

Analyzing Manufacturing Process By Enabling Process Mining on Sensor Data

Dina Bayomie^{a,d}, Kate Revoredo^c, Stefan Bachhofner^a, Kabul Kurniawan^{a,d},
Elmar Kiesling^a and Jan Mendling^c

^aVienna University of Economics and Business (WU), Vienna, Austria

^bCairo University, Cairo, Egypt

^cHumboldt University, Berlin, Germany

^dAustrian Center for Digital Production

Abstract

Typical manufacturing processes involve various machines, each of which may be equipped with a variety of sensors. Digital twins can be used to model how the machines operate and support analysts in issue identification and identifying potential improvements in the process. For a complete view of the status of a machine, however, models need to be enriched to identify patterns over changes in the measurements of sensors and correlations between these sensors. Process mining techniques could be usefully applied in this context, given that they provide descriptive analyses to explain and simulate physical objects based on event logs storing multi-perspective data about the process. However, although sensors generate a vast amount of data about the status of machines on the production floor, they cannot be directly used by process mining techniques. To tackle this issue, we introduce a method that creates a custom event log from sensor data based on the process analysts interests. To this end, we propose different encodings for the sensor data. An exploratory experiment using real-life data from an industrial partner shows the effectiveness of our approach.

Keywords

Sensor data, Event log creation, Process mining

1. Introduction

In the manufacturing industry, Digital Twins (DTs) [1] play increasingly important roles as the Industry 4.0 vision becomes reality. DTs facilitate the virtualization of physical objects to support analysts with an overview of the reality and a visualization of potential improvements and possible issues to be mitigated. An example of a physical object in a manufacture process is an injection molding machine used to produce plastic car parts through injection molding. As Figure 1 depicts, this machine can be equipped with numerous sensors. In this case having a model that focuses on how this machine operates is not sufficient for a full understanding of its status. In this context, it is beneficial to enrich the DT model with patterns of changes on the

PoEM'2022 Workshops and Models at Work Papers, November 23-25, 2022, London, UK

✉ dina.sayed.bayomie.sobh@wu.ac.at (D. Bayomie); kate.revoredo@hu-berlin.de (K. Revoredo);

stefan.bachhofner@wu.ac.at (S. Bachhofner); Kabul.Kurniawan@wu.ac.at (K. Kurniawan);


elmar.kiesling@ai.wu.ac.at (E. Kiesling); jan.mendling@hu-berlin.de (J. Mendling)

🆔 0000-0002-2549-6407 (D. Bayomie); 0000-0001-8914-9132 (K. Revoredo); 0000-0001-7785-2090 (S. Bachhofner);

0000-0002-5353-7376 (K. Kurniawan); 0000-0002-7856-2113 (E. Kiesling); 0000-0002-7856-2113 (J. Mendling)



© 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

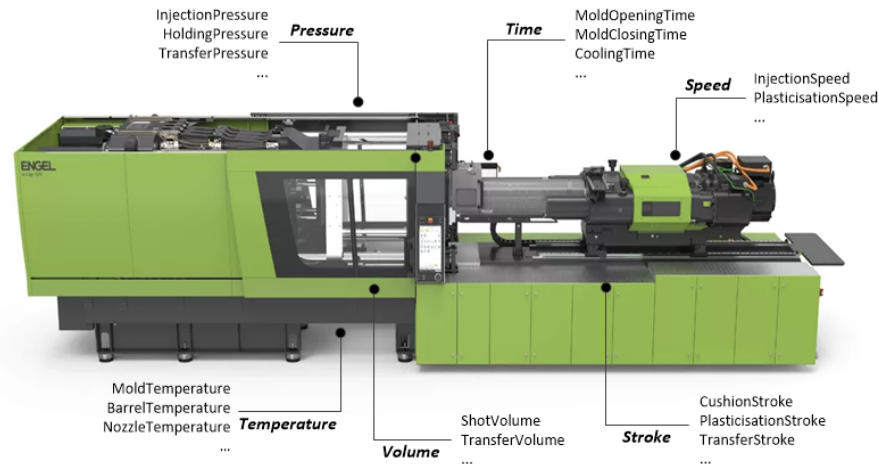


Figure 1: Injection Molding Machine instrumented with sensors.

measurements of sensors or correlations between these sensors, which may provide a more complete view of the machine status.

Process mining techniques [2] are able to provide descriptive analyses to explain and simulate physical objects. For that event logs storing multi-perspective data about the process are used. Sensors generate vast amounts of data about the status of machines on the production floor. However, this data cannot be directly used by process mining techniques, given that sensor measurements are structured as time serieses whereas event logs store the occurrence of discrete events of a process over time.

In this paper, we introduce a method that creates a custom event log from sensor data based on the process analysts interests. To this end, we explore multiple options to encode sensor measurements as process events and a set of techniques to group these events into cases, i.e., sequences of correlated events. We conduct exploratory experiments using real-life use case form our industrial partner Farplas. The experiments use specific encodings of the sensor data into an event log, which allow the use of process mining techniques for root-cause analysis and pattern discovery. The results shows the effectiveness of our approach.

The remainder of this paper is organized as follows. We discuss prior work in Section 2, describe our method in 3, and evaluate our method and discuss the findings in Section 4. Finally, we conclude our work and provide some future research directions in Section 5.

2. Related work

Various approaches to leverage process mining to support the analysis and modeling of real machines running on the shopfloor have been introduced in the context of Industry 4.0 [3]. In [4], an approach to integrate Digital Shadows to analyze shopfloor-level manufacturing processes is proposed. The different materials and sub-parts of a product are considered for the analysis. The integration of process events with structural data allows process mining techniques to learn an enriched process model. In this work, we consider a single machine with various sensors and focus on the change in the measurements of these sensors to define process

events. Our goal is to understand the correlation among these process events and how their interactions affect the output of the machine.

[5] addresses the scenario of using Digital Twins of organizations for business process improvement. To this end, process mining techniques are used to evaluate violations of constraints and produce required actions. A digital twin interface model is presented to make the current state of the business processes transparent, allowing the process analyst to visualize potential improvements to the business process.

In [6], an approach to transform sensor data into an event log is presented. The approach maps sensor measurements taken when users interact with smart products to human activities and grouping them into cases. The goal is to use the event log generated to discover models of human behavior. To this end, sensor data is encoded into an event log by segmenting the sensor measurements considering a fixed time window. Then the segments or groups of segments (when common characteristics are identified) are labeled as activities. The labeling process is performed manually by domain experts. Activities are grouped into cases based on the goal of the research, which is the relation of the sensor data with the user interaction. Therefore, each interaction with a user defines a new case. In the current work, we explore different ways for segmenting the sensor measurements and to create a case.

In [7], a method for analyzing time series data to be used in decision points in a process model is proposed. The approach learns a process model from an event log and if there are decisions in the model based on numeric attributes, then time series analysis is used to be considered at the decision point. In our work, we do not focus on decisions made based on the values of sensor data. Instead, we focus on investigating the changes in the sensor data to support the understanding of the process outcome.

In [8] a method for finding the interaction between sensor data and process knowledge is used to construct an event log. The sensor data used was the location of a specific object (e.g., the location of a patient in a hospital). In the current work, we focus on encoding different sensor data as an event log.

3. Method

In this section, we describe our method for creating an event log from sensor data, which we henceforth call EL-SD. It is inspired by the data science model [9].

Figure 2 illustrates the EL-SD process; the main data input is the sensor data, and the output is an event log. To create the event log, EL-SD relies on two main steps: (i) a pre-processing step in which the data is cleaned and an appropriate subset is selected, and (ii) an event log building step based on the process analyst's interest and the pre-processed data.

As shown in Figure 1, different types of sensors are associated with machines on the production floor. Each of these sensors provide time series data about the machine's status at a given time. Table 1 provides an example of the extracted entries of sensor data over time. The sensor data are raw data that track the machine status over time; Notice that "row #" column is a table pointer and does not denote the entry id. For example, in the second row, the time of the sensor reading is "2021-01-04 17:57:00 ", the temperature reading is 300.5, the pressure sensor reading is 222, the location sensor reading is L1, the volume sensor reading is 34.1, the

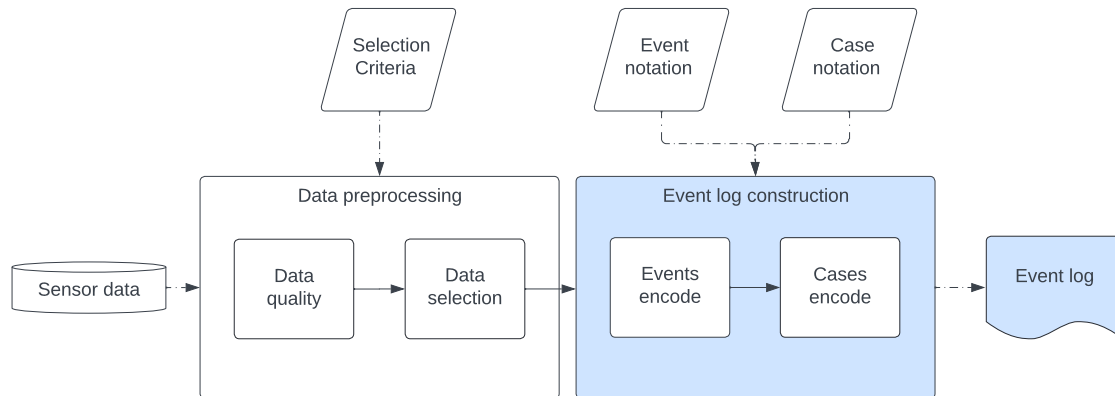


Figure 2: Creating an event log from sensor data (EL-SD)

Table 1

Sample of sensor data over the ending of two working days

| row # | Date | Temp | Pressure | Loc | Volume | Time | Speed |
|-------|---------------------|-------|----------|-----|--------|------|-------|
| 1 | 2021-01-04 17:55:30 | 300.0 | 222 | L1 | 34.0 | 34.0 | 100 |
| 2 | 2021-01-04 17:57:00 | 300.5 | 222 | L1 | 34.1 | 35.0 | 100 |
| 3 | 2021-01-04 17:58:30 | 300.6 | 221 | L25 | 34.3 | 35.5 | 90 |
| 4 | 2021-01-04 18:00:00 | 301.9 | 220 | L20 | 34.5 | 36.0 | 110 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| 5 | 2021-01-05 17:55:30 | 303.0 | 219 | L10 | 34.6 | 36.5 | 120 |
| 6 | 2021-01-05 17:57:00 | 303.5 | 219 | L1 | 34.9 | 37.0 | 100 |
| 7 | 2021-01-05 17:58:30 | 304.8 | 220 | L10 | 35.0 | 36.5 | 120 |
| 8 | 2021-01-05 18:00:00 | 304.9 | 219 | L1 | 35.0 | 38.0 | 100 |

time sensor reading is 35.0, and the speed sensor reading is 100.

3.1. Data Preprocessing

The first step is a preprocessing step to prepare the data in terms of data quality and data selection.

Data cleaning Time series data are essential in industry, where all kinds of sensor devices capture data from the industrial environment continuously. The time series data collected is typically large and affected by the limited reliability of sensor devices [10]. Consequently, data cleaning is an essential step [11] before subsequent analysis. Key quality issues with the sensor data are (i) unannotated data, i.e., the extracted data is not properly related to sensor metadata (e.g., sensor name, type etc.) (ii) Missing (null) values due to incomplete sensor readings. There are several techniques to handle such issues (cf. [11]).

Data Selection Due to the large number of sensors that production machines are typically instrumented with, the extracted sensor data is typically enormous. To perform meaningful

analyses, process analysts need to select data relevant for their analysis. The selection step may consist in slicing the data based on a time window, selecting specific sensors to analyze, or both. For example, suppose the analyst wants to understand the relation between temperature, pressure, and speed of the product produced on a machine. In that case, they can select the entries related to the respective sensors.

3.2. Event Log Construction

The second step of the EL-SD process consists in building an event log from the pre-processed time series sensor data.

3.2.1. Event log

In this section, we define the output of our method in terms with respect to *events*, *cases* and the constructed *event log*.

Definition 1 (Event). An event e represents a state in the system execution. An event has a set of attributes that describe it, mainly the *activity* attribute that defines what happens in this state and the *timestamp* attribute that marks when the state is happening.

Definition 2 (Case). A case $\sigma = \langle e_{\sigma_1}, \dots, e_{\sigma_m} \rangle$ is a finite sequence of length m of events e_{σ_i} with $1 \leq i \leq m$ induced by \preceq , i.e., such that $e_{\sigma_i} \preceq e_{\sigma_k}$ for every $i \leq k \leq m$. A case groups events to describe a specific object.

Definition 3 (Event log). An *event log* $L = \{\sigma_1, \dots, \sigma_n\}$ is a finite non-empty set of non-overlapping cases, i.e., if $e \in \sigma_i$, then $e \notin \sigma_j$ for all $i, j \in [1 \dots n]$, $i \neq j$.

Table 2 shows a possible event log L_1 containing three cases generated from the sensor data in Table 1. The first case $\sigma_1 = \langle e_1, e_2, e_3 \rangle$ has four events; e_1 is characterized by four attributes that describe the state of the system – e_1 .*Activity*, for instance, represents "Increase in Temp" from e_1 .*pvalue* = 300 to e_1 .*nvalue* = 300.5.

Based on Definition 3, an event log is defined through events that are grouped into cases. Analysts need to define what the events and cases represent. Our method provides several possible encodings for both the events and cases to build the event log. Inspired by [12], we use data objects for database schema to define different forms of cases and generate various event logs. In the following subsection, we explain different ways of defining the events and cases over sensor data.

3.2.2. Event encoding

There are various options for defining what constitutes an event in the context of sensor data that can help process analysts. As per Definition 1, an event describes the status of the system execution, therefore the analyst can determine how to represent this status. We propose five options to define an event and mainly express the event activity.

The *first* and naïve option is that every entry is an event representing the system status at this point. The event represents one activity ("Record status"). This encoding would be interesting

Table 2

A sample event log representing the sensor data in Table 1

| Case Id | Activity | Timestamp | pvalue | nvalue |
|---------|----------------------|---------------------|--------|--------|
| 1 | Increase in Temp | 2021-01-04 17:57:00 | 300 | 300.5 |
| 1 | Increase in Volume | 2021-01-04 17:57:00 | 34 | 34.1 |
| 1 | Increase in Time | 2021-01-04 17:57:00 | 34 | 35 |
| 2 | Increase in Temp | 2021-01-04 17:58:30 | 300.5 | 300.6 |
| 2 | Decrease in Pressure | 2021-01-04 17:58:30 | 222 | 221 |
| 2 | Change in Loc | 2021-01-04 17:58:30 | L1 | L25 |
| 2 | Increase in Volume | 2021-01-04 17:58:30 | 34.1 | 34.3 |
| 2 | Increase in Time | 2021-01-04 17:58:30 | 35 | 35.5 |
| 2 | Decrease in speed | 2021-01-04 17:58:30 | 100 | 90 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| 300 | Increase in Temp | 2021-01-05 18:00:00 | 304.8 | 304.9 |
| 300 | Decrease in Pressure | 2021-01-05 18:00:00 | 220 | 219 |
| 300 | Change in Loc | 2021-01-05 18:00:00 | L10 | L1 |
| 300 | Increase in Time | 2021-01-05 18:00:00 | 36.5 | 38 |
| 300 | Decrease in speed | 2021-01-05 18:00:00 | 120 | 100 |

Table 3

Entry encoding for the sensor data readings of the first four rows in Table 1

| Activity | Timestamp | Temp | Pressure | Loc | Volume | Time | Speed |
|---------------|---------------------|-------|----------|-----|--------|------|-------|
| Record status | 2021-01-04 17:55:30 | 300.0 | 222 | L1 | 34.0 | 34.0 | 100 |
| Record status | 2021-01-04 17:57:00 | 300.5 | 222 | L1 | 34.1 | 35.0 | 100 |
| Record status | 2021-01-04 17:58:30 | 300.6 | 221 | L25 | 34.3 | 35.5 | 90 |
| Record status | 2021-01-04 18:00:00 | 301.9 | 220 | L20 | 34.5 | 36.0 | 110 |

for analyzing the data perspective over sensor data by considering all of them as event data for the log using root-cause analysis techniques over the data. Table 3 shows an example of encoding the events in which an event represents the activity "system status", and all the sensor data are attributes that describe this status.

The *second* option encodes the events to represent changes in the system over each sensor between two successive readings of sensor data. Thus, we would have event activities of "change in sensor x" for each sensor that changed over the readings. Also, we maintain the status of the changes over the sensor data, whether it is increasing or decreasing. This encoding helps to analyze the detailed changing patterns over sensor data readings. Table 4 shows an example of encoding the events in which an event represents the change over the sensor data, and the attributes describe this change in terms of the previous value, i.e., "pValue" attribute, new value, i.e., "nValue" attribute and status attribute that indicate whether the sensor reading is increasing or decreasing. Notice that we do not state any change status for non-numerical sensor such as location (Loc) sensor.

The *third* option encodes the events similar to the previous encoding. However, an event represents the change over each sensor between two successive readings of sensor data when

Table 4

Detailed change encoding for the sensor data readings of the first three rows in Table 1

| Activity | Timestamp | pvalue | nvalue | status |
|------------------|---------------------|--------|--------|--------|
| Change in Temp | 2021-01-04 17:57:00 | 300 | 300.5 | + |
| Change in Volume | 2021-01-04 17:57:00 | 34 | 34.1 | + |
| Change in Time | 2021-01-04 17:57:00 | 34 | 35 | + |

Table 5

Threshold change encoding for the sensor data readings of the first three rows in Table 1, given a threshold of 0.5 for all numerical sensors

| Activity | Timestamp | pvalue | nvalue | status |
|----------------|---------------------|--------|--------|--------|
| Change in Temp | 2021-01-04 17:57:00 | 300 | 300.5 | + |
| Change in Time | 2021-01-04 17:57:00 | 34 | 35 | + |

Table 6

Aggregated change view of sensor data in Table 1

| Activity | Timestamp | minimum | average | maximum |
|--------------------|------------|---------|---------|---------|
| Change in Temp | 2021-01-04 | 300 | 300.75 | 301.9 |
| Change in Pressure | 2021-01-04 | 220 | 221.25 | 222 |
| Change in Volume | 2021-01-04 | 34 | 34.2 | 34.5 |
| Change in Time | 2021-01-04 | 34 | 35.125 | 36 |
| Change in speed | 2021-01-04 | 90 | 100 | 110 |
| Change in Temp | 2021-01-05 | 303 | 304.05 | 304.9 |
| Change in Pressure | 2021-01-05 | 219 | 219.25 | 220 |
| Change in Volume | 2021-01-05 | 34.6 | 34.875 | 35 |
| Change in Time | 2021-01-05 | 36.5 | 37 | 38 |
| Change in speed | 2021-01-05 | 100 | 100 | 110 |

the change exceeds a given *threshold*. Thus, we would have event activities of "change in sensor x" for each sensor within the readings such that the change over the two successive readings exceeds the analyst threshold for this sensor. Also, we maintain the previous and new values of the exceeding change and the status of the changes over the sensor data, whether increasing or decreasing. This encoding helps analyze to analyze interesting changing points within the sensor data readings based on the analyst thresholds. Table 5 shows an example of encoding the events based on changing threshold. The number of events generated using the threshold is less than that generated using the second encoding in Table 4 over the same sensor readings. Therefore, analysts can focus on changing points instead of an excessive number of granular changes.

The *fourth* option encodes the events to represent an aggregation overview of numerical sensor readings over a given time window specified by the analyst. There are two possibilities for event activities.

Table 7

Aggregated entry view of sensor data in Table 1

| Activity | Timestamp | Temp | Pressure | Volume | Time | Speed |
|---------------|------------|--------|----------|--------|--------|-------|
| Record status | 2021-01-04 | 300.75 | 221.25 | 34.2 | 35.125 | 100 |
| Record status | 2021-01-05 | 304.05 | 219.25 | 34.875 | 37 | 110 |

Table 8

Various activity types over the changes of the 2nd and 3rd rows in Table 1

| Activity | Timestamp | pvalue | nvalue |
|----------------------|---------------------|--------|--------|
| Increase in Temp | 2021-01-04 17:58:30 | 300.5 | 300.6 |
| Decrease in Pressure | 2021-01-04 17:58:30 | 222 | 221 |
| Change in Loc | 2021-01-04 17:58:30 | L1 | L25 |
| Increase in Volume | 2021-01-04 17:58:30 | 34.1 | 34.3 |
| Increase in Time | 2021-01-04 17:58:30 | 35 | 35.5 |
| Decrease in speed | 2021-01-04 17:58:30 | 100 | 90 |

First, we may conceive event activities as "changing in sensor behavior on average x" for each sensor that changed over the readings during the time window. Also, we maintain the minimum, average, and maximum values of the changes. This encoding can be combined with the third encoding option by allowing the analyst to provide change thresholds for the sensors. Table 6 shows an example of encoding the average changes of each numerical sensor over a one-day time window.

The second possibility is similar to the first option, setting the event activity to "Record the aggregate status". Then, we take the average of the sensor readings during the time window for each numerical sensor and consider them as event data attributes. Table 7 shows an example of encoding the average values of each numerical sensor over a one-day time window.

The idea of this encoding is to provide an aggregated view of the enormous amount of sensor data. Moreover, allow the analyst to see the changes over the aggregated values.

The *fifth* option encodes the events to represent the change in the system over each sensor between successive readings of sensor data. Unlike the second encoding, however, it has three different event activities that alter based on the changing behavior, such that (i) if the sensor data is non-numerical, then it is "change in sensor x"; (ii) if it is numerical and the change is increasing, then the event activity is "Increase in sensor x", and finally, (iii) if it is numerical and decreasing then it is "Decrease in sensor x". Also, we maintain the previous and new values of the changes over the sensor readings. This encoding can be combined with the third encoding option by allowing the analyst to provide change thresholds for the sensors.

As shown in Table 8, the first event indicates an increase in temperature sensor reading from 300 to 300.5.

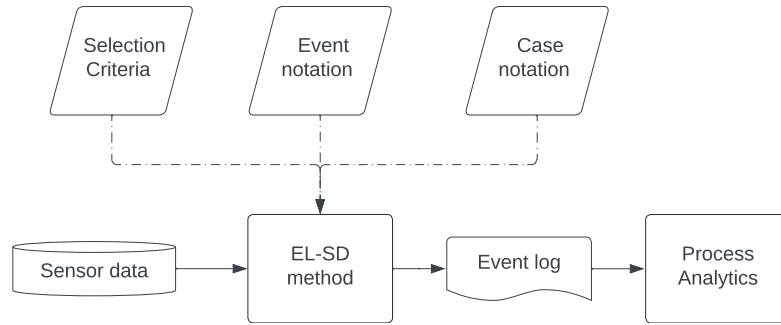


Figure 3: Evaluation process

3.2.3. Case encoding

After defining the event notation, we need to define the case notation to group the events and generate the logs for further analysis. There are several ways to group the events to formulate a case. We present two possible case encodings that can be combined with any of the event encodings presented in the previous section.

The first option is grouping the events to represent the change over two successive sensor readings. As shown in Table 2, the events are grouped based on the change over the successive data entries.

The second option is grouping the events based on the time window so that all events that occur within the same time window belong to the same case. For example, all events that occur on the same day in Table 6 have the same case id, so there will be two cases over these events.

Generating an event log allows the process analysts to use different process analytic techniques that improve the understanding and explainable of the manufacture process status.

4. Evaluation

Figure 3 shows the evaluation process we followed to conduct two exploratory experiments based on a prototypical implementation of our approach ¹. To evaluate the usefulness of the EL-SD method, we then used process analytics tools over the generated logs. Specifically, we used Disco² and EL-RM ³[13].

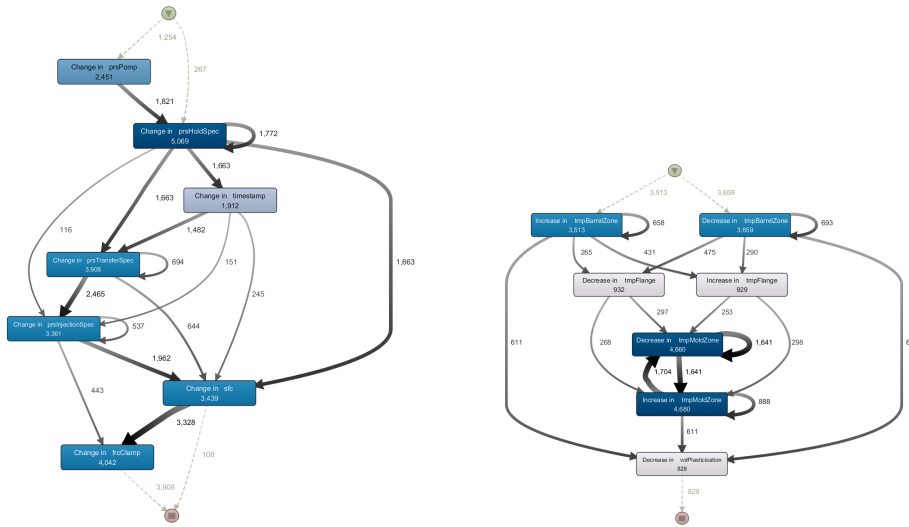
4.1. Dataset

We used one dataset from our industry partner in Teaming AI project, Farplas. Farplas is a full system solutions partner for the automotive industry. Farplas researches, develops, and manufactures superior automotive polymer systems, provides innovative solutions, and implements state-of-the-art technologies. We use their sensor data that describe the status of the production floor machines for polymer systems production, such as temperature, pressure, and volume

¹Build event log <https://github.com/DinaBayomie/GenerateEventLogFromSensorData>

²<https://www.fluxicon.com/disco/>

³<https://github.com/DinaBayomie/EL-RM>



(a) Cases per changes over detailed change events (b) Cases per day over detailed changes with different activities events

Figure 4: Changing pattern analysis: process model

sensors. The dataset contains 48 sensors with three non-numerical sensors and one quality detection sensor data.

We conducted two exploratory experiments with different analysis objectives to explore the usefulness of our method and the effectiveness of creating event logs from sensor data. Each experiment addresses an analysis scenario over the data to explore the benefits of having a specific event log that meets the process analyst’s interests.

4.2. Scenario 1: Change Pattern Analysis

The first experiment explores the sensor data using process discovery techniques to understand the changing patterns over the sensors. We generated two event logs using different event and case encodings. Then, we used Disco to discover the process model over these logs.

Figure 4 shows the process models discovered from the two generated event logs. Figure 4a illustrates the most changing patterns given that a case contains the events generated over changes of two successor entries. Using this encoding, the analyst can easily capture the most frequent sensors that change almost every reading. That helps to understand the physical state of the manufacturing machines over every reading by providing a virtual model that represents real-time changes over the sensors.

Figure 4b depicts the changing patterns over 22 days, given that each day represented a case. An event describes the changes over two successor entries with three possible activities that clarify the changes over the sensor readings (see the fifth encoding option in subsection 3.2.2). Using this encoding, the analyst can quickly grasp the most regular sensors that change over the days. Also, the analyst can see the changing behavior in terms of increasing or decreasing

the sensors' readings and how they affect each other. That helps to understand the physical interaction between the different environmental parameters, e.g., temperature, pressure, and speed, of the manufacturing machines over the day by modeling a virtual model that represents the daily behavior of the sensors.

Using the process models, analysts can investigate various patterns of the sensors. Being able to use different encodings allows them to explore the sensor day from multiple perspectives.

4.3. Scenario 2: Root cause Analysis

The second experiment performs a root-cause analysis to understand the relationship between the sensor data and the quality detection data. We generated one event log in which the case contained events that occurred on the same day. We encoded the events following the first encoding insubsubsection 3.2.2 in which each entry represents an event in order to analyze the sensor data from a data perspective. Then, we discovered the association rules over these logs using EL-RM[13]. We focus on the association rules that concentrate on the detection sensor data. Therefore, we select only the rules that include the not-ok quality detection data as a consequence of the rule. Following that, EL-RM discovered 20 association rules with a confidence of 0.9 that show a possible association between the changes over the sensors and the not-ok quality detection data. Using this encoding allows the analyst to conduct a root-cause analysis and gaining insights into the influence of the machine status from the sensor data over the quality detection control.

4.4. Discussion

As shown in both exploratory experiments, generating an event log from the sensor data supports creating a virtual model to understand the manufacturing of physical objects. Moreover, it allows the analysts to explore the sensor data using process analysis techniques that investigate it from a new perspective other than the time-series analysis. Also, it enriches the event model of Teaming.AI [14] which helps understand and support the AI agent.

5. Conclusion

In this paper, we proposed a method (EL-SD) for creating an event log from sensor data. Our method consists of two steps: a pre-processing step allows the analysts to prepare the data by applying data cleaning and selection techniques; the main step then creates the log based on the event notation and case notation specified by the analyst. We provide five event encoding options and two case encoding options. The results of our exploratory case scenarios show the potential of the method to investigate the sensor data from different perspectives and provide new insights into the production floor. Moreover, it constructs a virtual model that can contribute to accurate digital twins that capture the dynamic behavior manufacturing machines. As future work, we will investigate different ways to define the event activities such that they reflect the production process.

Acknowledgments

This work received funding from the Teaming.AI project in the European Union's Horizon 2020 research and innovation program under grant agreement No 95740. The work of J. Mendling was supported by the Einstein Foundation Berlin.

References

- [1] C. Semeraro, M. Lezoche, H. Panetto, M. Dassisti, Digital twin paradigm: A systematic literature review, *Comput. Ind.* 130 (2021) 103469.
- [2] W. M. P. van der Aalst, *Process Mining - Data Science in Action*, Second Edition, Springer, 2016.
- [3] S. Dreher, P. Reimann, C. Gröger, Application fields and research gaps of process mining in manufacturing companies, in: R. H. Reussner, A. Koziolk, R. Heinrich (Eds.), *INFORMATIK 2020*, Gesellschaft für Informatik, Bonn, 2021, pp. 621–634. doi:10.18420/inf2020_55.
- [4] T. Brockhoff, M. U. Seran, W. van der Aalst, Modeling digital shadows in manufacturing by using process mining, *Modellierung 2022 Satellite Events (2022)*.
- [5] G. Park, W. M. P. van der Aalst, Realizing A digital twin of an organization using action-oriented process mining, in: *ICPM, IEEE*, 2021, pp. 104–111.
- [6] M. L. van Eck, N. Sidorova, W. M. P. van der Aalst, Enabling process mining on sensor data from smart products, in: *2016 IEEE Tenth International Conference on Research Challenges in Information Science (RCIS)*, 2016.
- [7] R. Dunkl, S. Rinderle-Ma, W. Grossmann, K. Anton Fröschl, A method for analyzing time series data in process mining: Application and extension of decision point analysis, in: S. Nurcan, E. Pimenidis (Eds.), *Information Systems Engineering in Complex Environments*, Springer International Publishing, Cham, 2015, pp. 68–84.
- [8] A. Senderovich, A. Rogge-Solti, A. Gal, J. Mendling, A. Mandelbaum, The ROAD from sensor data to process instances via interaction mining, in: *CAiSE*, volume 9694 of *Lecture Notes in Computer Science*, Springer, 2016, pp. 257–273.
- [9] M. Souibgui, F. Atigui, S. Zammali, S. Cherfi, S. B. Yahia, Data quality in etl process: A preliminary study, *Procedia Computer Science* 159 (2019) 676–687.
- [10] S. R. Jeffery, G. Alonso, M. J. Franklin, W. Hong, J. Widom, Declarative support for sensor data cleaning, in: *International Conference on Pervasive Computing*, Springer, 2006.
- [11] X. Wang, C. Wang, Time series data cleaning: A survey, *Ieee Access* 8 (2019) 1866–1881.
- [12] E. G. L. de Murillas, H. A. Reijers, W. M. van der Aalst, Case notion discovery and recommendation: automated event log building on databases, *Knowledge and Information Systems (2019)*.
- [13] D. Bayomie, K. Revoredo, J. Mendling, Multi-perspective process analysis: Mining the association between control flow and data objects, in: *CAiSE*, 2022, pp. 72–89.
- [14] T. Hoch, B. Heinzl, G. Czech, M. Khan, P. Waibel, S. Bachhofner, E. Kiesling, B. Moser, Teaming. ai: enabling human-ai teaming intelligence in manufacturing, in: *11th International Conference on Interoperability for Enterprise Systems and Applications*, volume 3214, 2022, p. 0073.