The final authenticated version is available online at https://doi.org/10.1007/978-3-030-79108-7_4.

Sensor Data Stream Selection and Aggregation for the Ex Post Discovery of Impact Factors on Process Outcomes*

Matthias Ehrendorfer[1], Juergen Mangler[2], and Stefanie Rinderle-Ma[2]

[1] University of Vienna, *Faculty of Computer Science*, Vienna, Austria
matthias.ehrendorfer@univie.ac.at
[2] Technical University of Munich, *Department of Informatics*, Garching, Germany
{juergen.mangler,stefanie.rinderle-ma}@tum.de

Abstract. One target of process analysis, monitoring, and prediction is the process outcome, e.g., the quality of a produced part. The process outcome is affected by process execution data, including (external) sensor data streams, e.g., indicating an overheating machine. Challenges are to select the "right" sensors –possibly a multitude of sensors is available– and to specify how the sensor data streams are aggregated and used to calculate the impact on the outcome. This paper introduces process task annotations to specify the selected sensors, their aggregation, and initial impact functions. The initial impact functions are then refined, e.g., threshold values and the impact of sensor data streams are determined. The approach is prototypically implemented. Its applicability is demonstrated based on a real-world manufacturing scenario.

Keywords: Sensor Data Streams, Process Outcome, Process Impact Analysis

1 Introduction

Companies want to execute processes efficiently by exploiting all available possibilities to avoid undesired outcomes. However, data from sensors and machines being used in the process is often not taken into account, for example, when it does not directly contribute to the control flow of a process. Nonetheless, such data might determine the outcome of tasks or the process itself. This leads to a situation where experienced process operators can anticipate the progression and (final or intermediate) results of a process because they know (1) what they should pay attention to and (2) which behaviour signalises which outcomes.

To formalise the knowledge of experienced process operators and make it available at run-time for outcome prediction, we introduce the concept of *Impact*

* This work has been partly funded by the Austrian Research Promotion Agency (FFG) via the "Austrian Competence Center for Digital Production" (CDP) under the contract number 881843. This work has been supported by the Pilot Factory Industry 4.0, Seestadtstrasse 27, Vienna, Austria.

Factors. Impact factors can be derived from process data and external data, e.g., sensor data. The latter is implicitly connected to process tasks and can hold the key to predict the process outcome (see Fig. 1). In most cases, sensor data occurs in the form of a series of data points because machines and sensors measure continuously [7].

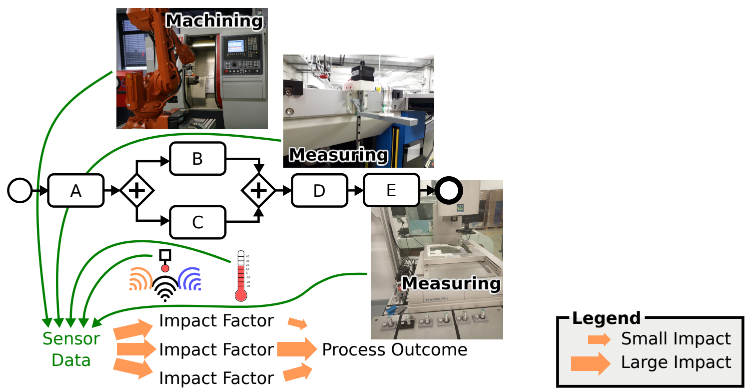


Fig. 1: Deduction of Impact Factors From Sensor Data

Determining impact factors based on sensor data raises several challenges. At first, in realistic settings, one has to possibly choose relevant sensors from a multitude of sensors. Secondly, the accessibility of the information, especially at the presence of many sensor streams is crucial. Third, the first and second point require to compare the impact of single sensors vs. the impact of a combination of sensors. Fourth, run-time deviations in sensor information might occur.

These challenges lead to the following research questions. (*RQ1*) How to annotate process models with data sources that are relevant in the context of process task execution? (*RQ2*) How to aggregate and contextualise sensor data for arbitrary process tasks at run-time? (*RQ3*) How to deduce impact factor predictions from the aggregated sensor data?

To tackle *RQ1* – *RQ3*, this paper provides a method for the structured collection, classification, and correlation of sensor-based impact factors to compare process instances and track their progression. The proposed method works along the following steps: (1) Annotating process tasks to define the extraction of series of data points from data streams (e.g., from sensors) that occur during the execution of those tasks. (2) Annotating process tasks to enable aggregation of the extracted data series. (3) The semi-automatic extraction of impact factors from the collected data. Steps (1) – (3) are implemented in a manufacturing scenario and evaluated based on the corresponding data set. The manufacturing process includes a comprehensible quality assurance as last step, thus allowing for the assessment of the quality of the approach presented in this paper.

The structure is as follows: Section 2 annotates sensors to process tasks and identifies impact factors. Section 3 evaluates the approach. Section 4 discusses the results and Sect. 5 related work. The paper is concluded in Sect. 6.

2 Approach

Manufacturing Scenario: The case presented in this paper is a manufacturing process carried out in the "Pilot Factory Industry 4.0" (`http://pilotfabrik.tuwien.ac.at/en/`). The steps of the manufacturing process are enacted by a number of machines, humans, and software systems. The orchestration of the steps is defined in a BPMN process model which is executed by a workflow engine. In addition to process data, (sensor-)data is collected. Considering both allows for a deeper analysis of the process along with the possibility for improvements. The manufacturing process used for this paper consists of the following steps on a high abstraction level: (1) Manufacturing with a turning machine. (2) Automated optical quality control measurement directly after part production - fast ($\sim$ 20 seconds) but imprecise measurement. (3) Automated final quality control measurement - precise but slow ($\sim$ 480 seconds) measurement.
The **Solution Design** aims at annotating process tasks with sensor data.

Sensors define what is extracted from associated data streams and how. As machine and sensor data is often not represented in processes, it must be collected inside the tasks themselves. Therefore, defining how to handle these data streams is necessary, e.g., split data from one machine (like temperature and noise level) or merge data from different sensors (like partial temperature readings).

Aggregators describe how to aggregate the extracted data for analysis. This is necessary because it has to be taken into account that sensors measure differently and therefore different characteristics of a measurement need to be used. For example, measuring a part might result in a massive point-cloud, describing a set of different properties that a part has. An aggregator might (a) throw some data away, and (b) group data so that it becomes accessible for later analysis.

Impact Functions operate on aggregated data, and define how to calculate the deviation between current data and expected data. Expected data leads to the desired outcome, current data might not. An impact function consists of two parts: (1) an expected target value or data pattern, and (2) a function that describes how much a deviation affects the overall process. For example for measurements there may be a certain tolerance until which a part is accepted but when the tolerance is exceeded, the part is considered faulty.

These three types of annotations can be used to derive an **impact value (IV)** by using an **aggregation (A)** for a particular **sensor (S)**. Based on one or more *impact values* together with an **impact function (IFU)** it can be defined how the *impact value(s)* are combined to retrieve an **impact factor (IF)**:

$$IV = (S, A) \text{ and } IF = (IV+, IFU)$$

One or more *impact factors* can then be used to build **impact profiles**, either for individual tasks **(TIP)** or for the whole process **(PIP)**. The combination of the *impact factors* into *impact profiles* is facilitated by an **impact profile function (IPF)**, which works similar to the *IFU* introduced above.

$$TIP = (IF+, IPF) \text{ and } PIP = ((IF, A)+, IPF)$$

While *impact profiles of tasks (TIP)* only use *IFs* of one task, *impact profiles of a process (PIP)* use *IFs* from multiple tasks. *PIPs* therefore need to handle the *aggregation* of *impact factors* differently, because an *impact factor* can be encountered more than once (e.g., in a loop).

When trying to find relevant impact factors for specific outcomes, several pieces of information need to be provided by a human. Firstly, the sensors and data handling have to be specified. E.g., as multiple sensors might be contained in one stream, it has to be split into different data series. This holds for the real-world data set used in this paper, as the turning machine delivers a total of 27 sensors in one data stream. Secondly, one or more aggregation methods that define how the extracted data is interpreted (e.g., is only a specific segment of the measurement important, is only the average of all values important, ...) need to be specified. Lastly, the general impact function (telling how the aggregation of sensor data behaves compared to one where a desired result is achieved) needs to be defined by the user. However, the first interaction of an impact function is seldom the optimum. Thus the impact function is typically refined after enough instances of a process are executed. Furthermore, the actual influence of an impact factor on an outcome is not given, as it is also unknown at design time. The presented approach tries to determine these two missing values based on executed process traces.

Process outcome: Impact factors have to be refined by determining the optimal impact function parameters as well as the influence of a specific impact factor on the outcome. This refinement requires the following steps: (1) Describe the characteristics of different sensors, i.e., how to aggregate individual values and initial impact functions. (2) Based on executed process traces calculate for each sensor for a specific outcome (a) the ROC curve and AUC value and (b) the impact function parameters to achieve the maximum accuracy. (3) Calculate the influence of individual impact factors on an outcome by using (a) the AUC value or (b) the accuracy achieved with the optimal impact function parameters. (4) Based on the refined impact factors, traces can now be assigned a value showing the severity of dissatisfied impact factors. This value makes it possible to distinguish between different results of the analysed outcome.

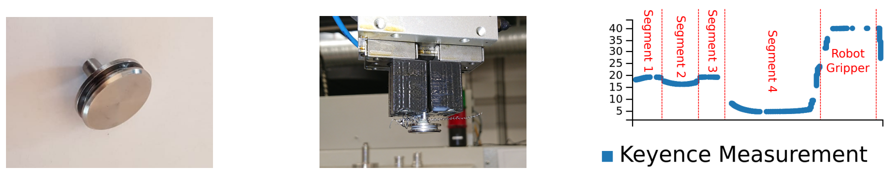
The impact of individual factors on the outcome can be determined by higher AUC or accuracy values. Using the share of the majority class for the accuracy or a diagonal ROC curve for the AUC value as a baseline (i.e., minimum expectation for the influence of an impact factor) rewards influential factors and penalises bad ones, thus compensating classes with a high share compared to other classes.

3 Evaluation

The log traces of two batches, referred to as batch 14[3] (38 parts) and 15[4] (41 parts), are used for the evaluation. Both batches contain a valve lifter for a gas turbine (Fig. 2a), produced in a real-world factory setting. The part is produced

[3] `http://cpee.org/~demo/DaSH/batch14.zip` [Online; accessed 02-April-2021]
[4] `http://cpee.org/~demo/DaSH/batch15.zip` [Online; accessed 02-April-2021]

in a turning machine and taken out by a robot (Fig. 2b). Then the diameter of the part's silhouette is measured by a Keyence measuring machine [5]. Based on semantic knowledge, different segments of the measurement time-series can be identified (Fig. 2c). Finally, a slow but more precise measurement is performed.



(a) Part Without Chip (b) Part Held by Robot (c) Measurement Segments

Fig. 2: Relationship Between Physical Parts and Optical Measurement

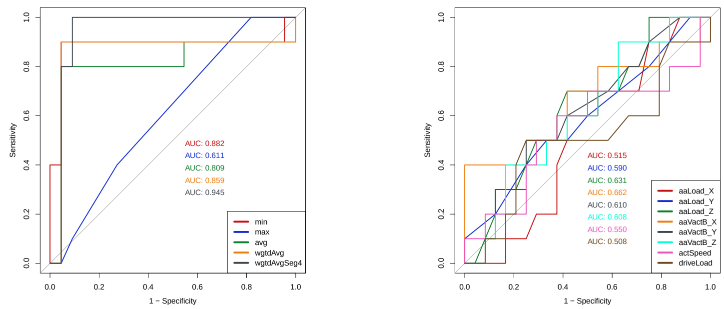
We use data from a measuring machine yielding the diameter of the part's silhouette and a measuring machine yielding multiple time-series: the workload of the drive (aaLoad) in percent and the axis speed (aaVactB) in millimeters per minute for X, Y, and Z axis alongside the actual speed of the spindle (actSpeed) in turns per minute and the workload of the spindle (driveLoad) in percent. Five aggregation methods are used: min, max, avg, wgtdAvg and wgtdAvgSeg4. To handle different measurement intervals, weighted average (wgtdAvg) assumes that a value is valid until a new one is measured. "Segment 4" (wgtdAvgSeg4, see Fig. 2c) uses only values occurring 5200 to 9600 milliseconds after the first data point. All five aggregation methods are used for analysing the optical measurement and the weighted average is used for the eight sensors observed during machining. A threshold which defines a boundary between different outcome classes is used as method for detecting violations of aggregated sensor data.

The evaluation examines (1) the occurrence of chips (only batch 15) and (2) the result of the "Zylinder Ø4,5-B – Durchmesser" quality control test.

Chip Prediction Using minimum, average, weighted average, and weighted average of "Segment 4" of the faster but less precise "Keyence" measurement leads to results with a high sensitivity and specificity while the maximum has less impact (Fig. 3a). It can be seen that using the weighted average of machining data (Fig. 3b) does not show if there is a chip.

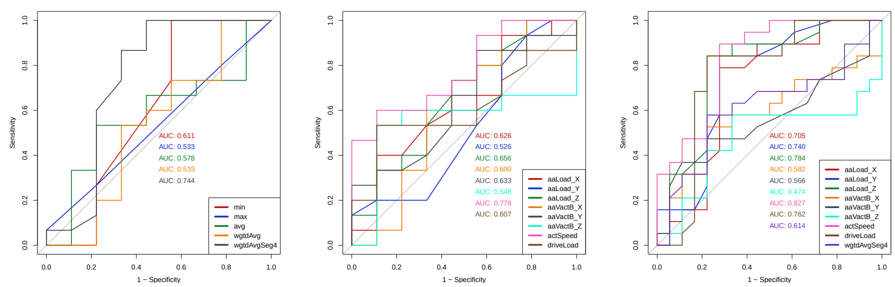
Quality Control Test Prediction When ignoring parts with chips, the weighted average of "Segment 4" has the highest impact (see Fig. 4a). For machining data (see Fig. 4b), actSpeed is the most promising impact factor. Using batch 14 leads to the results shown in Fig. 4c again highlighting actSpeed. "Keyence" measurements are not used because there is no possibility to exclude parts with chips which leads to bad optical measurements.

[5] `https://www.keyence.com/products/measure/micrometer/ls-9000/` [Online; accessed 02-April-2021]



(a) Based on Keyence Data (b) Based on Machining Data

Fig. 3: ROC Curves for Predicting Chip Occurrence for Batch 15



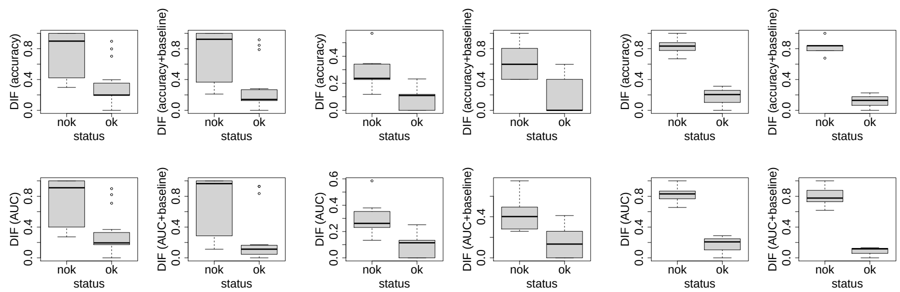
(a) Batch 15 Keyence (b) Batch 15 Machining (c) Batch 14 Machining

Fig. 4: ROC Curves for Predicting Quality Control Test

Using Impact Factors For Outcome Anticipation The result of calculating the overall impact with the available sensor data is shown in Figs. 5a and 5b for batch 14 and 15. Furthermore, Fig. 5c shows batch 14 results using a training set (75%/27 parts) and test set (25%/9 parts). Due to the low number of parts, data is only split to validate the results achieved and not for all analysis steps.

The results show the overall dissatisfied impact factors (DIF) using different methods (see Sect. 2). The weighted average in segment 4 of the "Keyence" measurement and the weighted average of the machining data are used as impact factors. Although not perfectly separated, higher overall sums of dissatisfied impact factors are calculated for parts being not ok (regarding the quality control test). The effect of using a baseline can be seen in Fig. 5. It has a stronger effect on batch 15 (nearly two-thirds of the part belong to one class) than on batch 14 (classes are evenly distributed). Identifying and combining impact factors as discussed above, is the basis for creating impact profile functions as defined in Sect. 2. The source-code used for the evaluation is available at gitlab [6].

[6] `https://gitlab.com/me33551/impact-factor-determination` [Online; accessed 02-April-2021]



(a) Batch 14 (b) Batch 15 (c) Batch 14 Test set

Fig. 5: Dissatisfied Impact Factors (DIF) for Quality Control Test

4 Discussion

Overall, the presented approach uses sensor data collected during process execution to identify impact factors influencing certain process outcomes. Only the impact of individual sensors on outcomes is examined, a next step would be to consider interdependent impact factors. Also the combination of impact factors to anticipate the outcome should be further examined as only one method (assigning weights representing the influence on the outcome) is exemplary used.

This paper focuses on manufacturing processes. However, transferring the approach to other domains would be interesting. The goal of the approach is to predict the outcome of a process by using sensor data collected throughout the process. A similar use case is the medical domain: the results of examinations or the dosing of administered medication can be collected as data while the health condition of the patient or the costs of the overall therapy process represent outcomes. Another application domain is logistics where sensors measuring temperature, speed, or concussion e.g, in vehicles can be used to find out how long the delivery of a product will take and in which condition it will arrive.

5 Related Work

Process mining mainly focuses on the control-flow perspective [1]. Some approaches examine further perspectives, also referred to as multi-perspective process mining [4], for example, process data [6]. The analysis of time sequence data for explaining concept drifts during run-time is tackled in [6, 7]. By contrast, the presented approach aims at including data stream information into the process model to make it usable. [8] compares different outcome-oriented predictive process monitoring techniques. However, the real-life event logs used in [8] do not contain detailed sensor data meaning there is no need to annotate process tasks.

The presented approach can also be positioned in the context of IoT and BPM based on the challenges provided in [3]. The annotation of process tasks with sensor data contributes to C3 (Connection of analytical processes with IoT).

Furthermore, C1 (Placing sensors in a process-aware way) is addressed because only explicitly represented sensors allow sensor-aware placement of new ones.

Building on the ideas of [5], this paper focuses on data collection in the context of the enacted process. Earlier work in this field includes [2] which analyses the log files of manufacturing processes containing contextualised data. This paper goes beyond this by explicitly representing data streams in the process model and analysing them with respect to different outcomes of the overall process.

6 Conclusion

This paper presents a way to annotate process tasks for contextualised data collection using (1) “Sensors” defining which data is collected and how this is done, (2) “Aggregators” describing how to aggregate it, and (3) “Impact Functions” allowing to detect the violation from expected sensor behaviour. This provides a basis for finding impact factors. Furthermore, different aggregation methods are evaluated and the conclusion that generic methods like minimum, maximum, or average can already reveal some characteristics is reached. However, advanced aggregations adjusted to the domain and specific case can yield in-depth analysis results. Finally, different methods to determine the impact of aggregated sensor data utilising accuracy and AUC value are presented. After classifying the aggregated data based on a threshold, the overall number of dissatisfied impact factors can then be obtained by combining them based on their influence on the outcome. The evaluation, based on a real-world data set, shows that deducing impact factors allows the prediction of quality variations. Supporting users in defining impact factors depends on domain knowledge, further automating this to improve prediction quality will be the subject of future work.

References

1. van der Aalst, W.M.P.: Process Mining - Data Science in Action, Second Edition. Springer (2016)
2. Ehrendorfer, M., Fassmann, J., Mangler, J., Rinderle-Ma, S.: Conformance checking and classification of manufacturing log data. In: 2019 IEEE 21st Conference on Business Informatics (CBI). vol. 01, pp. 569–577 (2019)
3. Janiesch, C., et al.: The internet of things meets business process management: A manifesto. IEEE Systems, Man, and Cybernetics Magazine (2020)
4. Mannhardt, F.: Multi-perspective Process Mining. Ph.D. thesis, Technische Universiteit Eindhoven, Eindhoven (Feb 2018)
5. Pauker, F., Mangler, J., Rinderle-Ma, S., Pollak, C.: centurio.work - Modular Secure Manufacturing Orchestration. In: BPM Industry Track. pp. 164–171 (Sep 2018)
6. Stertz, F., Rinderle-Ma, S.: Detecting and Identifying Data Drifts in Process Event Streams Based on Process Histories. In: CAiSE Forum. pp. 240–252 (2019)
7. Stertz, F., Rinderle-Ma, S., Mangler, J.: Analyzing process concept drifts based on sensor event streams during runtime. In: BPM. pp. 202–219 (2020)
8. Teinemaa, I., Dumas, M., Rosa, M.L., Maggi, F.M.: Outcome-oriented predictive process monitoring: Review and benchmark. ACM Trans. Knowl. Discov. Data **13**(2) (Mar 2019)