batches with penalty index equal to 0. This means that in the dataset no batch had all its products passing through the production line without any failure or deviation and within the accepted time intervals. Additionally, it can be seen over the three phases that the number of products per batch varies with a standard deviation around 200 and batches with a high penalty index (orange points) have small size. Without an in-depth inves-

tigation, this correlation is explainable. Indeed, the fact that these batches are small is probably due to process interruption or changes in configuration after a sequence of products with bad behaviour. This explains that their penalty index for them is pretty high.

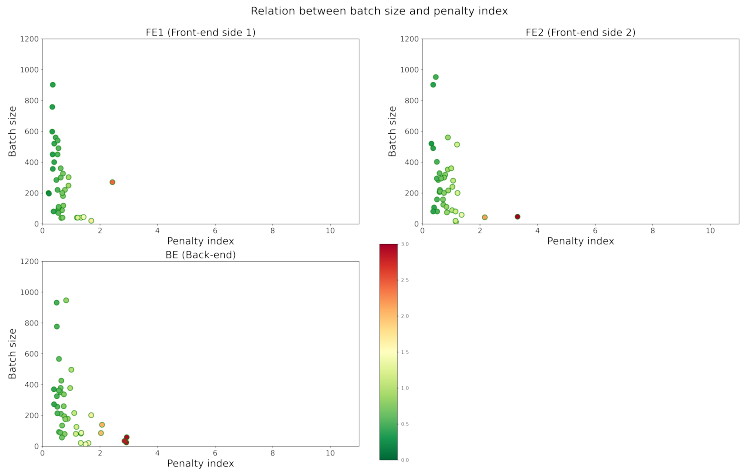

Fig. 11: Penalty indexes and batch sizes for each production phase

Table 4: Summary of batch quality

Phase	Batch size		Penalty index of batches	
	Mean	**Standard deviation**	**% batch with penalty index < 1** (Good batches, only delays)	**% batch with penalty index $= 0$** (Compliant batches)
FE1	269.42	217.25	84.21	0
FE2	261.03	213.74	69.23	0
BE	241.55	229.25	50	0

Table 5: Summary of product quality

Phase	**Mean**	**Median**	**Standard deviation**	**Range** (max − min)	**% products with penalty index < 1**
FE1	0.54	0.4	0.41	4.84	89.0
FE2	0.71	0.57	0.51	8.51	78.63
BE	0.8	0.75	0.38	3.52	76.38

– **Penalty index obtained by product in each phase:** The distributions of penalty indexes for products in the three phases are shown in Figure 12. Statistics calculated on these distributions which allow for comparison of production between phases are presented in Table 5. As expected, most

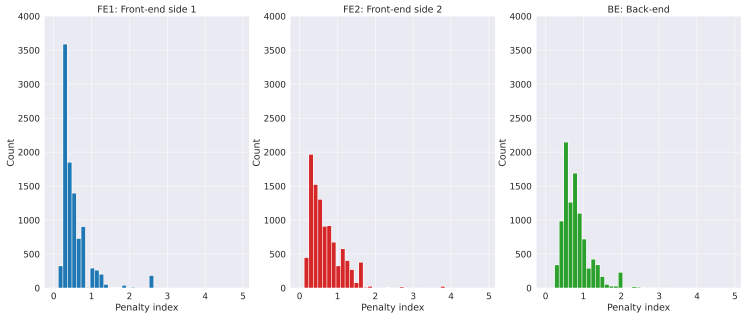

Fig. 12: Product quality for each production phase

products have penalty indexes between 0 and 1 (more than 75%) which indicates a good compliance with process model. Table 5 shows that among the three phases, $FE1$ is the most consistent with a mean and median value equal to 0.54 and 0.4. In particular, penalty indexes of products in $FE1$ phase have less variation than in $FE2$ (see their range and the standard deviation). Note that $FE1$ and $FE2$ are the same operations performed on the first and the second side of the PCB. These results are interesting and open the door for further examination and improvement actions.

– **Global penalty index for products:** the global penalty index is obtained once the product has performed all phases. The distribution of this index for all products is presented in Figure 13. As expected from previous results most of products (91.77%) have small penalty index, between 0 and 1. In association with process experts, detailed analysis should be performed on products with high penalty index.

7 Conclusion and Future work

This paper presents an event log based framework for learning the graphical model of a production process and assessing the quality of the manufactured products. Process mining techniques are used to build a timed process model and to check the conformance of product's paths. A penalty index balancing the path followed by the product along the line, in particular the fail operations that it has gone through, and the batches it belongs to in different process phases is presented and justified.

The proposed framework provides valuable information that can be leveraged in various problems related to process monitoring and optimisation. For instance in the process model, the information included about the number of products going through a given fail operation indicates that an operation/machine is causing problems and requires maintenance actions soon. It can hence be viewed as a health indicator. Self-loops around the nodes of the

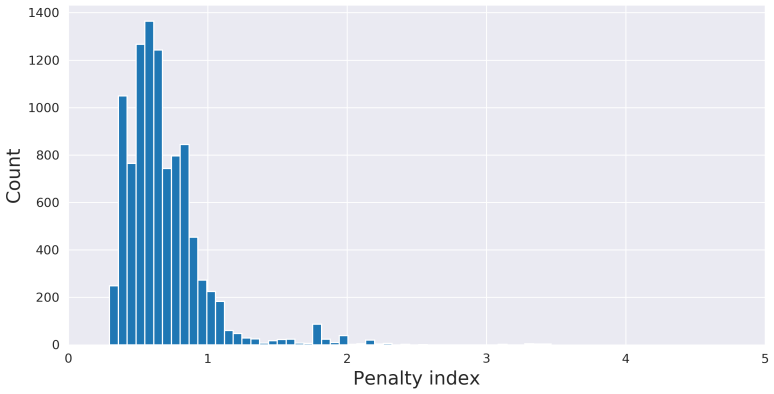

Fig. 13: Global penalty indexes for all products.

graph indicate repetitions of tests and operations which are also interesting to report. On the other hand, the process model may exhibit the number of products that do not finish the whole process and do not return to production. Time labels are also informative and interesting to monitor. In particular, an increasing upper bound is the symptom of a slowdown in between two operations, which may turn into upstream congestion.

The framework and algorithms are quite generic and can be applied as is to other manufacturing processes provided that circulating products are countable. The quality index itself accounts for several phases and it could be accommodated to more complex processes including discrete and continuous process phases. For the discrete phases, the framework of this paper could be used to obtain the individual quality index whereas a quality index would remain to be defined for continuous phases.

An experiment was conducted on a large dataset from a production line of the automotive manufacturing company Vitesco Technologies. The experiment demonstrates that the proposed framework is relevant. The results show to be effective for domain experts to have a deeper knowledge of the real production process. The proposed process model evaluates the behaviour of the products along the production time. It provides a global assessment of the product circulation and reveals abnormal ones. It is then leveraged to provide insight into the quality of the different products, which is then combined with the batches environment impact.

Future work will approach the problem of learning the parameters of the global penalty index based on after sales service data about products returned by customers. Another line of work is to consider an additional dimension that impacts product quality, which is the time slot during which the production takes place and in particular the time shifts.

8 Declarations

Acknowledgements This project is supported by ANITI through the French "Investing for the Future – PIA3" program under the Grant agreement №ANR-19-PI3A-0004.

Funding Not applicable

Conflicts of interest/Competing interests The authors declare there is no conflict of interest

Availability of data and material Not applicable for confidential issue

Code availability Not applicable for confidential issue

Compliance with Ethical Standards

Ethics approval (include appropriate approvals or waivers) Not applicable

Consent to participate (include appropriate statements) Not applicable

Consent for publication (include appropriate statements) Not applicable

References

1. Wil van der Aalst. Process mining: Overview and opportunities. *ACM Trans. Manage. Inf. Syst.*, 3(2), July 2012.
2. Pilsung Kang, Hyoung-joo Lee, Sungzoon Cho, Dongil Kim, Jinwoo Park, Chan-Kyoo Park, and Seungyong Doh. A virtual metrology system for semiconductor manufacturing. *Expert Systems with Applications*, 36(10):12554–12561, 2009.
3. Wil Aalst, H.A. Reijers, A. Weijters, B.F. Dongen, A.K. Medeiros, Minseok Song, and H.M.W.(E.) Verbeek. Business process mining: an industrial application. inf. syst. 32(5), 713-732. *Information Systems*, 32:713–732, 07 2007.
4. Minseok Song and Wil M. P. van der Aalst. Towards comprehensive support for organizational mining. *Decis. Support Syst.*, 46(1):300–317, December 2008.
5. Gustavo Bernardi Pereira, Eduardo Alves Portela Santos, and Marcell Mariano Corrêa Maceno. Process mining project methodology in healthcare: a case study in a tertiary hospital. *Network Modeling Analysis in Health Informatics and Bioinformatics*, 9(1):1–14, 2020.
6. Wil Van Der Aalst. Data science in action. In *Process mining*, pages 3–23. Springer, 2016.
7. A. Weijters, Wil Aalst, and Alves Medeiros. Process mining with the heuristics miner-algorithm. *Cirp Annals-manufacturing Technology*, 166, 01 2006.
8. Sander JJ Leemans, Dirk Fahland, and Wil MP van der Aalst. Discovering block-structured process models from event logs containing infrequent behaviour. In *International conference on business process management*, pages 66–78. Springer, 2013.

9. Sander JJ Leemans, Dirk Fahland, and Wil MP van der Aalst. Discovering block-structured process models from incomplete event logs. In *International Conference on Applications and Theory of Petri Nets and Concurrency*, pages 91–110. Springer, 2014.
10. Sander JJ Leemans, Dirk Fahland, and Wil MP van der Aalst. Scalable process discovery with guarantees. In *Enterprise, Business-Process and Information Systems Modeling*, pages 85–101. Springer, 2015.
11. Ana Karla A de Medeiros, Anton JMM Weijters, and Wil MP van der Aalst. Genetic process mining: an experimental evaluation. *Data Mining and Knowledge Discovery*, 14(2):245–304, 2007.
12. Josep Carmona, Boudewijn van Dongen, Andreas Solti, and Matthias Weidlich. *Conformance Checking*. Springer, 2018.
13. Anne Rozinat and Wil MP Van der Aalst. Conformance testing: Measuring the fit and appropriateness of event logs and process models. In *International Conference on Business Process Management*, pages 163–176. Springer, 2005.
14. Arya Adriansyah and Joos CAM Buijs. Mining process performance from event logs. In *International Conference on Business Process Management*, pages 217–218. Springer, 2012.
15. Arya Adriansyah, Boudewijn F Van Dongen, and Nicola Zannone. Controlling break-the-glass through alignment. In *2013 International Conference on Social Computing*, pages 606–611. IEEE, 2013.
16. Arya Adriansyah, Jorge Munoz-Gama, Josep Carmona, Boudewijn F van Dongen, and Wil MP van der Aalst. Measuring precision of modeled behavior. *Information systems and e-Business Management*, 13(1):37–67, 2015.
17. Anastasiia Pika, Wil MP Van Der Aalst, Colin J Fidge, Arthur HM Ter Hofstede, and Moe T Wynn. Profiling event logs to configure risk indicators for process delays. In *International Conference on Advanced Information Systems Engineering*, pages 465–481. Springer, 2013.
18. Anna Kalenkova, Andrea Burattin, Massimiliano de Leoni, Wil van der Aalst, and Alessandro Sperduti. Discovering high-level bpmn process models from event data. *Business Process Management Journal*, 2019.
19. Ronny S Mans, Wil MP Van der Aalst, and Rob JB Vanwersch. *Process mining in healthcare: evaluating and exploiting operational healthcare processes*. Springer, 2015.
20. Hanna Yang, Minjeong Park, Minsu Cho, Minseok Song, and Seongjoo Kim. A system architecture for manufacturing process analysis based on big data and process mining techniques. In *2014 IEEE International Conference on Big Data (Big Data)*, pages 1024–1029, 2014.
21. G. Schuh, A. Gützlaff, S. Cremer, S. Schmitz, and A. Ayati. A data model to apply process mining in end-to-end order processing processes of manufacturing companies. In *2020 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, pages 151–155, 2020.
22. T. B. Hong Tu and M. Song. Analysis and prediction cost of manufacturing process based on process mining. In *2016 International Conference on Industrial Engineering, Management Science and Application (ICIMSA)*, pages 1–5, 2016.
23. Edson Ruschel, Eduardo Alves Portela Santos, and Eduardo de Freitas Rocha Loures. Establishment of maintenance inspection intervals: an application of process mining techniques in manufacturing. *Journal of Intelligent Manufacturing*, 31(1):53–72, 2020.
24. S Son, Bernardo Nurgroho Yahya, Minseok Song, Sangsu Choi, Jeongho Hyeon, Bumgee Lee, Yong Jang, and Nakyun Sung. Process mining for manufacturing process analysis: a case study. In *Proceeding of 2nd Asia Pacific Conference on Business Process Management, Brisbane, Australia*, 2014.
25. M.J. Shaw. *Information-Based Manufacturing: Technology, Strategy and Industrial Applications*. Springer US, 2012.
26. Stephen Cole Kleene. *Mathematical logic*. Courier Corporation, 2002.
27. Alessandro Berti, Sebastiaan J van Zelst, and Wil van der Aalst. Process mining for python (pm4py): bridging the gap between process-and data science. *arXiv preprint arXiv:1905.06169*, 2019.
28. Anthony J Myles, Robert N Feudale, Yang Liu, Nathaniel A Woody, and Steven D Brown. An introduction to decision tree modeling. *Journal of Chemometrics: A Journal of the Chemometrics Society*, 18(6):275–285, 2004.